AIAA 94-4339
System Sensitivity Analysis Applied to the Conceptual Design of a Dual-Fuel Rocket SSTO

J. Olds
N.C. State University
Raleigh, NC

**5th AIAA/NASA/USAF/ISSMO
Symposium on Multidisciplinary
Analysis and Optimization**
September 7-9, 1994
Panama City Beach, FL

System Sensitivity Analysis Applied to the Conceptual Design of a Dual-Fuel Rocket SSTO

Dr. John R. Olds*
N. C. State University, Raleigh, NC

ABSTRACT

This paper reports the results of initial efforts to apply the System Sensitivity Analysis (SSA) optimization method to the conceptual design of a single-stage-to-orbit (SSTO) launch vehicle. SSA is an efficient, calculus-based MDO technique for generating sensitivity derivatives in a highly multidisciplinary design environment. The method has been successfully applied to conceptual aircraft design and has been proven to have advantages over traditional direct optimization methods.

The method is applied to the optimization of an advanced, piloted SSTO design similar to vehicles currently being analyzed by NASA as possible replacements for the Space Shuttle. Powered by a derivative of the Russian RD-701 rocket engine, the vehicle employs a combination of hydrocarbon, hydrogen, and oxygen propellants. Three primary disciplines are included in the design — propulsion, performance, and weights & sizing. A complete, converged vehicle analysis depends on the use of three standalone conceptual analysis computer codes.

Efforts to minimize vehicle dry (empty) weight are reported in this paper. The problem consists of six system-level design variables and one system-level constraint. Using SSA in a "manual" fashion to generate gradient information, six system-level iterations were performed from each of two different starting points. The results showed a good pattern of convergence for both starting points. A discussion of the advantages and disadvantages of the method, possible areas of improvement, and future work is included.

NOMENCLATURE

Ae	nozzle exit area
AR	area ratio of engine nozzle
C	vector of weights and sizing outputs
CONSIZ	configuration sizing and weights program
E_1	vector of engine outputs 1
E_2	vector of engine outputs 2
GSE	global sensitivity equation
I_{sp}	specific impulse
LH2	liquid hydrogen
$LH2\%_1$	LH2 propellant % in modes 1A and 1B
LOX	liquid oxygen
LSM	local sensitivity matrix
LSV	local sensitivity vector
M_{tr}	transition Mach number
MDO	multidisciplinary design optimization
MR	mass ratio (gross weight/burnout weight)
P	vector of trajectory outputs
POST	program to optimize simulate trajectories
RP1	hydrocarbon rocket propellant (kerosene)
S_{ref}	aerodynamic reference area (wing area)
SDV	sensitivity derivative vector
SSA	system sensitivity analysis
SSTO	single-stage-to-orbit
T_{vac}	engine vacuum thrust
$(T/W)_{eng}$	engine thrust-to-weight ratio
TPS	thermal protection system
$\dot{w}_2$	total propellant flow rate in mode 2
W	vehicle dry (empty) weight
X	vector of design variables
ΔV	velocity change

INTRODUCTION

Multidisciplinary Design Optimization (MDO) is a relatively new field consisting of a broad range of techniques and methods aimed at improving design efficiency, shortening design times, increasing insight into design options and optimums, and making more decision critical information available earlier in

* *- Visiting Assistant Professor, Mechanical and Aerospace Engineering Dept., Member AIAA.*

the design process. MDO methods include calculus-based optimization techniques, stochastic optimization methods (e.g. genetic algorithms, simulated annealing), parameter design methods (e.g. Taguchi methods, response surface methods), implementation of parallel computing strategies, concurrent engineering methods, and many others [1 - 3]. Application of these methods to aircraft and spacecraft design has produced significant improvements in both the design process and the design product [4 - 10].

As part of an ongoing effort to improve the space launch vehicle conceptual design process, NASA - Langley's Vehicle Analysis Branch has been conducting research on a variety of MDO techniques. The goal of the research has been to evaluate and understand the candidate methods that have the most potential to improve current branch design products and processes. Research efforts consist of a combination of literature searches, basic background research, and actual application of a candidate method to a "real world" branch design problem. Typical branch problems include systems-level, complete vehicle conceptual design problems (generally limited to 3-6 disciplines and fewer than 20 systems-level design variables), trajectory optimization (1-2 disciplines, 20 - 50 design variables and constraints), and technology assessment design problems (combinations of discrete variable options). Preferred methods are added to the "MDO toolbox" — a suite of methods that have proven valuable to solving branch problems and are considered "core" methods.

To date, most research emphasis has been placed on parameter methods based on design of experiments theory (Taguchi methods, response surface methods) [6, 8 - 13]. Parameter methods have proven to be a valuable tool for improving branch design products and have a number of advantages for certain problems [2]. In fact, Lepsch [13] successfully applied response surface methods to a dual-fuel SSTO design similar to the one used for the current research. However, parameter methods are approximate methods that optimize on a model of the design space rather than on the true design space itself. Therefore, the result of a parameter method is often a "near optimum", not a true optimum. For most conceptual design problems, a "near optimum" is often satisfactory, but in some cases, a true optimum might be desirable. To evaluate advantages of the latter approach, recent branch research has addressed optimization methods that work on the actual design space.

This paper reports the initial application of the System Sensitivity Analysis (SSA) method (outlined below) to a conceptual launch vehicle design of current interest to the branch — the dual-fuel SSTO. Specifically, this paper reports the somewhat "manual" application of this technique to a problem involving three *standalone* computer analysis codes, where "manual" refers to a non-automated process of generating derivatives by running the analysis codes separately and assembling finite difference derivatives manually. Also, as is often the case in actual design organizations, information in the current research is exchanged in a non-automated fashion between computer programs that may run on different platforms or be run by different analysis experts. For example, ascent trajectory information produced by the performance code is passed to the weights & sizing code (either via hard copy or electronically). The resulting weights & sizing information is then used by the performance expert to update the ascent trajectory input files, and the analysis proceeds in an iterative fashion until the process converges. SSA is well suited to such problems.

While the current paper discusses the solution of the design problem utilizing three standalone disciplinary analysis codes, a parallel effort was undertaken by Braun [14], to *integrate* the three analysis codes into a single, monolithic design code that was directly coupled to an optimizer. Code integration, where possible, has a number of advantages including reduced "human" time, reduced overall optimization time, and increased accuracy of the results [14]. Note that the SSA technique is not limited to "manual" application. References 4 and 7 discuss the application of the SSA technique to aircraft design problems with integrated analysis subroutines. However, since there are a number of problems in aerospace design that cannot be integrated, the present research is considered valuable as an initial application of a "manual" method that may find utility in a variety of future applications. Where appropriate, the optimization results of the present study are compared to the results of the integrated code of reference 14.

VEHICLE OVERVIEW

NASA is currently considering a number of options to possibly replace the Space Shuttle and to provide future low cost access to space and to the Space Station for both cargo and personnel. One of the options is a new, advanced single-stage-to-orbit (SSTO) launch vehicle as shown in figure 1. Such a vehicle would become operational in the 2005 - 2010 time frame.

