

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.



A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600

MULTIDISCIPLINARY DESIGN USING COLLABORATIVE OPTIMIZATION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ian Patrick Sobieski

August, 1998

UMI Number: 9908849

UMI Microform 9908849
Copyright 1998, by UMI Company. All rights reserved.

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

© Copyright by Ian Patrick Sobieski 1998

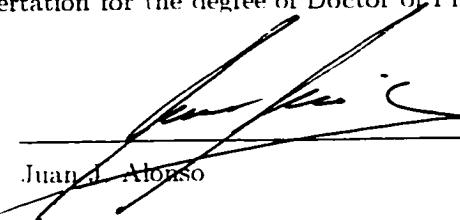
All Rights Reserved

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



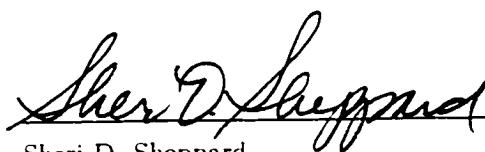
Ilan M. Kroo (Principal Advisor)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Juan J. Alonso

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.



Sheri D. Sheppard

Approved for the University Committee on Graduate Studies:



Thomas Wasow

Abstract

Management of the modern aircraft design process is a substantial challenge. Formal iterative optimization is commonly used with disciplinary design tools to aid designers in the definition of optimal subsystems. However, the expense in executing high fidelity analysis, the decomposition of the design expertise into disciplines, and the size of the design space, often precludes the use of direct optimization in the overall design process.

Collaborative optimization is a recently developed methodology that shows promise in enabling formal optimization of the overall design. The architecture preserves disciplinary design autonomy while providing a coordinating mechanism that leads to interdisciplinary agreement and improved designs. The basic formulation has been applied to a variety of sample design problems which demonstrate that the method successfully discovers correct optimal solutions.

This work places collaborative optimization in the context of other multidisciplinary design optimization methods and characterizes problems for which the basic formulation is applicable. Artifacts of the problem formulation are discussed and methods for handling high bandwidth coupling, such as that found in aeroelasticity, are presented.

The use of response surfaces for representing expensive analyses has become increasingly popular in design optimization. Response surfaces are smooth analytic functions that are inexpensive to evaluate and may be generated from data points obtained from the parallel execution of analyses. These properties motivate the introduction of response surfaces into collaborative optimization. Response surfaces have been previously used to model subproblem analyses and were generated just once. Here, approximate models are used to represent the subproblem optimization results, not the analysis, and are regenerated as the design is modified. The use of response surfaces in collaborative optimization requires an inexpensive method for generating the approximate models. Several approaches to generating quadratic response surfaces have been implemented and their relative merits are discussed. Special information inherent in each subproblem optimization is used to generate quadratic models requiring fewer data points than design variables.

The algorithms and techniques for using response surfaces in collaborative optimization have been developed on a series of problems that are currently of interest to industry. These include the specification of an ocean-going oil tanker, the design of a tailless unmanned aircraft, and the integration of industry standard computational tools in the design of a high speed civil transport.

Acknowledgments

In performing the research for this project I was always supported by the members of the Aircraft Design Group at Stanford: Mark Meyer, Mike Holden, Valerie Manning, Peter Gage, Bobby Braun, Steve Altus, Ben Tigner, Sean Wakayama, and David Rodriguez. I was truly fortunate to work in an environment where my close colleagues became my close friends. I am thankful for the editorial help of Peter Gage in reading copies of this manuscript during its development. I would also like to thank the rest of my dissertation committee, Profs. Ilan Kroo, Juan Alonso, Sheri Sheppard, and Thomas Kenny for their input to this final manuscript. I am grateful to Prof. Ilan Kroo, a superlative advisor and example, who made it possible for me to come to California and is responsible for the wonderful environment in which I had the privilege to work. I'd not have been here at all if it were not for the transmitted values, encouragement, example, love, and support provided by my parents and my sister, Margaret.

It has been a long time since I came to the Golden State to start work on this project. Whatever value I've provided here is small compared to that I received from my friends, both on and off the farm. To my friends—thank you for being part of my life.

This work was supported by NASA-Langley Research Center, Contract NAG1-1558 and NCC-1253, and by the Lockheed Corporation, Contract SNG30G4400R-1. I am not unmindful of the fact that academic research, such as this thesis, is a luxury that is easily taken for granted. I am sincerely grateful to the society I was fortunate to be born into and whose benefits I reap daily: a society that makes possible and supports splendid institutions like Stanford, and my undergraduate university, Virginia Tech.