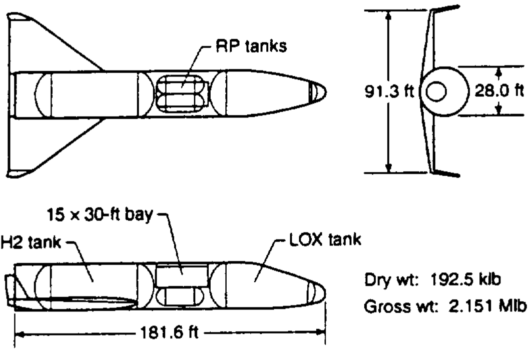

Figure 1 - Dual Fuel SSTO Layout

This particular vehicle makes use of a dual-fuel propulsion system (fueled by both LH2 and RP1) built around derivatives of a Russian RD-701 (sometimes referred to as the RD-704) engine concept. Compared to similar all hydrogen fueled vehicles, dual-fuel vehicles have higher propellant bulk densities, smaller tank volume requirements, and lighter dry weights [13].

Operability, maintainability, and reliability are emphasized throughout the vehicle in keeping with a "design for operations" rather than a "design for performance" philosophy. Integral propellant tanks are constructed of aluminum-lithium, while other major structural components are constructed of graphite-polyether-etherketone composite. A 15% margin on empty weight is included.

Mission

The vehicle is designed to deliver two crew and a 25 klb payload to a 220 Nmi. x 220 Nmi. x 51.6° orbit from Kennedy Space Center. The payload bay is 15 ft. in diameter and 30 ft. long. Initial orbit insertion is to a 50 Nmi. x 100 Nmi. parking orbit. 1100 fps of ΔV is included for orbital transfer to the final mission orbit, for additional on-orbit maneuvering, and for deorbit. After a mission duration of up to five days, the vehicle returns to an unpowered landing at Kennedy Space Center. The winged configuration provides adequate cross range capability to provide several landing opportunities per day.

Propulsion System

The RD-701 derivative, dual-fuel, dual-nozzle position rocket engine is capable of operating in a number of distinct modes as shown in table 1. Nozzle positions are shown in figure 2.

Table 1 - Engine Operating Modes

	Mode 1A	Mode 1B	Mode 2
Fuel	LH2-RP1	LH2-RP1	LH2
Oxidizer	LOX	LOX	LOX
Nozzle	compact	extended	extended

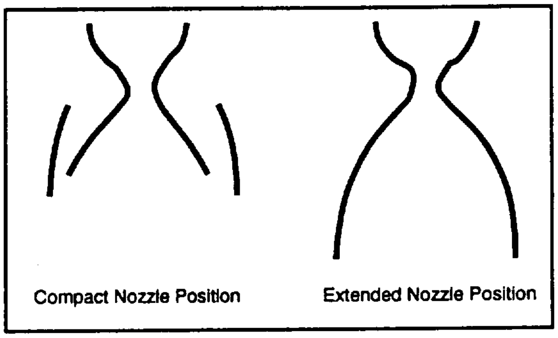

Figure 2 - Engine Nozzle Positions

From liftoff, the engine operates in mode 1A, burning both LH2 and RP1 fuels and with the nozzle in compact position to minimize nozzle back pressure losses. At some optimal transition Mach number (M_{tr1}), the nozzle is extended giving a higher nozzle area ratio and higher effective I_{sp}. At a second optimal transition Mach number (M_{tr2}), the engine transitions to burning only hydrogen fuel at a predetermined propellant mixture ratio (LOX/LH2 by weight). Mode 2 operation has a lower thrust and higher I_{sp} than either mode 1A or 1B. As explained below, many of the engine operating characteristics will be treated as design variables to minimize the overall vehicle dry (empty) weight.

CONCEPTUAL DESIGN

Launch vehicle conceptual design is highly multidisciplinary — involving vehicle geometry, aerodynamics, aeroheating, structures, propulsion, performance, and weights & sizing. In a fully coupled design, many time consuming iterations are required between all of the disciplines in order to converge on a vehicle solution for a single set of design variable values. Optimization (a process requiring many individual design solutions) of a fully coupled vehicle design is a very large, if not impossible, problem given today's computing capabilities. In order to reduce design time and make the process more suitable for the current optimization process, several approximations can be made.

Vehicle external geometry and internal packaging layouts are determined for a reference vehicle and small perturbations from the reference vehicle do not invalidate the geometry. In most cases, the vehicle external shape is only allowed to change based on overall scale — not relative size changes. That is, the shape of the external mold lines is conserved and the entire vehicle grows larger or smaller. The effect of changes in the relative volumes of the internal propellant tanks on the overall body volume is modeled as a simple second-order equation.

Since the external vehicle mold lines are not allowed to change, aerodynamic coefficients for a given Mach number and vehicle attitude will scale primarily with aerodynamic reference area. Reynolds number effects based on a reference trajectory are considered constant. Therefore, aerodynamic coefficients versus Mach number and angle-of-attack generated for a reference vehicle are considered constant for the optimization process. In some cases, relative wing size is changed slightly during the optimization process to maintain a certain wing loading at landing. The effects of these small changes on vehicle aerodynamics are typically ignored.

Aeroheating rates and temperatures are determined for a reference vehicle geometry and entry. Thermal protection system (TPS) requirements for various locations on the vehicle are then determined from these values. For small changes in the reference vehicle, the TPS requirements on a per area basis are considered constant. TPS weights, therefore, are scaled with wing and fuselage wetted areas.

Complete finite element structural analysis is typically too time consuming to include in a launch vehicle conceptual design process. Using a reference vehicle and reference loads, structural unit weights are calculated for various structural components of the vehicle (wing, tanks, body, etc.). These unit weights are then used to calibrate mass estimating relationships in the weights & sizing code which, in turn, is used to estimate changes in structural weights with respect to changes in vehicle size.

For the current vehicle, previous work by engineers at NASA - Langley had established the reference geometry layout, aerodynamic coefficient tables, aeroheating requirements, and structural unit weights. The remaining three disciplines (propulsion, performance, and weights & sizing) are highly coupled and interrelated.

Propulsion

Parametric relationships describing the propulsion system performance and engine weight were provided by Pratt & Whitney. These relationships were originally derived from the application of second-order response surface methods to a set of RD-701 engine conceptual designs. For example, an equation for mode 2 engine vacuum I_{sp} was determined as a second-order function of mixture ratio in mode 2 and area ratio in mode 2 (AR_2). Similar equations were available for thrusts, exit areas, I_{sp}'s, propellant bulk density, engine weight, etc. These equations were entered into a computer spreadsheet program so that engine characteristics in all three operation modes could be rapidly determined as functions of inputs (design variables). The four engine design variables are:

1) percentage of total propellant flow in mode 1A and mode 1B that is LH2 fuel ($LH2\%_1$)
2) nozzle area ratio in compact position (AR_1),
3) nozzle area ratio in extended position (AR_2),
4) LOX/LH2 mixture ratio (by weight) in mode 2.

All other propulsion parameters were functions of these four variables.

Performance

The vehicle ascent trajectory was optimized using POST-3D (Program to Optimize Simulated Trajectories) [15]. The baseline ascent trajectory contained 21 independent variables (primarily inertial pitch angles) and 9 constraints (including a maximum wing loading condition of 2.5*vehicle dry weight, a maximum dynamic pressure of less than 1000 psf, and several orbital insertion target constraints). Three trajectory constraints were inequality constraints and six were equality constraints. An initial, overall vehicle thrust-to-weight ratio of 1.2 was used for all ascent trajectories.

Note that the performance discipline actually involves the optimization of the ascent subproblem. POST uses a nonlinear quadratic programming optimizer internally. For a given set of propulsion system characteristics, vehicle gross weight, aerodynamic reference area, and vehicle dry weight, POST calculates an optimized trajectory based on minimum fuel weight consumed. The use of minimum fuel weight consumed (i.e. minimum mass ratio) in the ascent subproblem is somewhat inconsistent with the overall system objective to minimize vehicle dry weight. However, POST is incapable of minimizing vehicle dry weight since dry weight is calculated in a separate analysis code.

As part of the ascent trajectory subproblem, values for transition Mach numbers from propulsion mode 1A to 1B (M_{tr1}) and from mode 1B to 2 (M_{tr2}) are required. Because of the large effect these two variables have on LH2 and RP1 propellant fractions (and thus tank volumes and overall vehicle dry weight), they were treated as system-level design variables. The remaining 21 local trajectory design variables were assumed to have a less significant effect on dry weight (primarily influencing trajectory constraints) and therefore the inconstancy regarding POST's use of minimum fuel consumption as an objective function was assumed to be small.

Weights and Sizing

Vehicle size (i.e. overall scale) and weights were determined using CONSIZ [16]. CONSIZ is a standalone program containing a series of mass estimating relationships derived from historical regression, perceived technology level, and finite element structural analysis. For a given propellant requirement (typically in the form of the useful scaling parameter mass ratio, MR) and individual propellant fractions, CONSIZ can quickly iterate toward a solution for which the actual vehicle mass ratio matches the required mass ratio. CONSIZ works by scaling the propellant tanks up (and recalculating related areas, volumes, and weights) to increase MR (increase the propellant fraction) or down to decrease MR. When a final solution is reached, estimates for structural weights, propellant weights, dry weight, gross weight, and vehicle scale are all available from the CONSIZ output file.

DESIGN VARIABLES

The six system-level design variables used for this optimization problem are shown in table 2. Four of the variables are directly related to the propulsion systems. Two (Mtr_1 and Mtr_2) are primarily trajectory variables, but have a significant effect on the overall propellant fractions.

Table 2 - System-level Design Variables

Variable	Description
$LH2\%_1$	LH2 propellant percentage in modes 1A and 1B
AR_1	nozzle area ratio in modes 1A and 1B
AR_2	nozzle area ratio in mode 2
M_{tr1}	transition Mach number from mode 1A to mode 1B
M_{tr2}	transition Mach number from mode 1B to mode 2
Mixture Ratio 2	LOX/LH2 ratio in mode 2

The objective of the system-level optimization process is to minimize vehicle dry (empty) weight. A single system-level constraint limits AR_2 to less than twice AR_1, i.e.

$$\frac{AR_2}{AR_1} - 2 \leq 0 \qquad (1)$$

MULTIDISCIPLINARY ANALYSIS PROCESS

For a given set of design variables, a multidisciplinary, iterative analysis process is required to determine the vehicle dry weight. Information flows between the three analysis disciplines/codes as shown in figure 3.

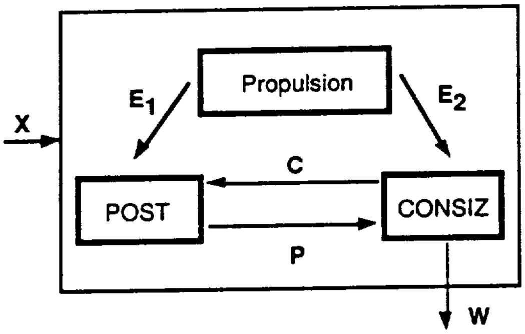

Figure 3 - Analysis Process Information Flow

Note that engine information only flows *from* the engine code *to* POST and CONSIZ. There is no feedback to the engine code. However, POST and CONSIZ are tightly coupled, and information flows both ways between the two. The vector of six design variables is represented by the **X** vector in figure 3. Of the six possible design variables, the engine code depends on four (LH2%$_1$, AR$_1$, AR$_2$, and Mixture Ratio 2), POST depends on four (LH2%$_1$, M_{tr1}, M_{tr2}, and Mixture Ratio 2), and CONSIZ depends on one (LH2%$_1$). The vectors $\mathbf{E}_1$, $\mathbf{E}_2$, **C**, **P**, and **W** represent an additional 21 ***internal*** design variables (table 3). The 1 x 1 vector **W**, the dry weight, is actually the objective function.

Table 3 - Internal Design Variables

Vector	Variables
$\mathbf{E}_1$	T_{vac1A}, Ae_{1A}, T_{vac1B}, Ae_{1B}, $\dot{w}_2$, T_{vac2}, Ae_2, RP1%$_1$
$\mathbf{E}_2$	T_{max}/T_{sl}, $I_{sp\ vac2}$, $I_{spsl\ 1A}$, $(T/W)_{eng}$, RP1%$_1$, propellant bulk density
C	S_{ref}, gross weight, max. wing normal force
P	mass ratio, LH2%$_T$, RP1%$_T$
W	dry (empty) weight

A typical multidisciplinary analysis cycle proceeds as shown in figure 4. The POST/CONSIZ iteration proceeds until the mass ratio from one iteration to the next results in an absolute change of less than 10^{-5} — typically about 4 iterations. The tolerance on MR yields a dry weight convergence of less than a pound and a gross weight convergence of less than a few pounds. Execution of this analysis cycle is required many times during the design process.

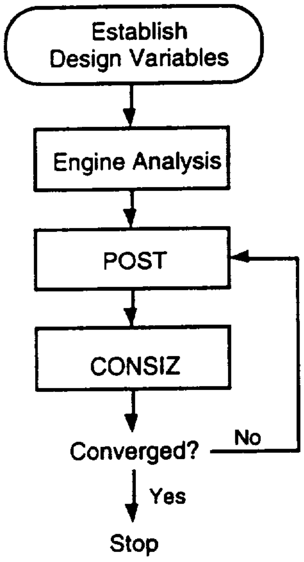

Figure 4 - Analysis Cycle

SYSTEM SENSITIVITY ANALYSIS

It is the goal of every design process to determine the best combination of design variables that meets all constraints and optimizes the objective function. There are many optimization techniques available based on exploitation of the local gradient of an objective function as a favorable search direction [17-18]. The specific system-level optimization method used in this research was the method of feasible directions [19].

System Sensitivity Analysis (SSA) is a technique introduced by Sobieski [20-24] for determining gradient information (also referred to as sensitivity derivatives). The method is particularly suited for use in multidisciplinary design environments due to the unique method of determining the sensitivity derivatives. A brief description of the adaptation of the method for the current problem method is included here.

Using the information flow as shown in figure 3 , we can write the following equation for the variables contained in the $\mathbf{E}_1$ vector:

$$\mathbf{E}_1 = \mathbf{E}_1(\mathbf{X}) \tag{2}$$

and differentiating both sides,

$$d\mathbf{E}_1 = \frac{\partial \mathbf{E}_1}{\partial \mathbf{X}} d\mathbf{X} \tag{3}$$

and,

$$\frac{d\mathbf{E}_1}{d\mathbf{X}} = \frac{\partial \mathbf{E}_1}{\partial \mathbf{X}} \tag{4}$$

where the term $\partial \mathbf{E}_1/\partial \mathbf{X}$ is an 8 x 6 submatrix since there are eight internal variables in the $\mathbf{E}_1$ vector and each one must be differentiated by each of the six design variables in **X**.