Contents

Abstract	iv
Contents	viii
List of Tables	xiii
List of Figures	xiv
Chapter 1 Toward a New Design Process	1
1.1 The Design Process.....	3
1.2 The Distributed Design Environment.....	6
1.3 How the Design Proceeds.....	9
1.4 Designing the Design Process.....	11
1.5 Collaborative Optimization As a Design Architecture.....	12
Chapter 2 Multidisciplinary Design Optimization	15
2.1 Nomenclature	15
2.2 Approaches to MDO.....	16
2.2.1 Single Level Design Optimization.....	16
2.2.2 Design by Sequential Optimization.....	18
2.2.3 Optimizer-Based Decomposition (OBD).	20

2.2.4 Discipline-Feasible Constraint Methods	21
2.2.5 Collaborative Optimization	21
2.2.6 Summary	25
2.3 Wing Design Example	26
2.3.1 Design Problem Description	26
2.3.2 Single Level Implementation	28
2.3.3 Optimizer Based Decomposition (OBD) Implementation	29
2.3.4 Design by Sequential Optimization	30
2.3.5 Collaborative Optimization	31
2.3.6 Summary of Wing Design Results	33
2.4 Summary	37
Chapter 3 Performance Properties of Collaborative Optimization	38
3.1 Theoretical Estimate of Performance	38
3.1.1 Comparison Assumptions	39
3.1.2 Nomenclature	39
3.1.3 Single Level Optimization	41
3.1.4 Optimizer Based Decomposition (OBD)	41
3.1.5 Collaborative Optimization	42
3.2 Effect of Coupling Bandwidth on CO Applicability	44
3.2.1 Comparison of OBD and CO	45
3.2.2 Performance Insensitive to Coupling Strength	48
3.2.3 Summary of Appropriate Problem Qualities	49
3.3 Artifacts of Decomposition	50
3.4 Performance on Sample Problem	54
3.4.1 Description of the Aircraft Sizing Problem	55
3.4.2 Optimizer Based Decomposition Solution	56
3.4.3 Collaborative Optimization	57
3.4.4 Aircraft Design Results	58

3.4.5 Implementation Details Critical to Performance	61
3.4.6 Performance of Example Airplane Problem	63
3.4.7 Convergence Behavior	63
3.4.8 Effect of Increasing Subspace Dimensionality	65
3.5 Summary	66
Chapter 4 Reducing High Bandwidth Coupling	67
4.1 Special Case of High Bandwidth Coupling	67
4.2 Reduced Basis Modeling	68
4.2.1 Measure of Merit	69
4.2.2 Model Requirements Define Appropriate Basis Method.	69
4.2.3 Modeling a 2-D Pressure Distribution	71
4.2.4 Fitting the Spanwise Force Distribution	72
4.2.5 Fitting the Spanwise Moment Distribution	74
4.2.6 Incompatible Models of Torque, Force, and Moment	78
4.2.7 Optimal Fit with Assumed Force Functional Form	79
4.2.8 Coefficient Meaning	82
4.3 Summary of Approaches	82
Chapter 5 Using Response Surfaces in Collaborative Optimization	84
5.1 Response Surfaces in Design	84
5.2 Using Response Surfaces in CO	86
5.2.1 Modeling the Subproblem Design	87
5.2.2 Model Quality	88
5.2.3 Management of Response Surface Updating	90
5.2.4 Form of Objective Function	92
5.2.5 Theoretical Performance	93
5.2.6 Convergence Tolerance and RS Models	94
5.3 Ship Design Using Response Surfaces in Collaborative Design	96

5.3.1 Nomenclature	96
5.3.2 Tanker Design Problem	97
5.3.3 Basic Collaborative Implementation	98
5.3.4 Results Using Basic Collaborative Optimization	99
5.3.5 Ship Design with CO and Response Surfaces	101
5.4 Summary	104
Chapter 6 Response Surface Improvements	106
6.1 Modeling Optimal Subproblem Designs	106
6.2 Estimating the Subproblem Objective, J^*	108
6.3 Fit Using Post-Optimality Gradients	110
6.4 Modeling Subproblem Optimal Interdisciplinary Design Vector	113
6.5 Fit Using Implicit Extra-Point Information	115
6.6 Summary and Comparison of Fitting Approaches	117
Chapter 7 Application to Example Design Problems	120
7.1 Nomenclature	120
7.2 Tailless UAV	121
7.2.1 The Analyses	123
7.2.2 Single-Level Optimization Solution	125
7.2.3 Collaborative Implementation	125
7.2.4 Collaborative Results with Response Surfaces	127
7.3 HSCT Design	130
7.3.1 HSCT Aerodynamic Design Analysis	131
7.3.2 HSCT Structural Design Analysis	133
7.3.3 Collaborative Implementation with Response Surfaces	135
7.3.4 HSCT Design Results	137
7.4 Summary	140

Chapter 8 Conclusions and Development Directions	141
8.1 Summary and Conclusions	141
8.1.1 Review of Basic Formulation	141
8.1.2 Introduction of Response Surfaces	142
8.1.3 Response Surface Improvements.....	142
8.1.4 Applications.....	143
8.2 Design in the Collaborative Environment.....	144
8.3 Directions for Future Work.....	145
8.3.1 Trust Region Updates	146
8.3.2 Response Surface Construction	146
References	148