Similarly, the POST analysis module is a function of three vector inputs:

$$\mathbf{P} = \mathbf{P}(\mathbf{E}_1, \mathbf{C}, \mathbf{X}) \tag{5}$$

$$d\mathbf{P} = \frac{\partial \mathbf{P}}{\partial \mathbf{E}_1} d\mathbf{E}_1 + \frac{\partial \mathbf{P}}{\partial \mathbf{C}} d\mathbf{C} + \frac{\partial \mathbf{P}}{\partial \mathbf{X}} d\mathbf{X} \tag{6}$$

and,

$$\frac{d\mathbf{P}}{d\mathbf{X}} = \frac{\partial \mathbf{P}}{\partial \mathbf{E}_1}\frac{d\mathbf{E}_1}{d\mathbf{X}} + \frac{\partial \mathbf{P}}{\partial \mathbf{C}}\frac{d\mathbf{C}}{d\mathbf{X}} + \frac{\partial \mathbf{P}}{\partial \mathbf{X}} \tag{7}$$

where the term **dP/dX** is a 3 x 6 submatrix and each of the other terms in the equation is an appropriately sized submatrix. Continuing the process on the other output vectors yields:

$$\frac{d\mathbf{C}}{d\mathbf{X}} = \frac{\partial \mathbf{C}}{\partial \mathbf{E}_2}\frac{d\mathbf{E}_2}{d\mathbf{X}} + \frac{\partial \mathbf{C}}{\partial \mathbf{P}}\frac{d\mathbf{P}}{d\mathbf{X}} + \frac{\partial \mathbf{C}}{\partial \mathbf{X}} \tag{8}$$

$$\frac{d\mathbf{W}}{d\mathbf{X}} = \frac{\partial \mathbf{W}}{\partial \mathbf{E}_2}\frac{d\mathbf{E}_2}{d\mathbf{X}} + \frac{\partial \mathbf{W}}{\partial \mathbf{P}}\frac{d\mathbf{P}}{d\mathbf{X}} + \frac{\partial \mathbf{W}}{\partial \mathbf{X}} \tag{9}$$

$$\frac{d\mathbf{E}_2}{d\mathbf{X}} = \frac{\partial \mathbf{E}_2}{\partial \mathbf{X}} \tag{10}$$

Equations 4, 7, 8, 9, and 10 can be combined and written in matrix form.

$$\begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{I} & 0 & 0 & 0 \\ -\frac{\partial \mathbf{P}}{\partial \mathbf{E}_1} & 0 & \mathbf{I} & -\frac{\partial \mathbf{P}}{\partial \mathbf{C}} & 0 \\ 0 & -\frac{\partial \mathbf{C}}{\partial \mathbf{E}_2} & -\frac{\partial \mathbf{C}}{\partial \mathbf{P}} & \mathbf{I} & 0 \\ 0 & -\frac{\partial \mathbf{W}}{\partial \mathbf{E}_2} & -\frac{\partial \mathbf{W}}{\partial \mathbf{P}} & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{E}_1}{d\mathbf{X}} \\ \frac{d\mathbf{E}_2}{d\mathbf{X}} \\ \frac{d\mathbf{P}}{d\mathbf{X}} \\ \frac{d\mathbf{C}}{d\mathbf{X}} \\ \frac{d\mathbf{W}}{d\mathbf{X}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{E}_1}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{E}_1}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{P}}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{C}}{\partial \mathbf{X}} \\ \frac{\partial \mathbf{W}}{\partial \mathbf{X}} \end{bmatrix} \tag{11}$$

Equation 11 is called the global sensitivity equation (GSE) and has the form **[A]X=b**. The matrix in the "A" position is called the local sensitivity matrix (LSM). The matrix in the "**b**" position is called the local sensitivity vector (LSV). The matrix in the "X" position contains the system-level sensitivity derivatives (total derivatives) and will be referred to as the sensitivity derivative vector (SDV). Given the LSM and LSV matrices, the SDV can be calculated using matrix inversion methods. One of the elements of the SDV, the 1 x 6 submatrix **dW/dX**, contains the gradient of the objective function with respect to each of the design variables. Note that each term in the local sensitivity vector, the local sensitivity matrix, and sensitivity derivative vector is actually a submatrix. The LSM is a 21 x 21 matrix. The LSV and SDV are both 21 x 6 matrices. The LSM is a relatively sparse matrix for this application while the LSV and SDV are well populated.

The terms in the LSM all describe partial derivatives of internal variables with respect to other internal variables. That is, all of the derivatives represent the isolated (uncoupled) changes in each discipline with respect to changes in the other disciplines. Similarly, the LSV represents the local changes in each discipline with respect to the design variables. Herein lies one of the advantages of SSA — the terms of the LSV and LSM can be calculated in parallel (each discipline at the same time) and all sensitivity derivatives are then determined simultaneously in one matrix operation. That is, complete, time-consuming, iterative solutions of the entire system are not required for every gradient calculation. In fact, depending on the internal linearity of the problem, it is often possible to reuse the LSM information for 2-3 iterations and use only updated

LSV information to calculate a new SDV. Reuse of the LSM improves the efficiency of the method by reducing the number of analysis code runs required to generate partial derivative values for each iteration. However, violations of the linearity assumption could lead to errors in the calculation of the SDV.

OPTIMIZATION PROCESS

Before discussing the results of the optimization, it is necessary to discuss the methods used to calculate the partial derivatives in the LSM and LSV and the methods used to exploit a search direction via a line search.

Partial Derivatives

Accurate calculation of partial derivatives is absolutely critical to the use of the SSA method. Since the propulsion variables $\mathbf{E}_1$ and $\mathbf{E}_2$ were available as relatively simple (typically second-order) equations, *analytical* derivatives were taken for $\partial \mathbf{E}_1/\partial \mathbf{X}$ and $\partial \mathbf{E}_2/\partial \mathbf{X}$. However, all other derivatives were calculated using finite differences.

Initially, forward differences were attempted in order to reduce the number of function evaluations. Forward differences require only one extra function evaluation per derivative. The partial derivative of some function, f, with respect to some variable, x, can be approximated by:

$$\frac{\Delta f}{\Delta x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} \qquad (12)$$

where all other inputs to f are held constant and Δx is some small value. The accuracy of the partial derivative estimate improves as Δx is reduced. However, neither POST nor CONSIZ produces enough accuracy in their output files to allow very small Δx values without introducing numerical roundoff errors. Using a Δx value of 1%, several iterations were performed from an initial starting point before it became apparent that iterations were not producing sufficient reductions in objective function (figure 5), and that $LH2\%_1$ appeared to be diverging (figure 6) from the optimum obtained by use of the integrated design code of reference 14. (Note that the optimum from reference 14 applied an additional angle-of-attack limit on the ascent trajectory not employed in this research. To allow accurate comparisons, the dry weight at the optimum design variable settings from ref. 14 was recalculated without the angle-of-attack constraint).

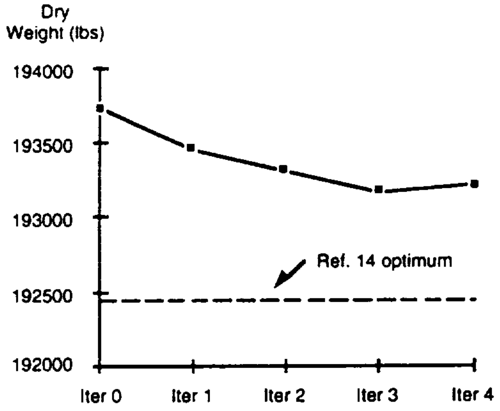

Figure 5 - Dry Weight Progress for Forward Diffs

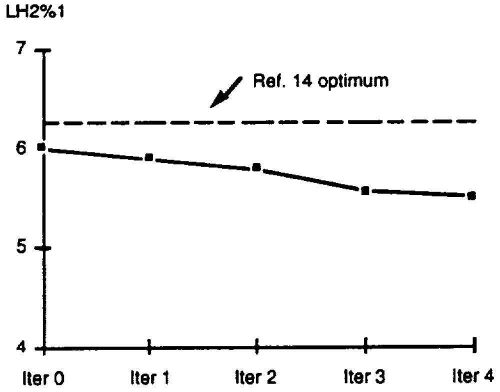

Figure 6 - $LH2\%_1$ Progress for Forward Differences

After iteration 4, complete converged test solutions were calculated for an increased $LH2\%_1$ and a decreased $LH2\%_1$. The results showed a lower dry weight for the *higher* $LH2\%_1$ and cast doubt on the accuracy of the derivatives. Further investigation using more accurate central differences revealed an interesting problem. The total derivative for change in weight with respect to $LH2\%_1$, is calculated from the following equation extracted from the GSE,

$$\frac{dW}{dLH2\%_l} = \frac{\partial W}{\partial P}\frac{dP}{dLH2\%_l} + \frac{\partial W}{\partial E_2}\frac{dE_2}{dLH2\%_l} + \frac{\partial W}{\partial LH2\%_l} \qquad (13)$$

where a key term in the $d\mathbf{P}/dLH2\%_1$ submatrix is $dMR/dLH2\%_1$ which, in turn, is highly dependent on the difference between a large number times $\partial MR/\partial T_{vac2}$ and a large number times $\partial MR/\partial \dot{w}_2$. At the point tested, errors of about 3% in these two values caused a combined error of about 25% in $dMR/dLH2\%_1$. The 25% error in $dMR/dLH2\%_1$ causes a complete sign reversal (direction change) in $dW/dLH2\%_1$!

As a result, one must draw the conclusion that due to the way that SSA calculates total derivatives as a sum of a series of derivative products, the method is highly dependent on accurate derivatives and in some cases, small errors in local derivatives can combine in adverse ways to produce large errors in the system-level total derivatives. For that reason, the remainder of this research relied on more accurate central differences with a 1% Δx of the form:

$$\frac{\Delta f}{\Delta x}=\frac{f(x+\Delta x)-f(x-\Delta x)}{2\Delta x} \qquad (14)$$

The disadvantage of having to use central differences lies in the fact that two function evaluations are required per derivative — effectively doubling the computer time required to populate the LSV and LSM.

Although not used for this application, a potentially useful tool for future SSA applications is automatic code differentiation. Research in the field (for example, the ADIFOR code in reference 25) has demonstrated the ability to take an existing source code and automatically modify it to create analytical derivatives of all variables during runtime. These derivatives would be local derivatives — exactly the partial derivatives used in the GSE. Automatic differentiation of individual analysis codes could result in significantly more accurate and faster calculation of derivatives versus methods employed in this paper.

A second possibility to improve the efficiency by which partial derivatives are determined concerns the use of post-optimality criteria [26]. Every trajectory evaluated by POST (even those used for finite difference derivatives) requires the solution of a suboptimization problem to find the minimum propellant consumption trajectory that meets all ascent constraints. POST is the most time consuming disciplinary analysis. It may be possible to make use of the Lagrange multipliers at a reference trajectory solution to help approximate the sensitivities to changes in trajectory parameters without having to reoptimize the entire problem. The result could be a significant reduction in the time required to populate the GSE.

Line Searches

Once the LSM and LSV was populated with partial derivatives, Matlab was used to solve the GSE for the sensitivity derivative vector. The Automated Design Synthesis (ADS) code was then used to determine an appropriate search direction using the method of feasible directions [19]. The optimization process depends on finding the minimum dry weight along the direction of search. Several options were examined for the line search scheme.

1) Perform a full iterative solution of the entire multidisciplinary analysis problem (figure 4) at two points along the line search so that the starting point and the last of the two new points bound a minimum dry weight along the line search. A second-order polynomial is then fit through all three points. The minimum of the polynomial is then used as the optimum step size along the search direction. This is the most time consuming option, but also the most accurate of the four options considered.

2) Use a simple linear extrapolation of the dry weight (since the gradient is already known) while imposing move limits on the design variables. This is the simplest, least time consuming option.

3) Use linear estimates for the changes in **P** and E_2 (available in the SDV) as inputs into CONSIZ, and then calculate a new dry weight, **W**, for two new points along the line search. Use a polynomial interpolation similar to that described in option 1. This option depends on a linear estimate for changing mass ratio (i.e. a linear **P** vector). Move limits may also be necessary for this option.

4) Use linear estimates for changes in $\mathbf{E}_1$ and **C** to run one full POST analysis and then use the actual **P** vector and a linear estimate of E_2 in CONSIZ

to calculate a dry weight, **W**. Execute this process for two points along the search direction and perform a polynomial interpolation similar to that described in option 1. This option does not depend on a linear approximation for changing mass ratio. Move limits may be required for this option.

For a typical line search, figure 7 shows the effect of each of the four line search options versus the distance, alpha, along the line search direction.

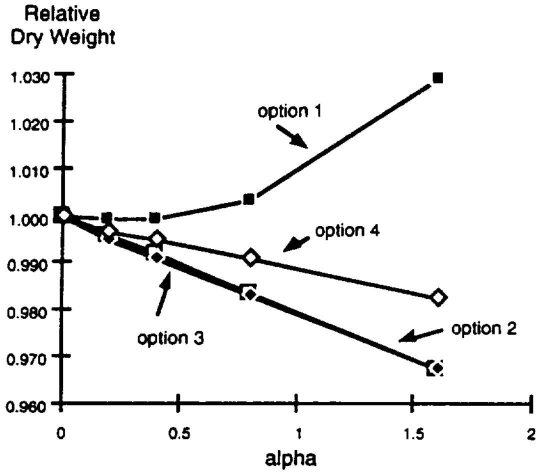

Figure 7 - Line Search Options

The mass ratio is a very non-linear function of input variables into POST. The linear extrapolation option 2 and the linear approximations into CONSIZ (option 3) were deemed incapable of capturing the effects of curvature along a line search and were not selected. The full iterative solution at each point (option 1) is the most accurate and captures the curvature effect of MR well. However, option 1 is the most time consuming since it requires 3-4 POST runs and 3-4 CONSIZ runs for each point solution. For these reasons, option 4 was selected for line searches for this work. Option 4 requires a single POST run (with linear updates in $\mathbf{E}_1$ and **C**) and one CONSIZ run (with the new **P** and a linear update of $\mathbf{E}_2$) per solution and represents a compromise between accuracy and run time. Option 4 does calculate an actual MR (not a linear approximation), but linear approximations for other internal variables are used in POST and CONSIZ.

Using line search option 4, move limits representing a maximum 10% change in the design variables were used for the first starting point. 20% move limits were used for the second starting point. Option 4 generally performed well, but broke down very close to the optimum with the smaller move limits or moderately close to the optimum with the larger move limits. Therefore, the use of the full iterative solution (option 1) was required for iterations 5 and 6 for both starting points (results below).

The evaluation of dry weight along a line search may be a place to improve the efficiency of the SSA method. Additional work may reveal an appropriate approximation that will allow a line search to combine the accuracy of the full iterative solution (option 1) and the rapid evaluation of options 2 or 3. That is, a simple second-order approximate model for changing dry weight as a function of alpha might be determined with more knowledge of the system — particularly changing MR. Future work will address this issue.

OPTIMIZATION RESULTS

Using central finite differences on POST and CONSIZ variables and the line search option 4 discussed above, six system-level iterations for two different starting points were performed using the method of feasible directions. A full iterative solution was performed at the end of each line search. The LSM was updated only on the 1st and 4th iterations. The LSV was updated every iteration. The Matlab program was used to calculate the SDV for each iteration. The results of the six iterations are shown in table 5 and figures 8 - 9. The optimum results of the integrated code of reference 14 are shown for useful comparison in table 4 (the dry weight shown is for the ref. 14 angle-of-attack ascent constraint removed).

Table 4 - Optimum from Reference 14

Dry Weight (lbs)	192443
LH2%$_1$	6.26
AR$_1$	59.67
AR$_2$	119.35
M_{tr1}	0.899
M_{tr2}	7.31
Mixture Ratio 2	6.996

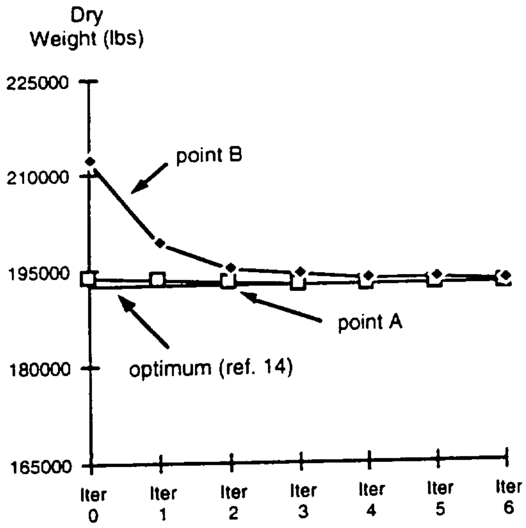

Figure 8 - Dry Weight Convergence

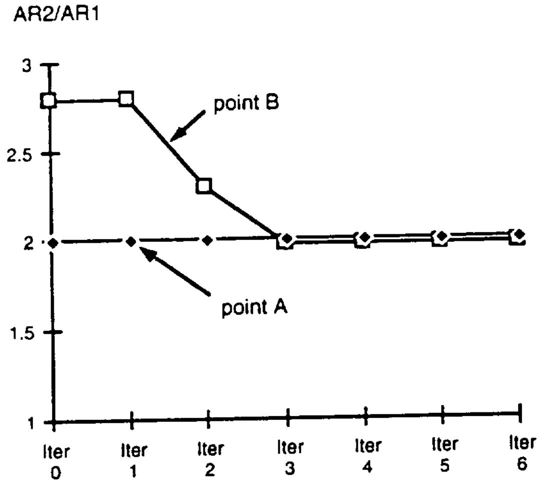

Figure 9 - System-Level Constraint

Table 5 - Optimization Results for Two Starting Points

Starting Point A

	Start	Iter 1	Iter 2	Iter 3	Iter 4	Iter 5	Iter 6
Dry Weight	193,812	193,323	192,988	192,665	192,649	192,478	192,477
$LH2\%_1$	6	6.0870	6.2282	6.6054	6.5295	6.5286	6.4908
AR_1	60	60.001	60.002	60.005	60.008	60.007	60.007
AR_2	120	120.002	120.004	120.010	120.015	120.013	120.013
M_{tr1}	1.2	1.0800	0.9720	0.8748	0.9623	0.9346	0.9086
M_{tr2}	6	6.1230	6.2865	6.6375	7.0050	7.0500	7.0700
Mix Ratio 2	7.5	7.3235	7.2343	7.1455	6.7655	7.0228	7.0493
MR	8.30707	8.27507	8.24927	8.20145	8.19325	8.21671	8.22505

Starting Point B

Dry Weight	212,297	199,189	195,045	194,302	193,347	193,150	193,086
$LH2\%_1$	4	4.8000	5.7600	6.5473	6.5648	6.6477	6.7968
AR_1	50	50.0175	59.165	66.665	66.648	66.664	66.615
AR_2	140	139.995	135.426	131.680	131.682	131.678	131.674
M_{tr1}	2	1.6860	1.3927	1.2622	1.0098	1.0016	0.9850
M_{tr2}	5	5.5920	5.8557	5.9676	6.1145	6.2539	6.4880
Mix Ratio 2	9	8.2150	7.9036	7.7520	7.008	7.2430	7.2941
MR	8.91954	8.53579	8.32264	8.20199	8.11520	8.13126	8.12640

Point A was considered a "good" starting point — starting on the constraint boundary and within 1% of the optimum dry weight. Initial design variables differed from the optimum values by a maximum of 33%. Point B was considered a "bad' starting point — starting in the infeasible region and 10% away from optimum dry weight. Initial design variables for point B differed from the optimum values by a maximum of 122%. Both starting points proceed well toward convergence. Point A reaches to within 40 lbs. (about .02%) of the optimum within six iterations and it is reasonable to assume that the point B would reach a point that close to the optimum in 3 - 6 more iterations.

While these results show good progression toward an optimum in dry weight, by comparison, the actual design variables are still relatively far away from the optimum values after six iterations (as much as a 3% - 4% difference in optimum design variable values for point A at iteration 6 compared to a .02% difference in optimum dry weight). The reason is that the design space for this vehicle is very flat with respect to these design variables. For example, small savings in dry weight depend on differences between the effects of improved (smaller) mass ratio that comes from *increased* use of hydrogen fuel and increased propellant bulk density that results from *decreased* use of hydrogen fuel. These two competing effects are roughly the same magnitude near the optimum so that excursions of the design variables of a few percent have relatively little effect on the overall dry weight. However, with more iterations it is likely that the design variables would move closer to the optimum values shown in table 5.

Note that M_{tr2} represents a key design variable in the trade between increased MR and increased propellant bulk density. As M_{tr2} moves higher, the transition from LH2-RP1 fuel (mode 1B) to LH2-only fuel (mode 2) is delayed and the overall propellant bulk density is increased. However as M_{tr2} increases, the overall mass ratio worsens (increases). These two effects are carefully balanced in the selection of an optimum M_{tr2}.

CPU Time

For both starting point A and starting point B, the local sensitivity matrix is only updated prior to iteration 1 and iteration 4. The local sensitivity vector is updated prior to every iteration. By using central differences, 22 POST runs (each a suboptimum problem of its own) are required to populate the POST-related derivatives in the LSM and 8 POST runs are required to populate the LSV. Similarly, 18 CONSIZ runs are required to populate the CONSIZ-related derivatives in the LSM and 2 CONSIZ runs are required to populate the LSV. POST runs are by far the most time consuming analysis — averaging about 585 cpu seconds per run on a SGI Indigo 2 computer. Assuming 2 additional POST runs for a line search and 4 POST runs for a fully converged solution at the end of a line search, the 36 POST runs required for a full iteration with a new LSM take about 21,060 cpu seconds (about 5.85 cpu hours) (table 6). By comparison, CONSIZ runs take about 15 cpu seconds each on an SGI Indigo 2 computer. The 26 CONSIZ runs required for a full iteration (with LSM) take approximately 390 cpu seconds — about 2% of the POST run times. Recall that the engine-related derivatives (dE_1/dX and dE_2/dX) are calculated using analytical formulas on a spreadsheet program (less than 1/2 second on a Macintosh Quadra for all derivatives) so that calculation of engine-related derivatives in the LSM and LSV is considered insignificant with respect to cpu time. Calculation of the SDV (for a given LSM and LSV) in Matlab takes about 2 cpu seconds on a Sun Sparcstation. Table 6 shows cpu times for each analysis code based on approximate average cpu times for each analysis run.

Table 6 - Approximate CPU Times (secs)

	POST	CONSIZ	Engine	Total
Full LSM & LSV iter.	21,060	390	< 1	21,451
LSV-only iter.	8,190	120	< 1	8,311
total for 6 iterations	74,880	1,260	< 3	76,143

Of course, for this "manual" application of the method, discussions of cpu time are not as important as "real" iteration time. "Real" iteration time is dominated by editing input files, transferring files from one computer to another, data entry of partial derivatives, and calculation of search directions.

Actual time to complete an iteration with recalculation of the LSM was about 10-12 hours. LSV-only iterations took about 6-8 hours of real time each.

Note that one of the primary advantages of the SSA method is its ability to calculate the partial derivatives of each of the disciplines simultaneously. For example, since POST and CONSIZ are both standalone codes, two different experts could be running those two codes at the same time. Theoretically, parallel calculation of these derivatives can reduce the overall design time. This problem, however, is dominated by the execution of the POST code. Parallel execution in this case would only save about 2% of cpu time. Therefore, one of the advantages of SSA cannot be realized in this problem.

Since the parallel execution advantage of SSA is not very important to this problem, it is reasonable to consider the application of a direct optimization method at the system level. That is, calculation of gradient information by perturbing each of the six design variables at the system level by +1% and -1%, one at a time, followed by a complete iterative solution of the entire design at the new point. Assuming that 4 POST runs and 4 CONSIZ runs are required for each full iterative solution and that two solutions are required per design variable (using central differences), then 48 POST runs (28,080 cpu seconds) are required to generate a single objective function gradient vector for all six design variables.

Therefore, the GSE approach to calculation of the objective function gradient is more efficient than a direct systems-level optimizer for this problem. In fact, if the direct-optimizer required only *forward* finite differences (24 POST runs for one gradient using 14,040 CPU seconds), the fact that the Local Sensitivity Matrix (LSM) can be reused for several iterations in the SSA method, would still leave SSA the more efficient method when taken over several iterations. That is, six iterations (recalculating the LSM only twice) of SSA take approximately 74,880 cpu seconds for POST runs and 6 iterations of a direct system-level optimizer would take 84,240 cpu seconds for POST runs just to calculate the six gradients. These conclusions are specific to the current design problem and are strongly dependent on the number of internal design variables (related to the number of POST runs per LSM calculation) and the number of analysis runs required for a full iterative solution of the POST/COSIZ loop (related to the number of POST runs required for a direct optimization gradient calculation).

FINAL VEHICLE

While not fully converged, the results of the sixth iteration from starting point A (table 5) are close to the optimum. A simplified weight statement for this vehicle is shown in table 7. A three-view drawing of this vehicle was previously shown in figure 1. The overall vehicle length is 181.6 ft. and the wingspan is 91.3 ft. The theoretical wing planform area is 4043.9 ft^2. The optimized trajectory is shown in figure 10 — altitude (ft.) and wing normal force(lbs.) vs. time — and figure 11 — thrust (lbs.) and I_{sp} (sec.) vs. time.

Table 7 - Simplified Vehicle Weight Statement

Item	Weight (lbs)
Structures	71,015
TPS	18,845
Main Engines	51,359
Other Propulsion	5,792
Other Weights	20,360
Margin (15%)	25,106
Dry Weight	**192,477**
Payload	25,000
LH2 Propellant	158,862
RP1 Propellant	145,242
LOX Propellant	1,585,351
Other Weights	44,038
Gross Weight	**2,150,970**

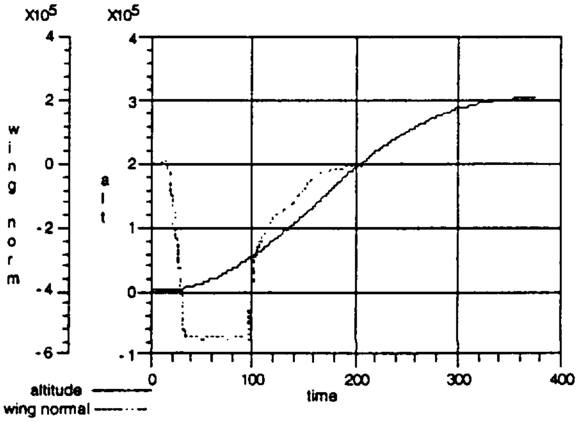

Figure 10 - Final Vehicle Ascent Trajectory

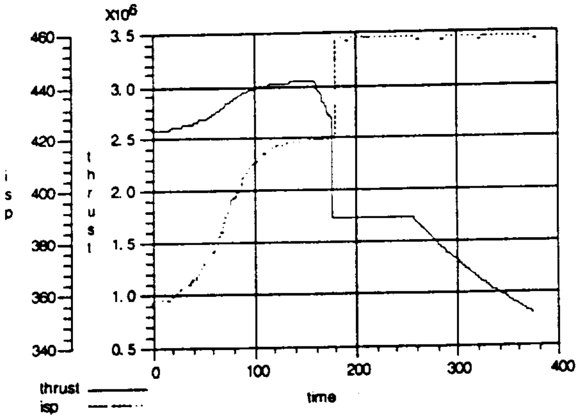

Figure 11 - Final Vehicle Ascent Trajectory

CONCLUSIONS

The results reported in this paper represent the initial application of the system sensitivity analysis method to the conceptual design of a dual-fuel launch vehicle design. The application was generally successful — showing good convergence from two separate starting points. Several other conclusions can also be drawn.

1) SSA is well suited for "manual" application as demonstrated in this research — that is, application to multidisciplinary design problems consisting of non-integrated, separate design codes. The uncoupling of the design problem through the use of local (partial) derivatives and the global sensitivity equation (GSE) is ideally suited to separate analysis codes. Although previous SSA applications have been to design problems with automated execution of disciplinary analysis codes or integrated, monolithic codes, a larger range of problems that cannot be integrated should also be opened to SSA.

2) The method is highly dependent on accurate partial derivatives. At one point in this research, forward finite differences proved to be too inaccurate at 1% perturbation sizes because an adverse accumulation of small local derivative errors led to a major error in a system-level derivative. This effect is primarily due to the way that SSA assembles the GSE as the sum of several individual derivatives. More accurate central differences eliminated the problem, but took twice as many function evaluations per derivative.

3) One of the large advantages of SSA — parallel calculation of partial derivatives — was not realized for this particular problem because the cpu time of one of the disciplinary analysis codes, POST, dominated the total computer time required. Parallel calculation would save very little time for this problem.

4) The reuse of the Local Sensitivity Matrix for several iterations (three in this case) considerably improves the efficiency of the method and reduces the average number of disciplinary calculations required per iteration. For this problem, basic SSA requires less cpu time per iteration than would a direct optimization method calculating gradient information at the system-level. SSA is even more competitive when the efficiency gained from reusing the LSM is taken into account. However, reuse of the LSM for several iterations could introduce errors in the calculation of sensitivity derivatives for highly non-linear problems.

FUTURE WORK

This research represents only the initial application of SSA to a conceptual launch vehicle design problem of the type of interest to the Vehicle Analysis Branch at NASA-Langley. Additional research is needed before a decision is made on adding the method to the "MDO toolbox". The greatest advantage of SSA may lie in its ability to be applied as a "manual" method. As mentioned previously, an *integrated* analysis code with a coupled optimizer was created for the current problem in a parallel research effort [14]. The integrated code produced very good results. While the present research also produced good results, the advantages over an automated method over a manual method in terms of reduced design time, reduced "human" work, and accuracy are obvious. Future research on SSA will focus on problems that cannot be integrated. For example, problems where disciplinary analysis is performed by different experts in different geographically located areas, when the time required to integrate the individual analyses is not warranted for a "unique" problem, or when the source code of a particular analysis code cannot be modified to accommodate code integration (proprietary or commercial codes) may all be "niche" areas for SSA. Future work will also focus on the following areas:

1) Automatic generation of derivatives either a) via an automated process of executing individual analysis codes in a repetitive cycle to generate finite difference derivatives or b) via the automatic generation of analytical derivatives from an existing analysis program source code that has been specially modified by a second code (e.g. ADIFOR). The former approach must also stress improved accuracy of the finite differences due to the importance of accurate derivatives to SSA. Either method could greatly reduce the "real" time required to populate the GSE.

2) Use of post-optimality methods in POST to take advantage of existing Lagrange multipliers at an optimized solution to help calculate sensitivities of the optimized solution and objective function to changes in input parameters. Post-optimality methods can potentially eliminate the current need to completely reoptimize the trajectory subproblem twice for every derivative calculated with central finite differences.

3) Improvements in the line search method used in this research (a single POST run and a single CONSIZ run). A future method should be efficient and require no more cpu time than the current method, but it should more accurately predict the true effect of changing dry weight along a line search (i.e. as accurate as a full iterative solution at each point along a line search). This task might involve creation of a more detailed approximate model for certain internal design variables (particularly mass ratio) as they change along a line search.

ACKNOWLEDGEMENTS

This work was supported under cooperative agreement NCC1-185 between North Carolina State University and the NASA-Langley Research Center. The author would like to thank Bobby Braun, Roger Lepsch, and Larry Rowell of NASA-Langley for their help and support.

REFERENCES

1. Sobieszczanski-Sobieski, J. "Multidisciplinary Design Optimization: An Emerging New Engineering Discipline." presented at the World Congress on Optimal Design, Rio de Janeiro, Brazil, August, 1993.

2. Olds, J.R. "The Suitability of Selected Multidisciplinary Design Techniques to Conceptual Aerospace Vehicle Design." AIAA 92-4791, September, 1992.

3. Frank, P. D., et. al. "A Comparison of Optimization and Search Methods for Multidisciplinary Design." AIAA 92-4827, September, 1992.

4. Levine, M., H. Ide, and S. Hollowell. "Multidisciplinary Hypersonic Configuration Optimization." presented at the 3rd Air Force/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, September, 1990.

5. Barnum, J., et al. "Advanced Transport Design Using Multidisciplinary Design Optimization." AIAA 91-3082, September, 1991.

6. Olds, J.R., and G. Walberg. "Multidisciplinary Design of a Rocket-Based Combined-Cycle SSTO Launch Vehicle using Taguchi Methods." AIAA 93-1096, February, 1993.

7. Consoli, R. D., Sobieszczanski-Sobieski, J. "Application of Advanced Multidisciplinary Analysis and Optimization Methods to Vehicle Design Synthesis." Journal of Aircraft, vol. 29, no. 5, Sept. - Oct., 1992.

8. Stanley, D. O., et al. "Rocket-Powered Single-Stage Vehicle Configuration Selection and Design." AIAA 93-1053, February, 1993.

9. Stanley, D.O, R. Unal, and C. R. Joyner. "Application of Taguchi Methods to Dual Mixture Ratio Propulsion System Optimization for SSTO Vehicles." AIAA 92-0213, January, 1992.

10. Olds , J. R. "Results of a Rocket-Based Combined-Cycle SSTO Design Using Parametric MDO Methods." SAE 94-1165, April, 1994.

11. Bush, L., and R. Unal. "Preliminary Structural Design of a Lunar Transfer Vehicle Aerobrake." AIAA 92-1108, February, 1992.

12. Engelund, W. C., et. al. "Aerodynamic Configuration Design Using Response Surface Methodology Analysis." AIAA 93-3967, August 1993.

13. Lepsch, R.A., D. Stanley, and R. Unal. "Application of Dual-Fuel Propulsion to a Single Stage AMLS Vehicle." AIAA 93-2275, June, 1993.

14. Braun, R., et .al. "Multidisciplinary Optimization Strategies for Launch Vehicle Design." AIAA 94-4341, September, 1994.

15. Brauer, G. L., et. al. "Program to Optimize Simulated Trajectories (POST)." NASA contract NAS1-18147, September, 1989.

16. Lepsch, R. A., et. al. "Utilizing Air-Turborocket and Rocket Propulsion for a Single-Stage-to-Orbit Vehicle." Journal of Spacecraft and Rockets, vol. 28, no. 5, Sept. - Oct., 1991.

17. Fox, R. L. Optimization Methods for Engineering Design. Reading, MA: Addison-Wesley Publishing Company, 1971.

18. Gill, P., W. Murray, and M. Wright. Practical Optimization. London: Academic Press, 1981.

19. Vanderplaats, G. N., H. Sugimoto, and C. Sprague. "ADS-1: A New General Purpose Optimization Program," AIAA Journal, vol. 22, no. 10, October, 1984.

20. Hajela, P., C. Bloebaum, and J. Sobieszczanski-Sobieski. "Application of Global Sensitivity Equations in Multidisciplinary Aircraft Synthesis." Journal of Aircraft, vol. 27, no. 12, December, 1990.

21. Sobieszczanski-Sobieski, J. "Sensitivity Analysis and Multidisciplinary Optimization for Aircraft Design: Recent Advances and Results." Journal of Aircraft, vol. 27, no. 12, December, 1990.

22. Sobieszczanski-Sobieski, J. "A System Approach to Aircraft Optimization." NASA TM-104074, March, 1991.

23. Sobieszczanski-Sobieski, J. "Sensitivity of Complex, Internally Coupled Systems." AIAA Journal, vol. 28, no. 1, January, 1990.

24. Sobieszczanski-Sobieski, J. "On the Sensitivity of Complex, Internally Coupled Systems." AIAA 88-2378, April, 1988.

25. Bischof, C., et. al. "ADIFOR: Generating Derivative Codes from FORTRAN Programs." MCS-P263-0991, Mathematics and Computer Science Division, Argonne National Laboratory, IL, and Center for Research on Parallel Computation, Rice University, TX, 1991.

26. Braun, R. D., I. Kroo , and P. Gage. "Post-Optimality Analysis in Aerospace Vehicle Design." AIAA 93-3932, August, 1993.