List of Tables

TABLE 2.1	Wing design problem fixed parameters.....	28
TABLE 3.1	Equally sized disciplinary design problem.....	46
TABLE 3.2	One large disciplinary design problem	47
TABLE 3.3	One large and two medium sized disciplines	48
TABLE 3.4	Effect of SQP implementation on CO efficiency	62
TABLE 3.5	Effect of optimality tolerances on derivative accuracy	62
TABLE 3.6	Computational expense of airplane design problem	63
TABLE 6.1	Number of points required for full quadratic fit.....	118
TABLE 6.2	Comparison of RS fit error on “performance” subproblem	119
TABLE 7.1	Fixed parameters in tailless UAV design problem	122
TABLE 7.2	Optimal values of design variables and computed parameters...	125
TABLE 7.3	HSCT baseline design.....	131

List of Figures

Figure 1.1	The Ryan NYP was designed and built in 3 months.....	1
Figure 1.2	The Boeing 777-300.....	2
Figure 1.3	Three view of HSCT concept	6
Figure 1.4	Disciplinary design dependency diagram	7
Figure 1.5	Macro-scale design dependency diagram	9
Figure 1.6	Design flowchart	10
Figure 2.1	Single-level approach to design optimization	17
Figure 2.2	Sequential optimization	19
Figure 2.3	Optimizer based decomposition (OBD) architecture.....	20
Figure 2.4	The basic collaborative optimization architecture.....	22
Figure 2.5	Wing design problem	27
Figure 2.6	Single-level optimization.....	28
Figure 2.7	Wing design by optimization based decomposition	29
Figure 2.8	Wing design by sequential optimization.....	31
Figure 2.9	Wing design by collaborative optimization.....	33
Figure 2.10	Optimal values of wing design parameters	34
Figure 2.11	Lift distribution.....	35
Figure 2.12	Wing design was executed collaboratively on three parallel processors	36
Figure 3.1	Single level optimization.....	40
Figure 3.2	Optimizer based decomposition.....	42
Figure 3.3	Collaborative implementation.....	43

Figure 3.4	Collaborative formulation for sample problem from [53].	48
Figure 3.5	Decomposition can lead to poor search directions.	51
Figure 3.6	Solution to single level problem with two constraints is local optimum.	52
Figure 3.7	Decomposition of constraints allows movement toward global optimum.	53
Figure 3.8	CO finds a better design point for Eqn. 3-2 than single level optimization	54
Figure 3.9	Aircraft sizing problem.	55
Figure 3.10	OBD implementation of aircraft sizing problem	57
Figure 3.11	Collaborative solution of aircraft sizing problem.	58
Figure 3.12	Variation of wing area in CO and OBD design cycles	59
Figure 3.13	Variation of aspect ratio in CO and OBD design cycles.	59
Figure 3.14	Design Variables and Computed Results	60
Figure 3.15	Spanwise thickness distribution for OBD and CO solutions	61
Figure 3.16	Variation in objective function and discrepancy constraints with major iteration.	64
Figure 3.17	Variation in number of major iterations with local variables.	65
Figure 4.1	A502 Panel Model of Wing.	70
Figure 4.2	FESMDO structural model of generic wing	70
Figure 4.3	Lift distribution modeled with two Fourier coefficients	72
Figure 4.4	Approximated force and resulting integrated moment based on two Fourier coefficient model of lift distribution from 2-D panel code	73
Figure 4.5	Cubic polynomial model of spanwise moment distribution and corresponding force distribution (d^2F/dx^2) and actual moment and force distribution	75
Figure 4.6	Spline model of spanwise moment distribution and corresponding force distribution (d^2F/dx^2) and actual moment and force	77
Figure 4.7	Monotonic spline model of spanwise moment distribution and corresponding force distribution (d^2F/dx^2) and actual moment and force	78

Figure 4.8	Piecewise linear model of force to satisfy Eqn. 4-2 and actual moment and force	80
Figure 4.9	Fourier model of force to satisfy Eqn. 4-2 and actual moment and force.	81
Figure 4.10	Normalized sum squared error of reduced basis strategies for given loading.	83
Figure 5.1	(a) Standard collaborative implementation, (b) RS models analysis, (c) RS models optimization	88
Figure 5.2	Variation in subproblem function and quadratic response surface model, from ship problem (Section 5.3) propulsive design.	89
Figure 5.3	Collaborative optimization with response surface estimation	94
Figure 5.4	Variation in response surface shape as a function of subproblem optimization convergence tolerance	95
Figure 5.5	Number of analysis executions as a function of subproblem optimization convergence tolerance	96
Figure 5.6	Conceptual ship design problem	97
Figure 5.7	Conceptual ship design analysis flow	98
Figure 5.8	Standard collaborative implementation of ship design problem	99
Figure 5.9	System level objective function, ROI, as a function of major iteration	100
Figure 5.10	Compatibility constraint value as a function of major iteration	101
Figure 5.11	Use of response surfaces to represent ship optimal subproblem designs	101
Figure 5.12	Target ROI vs. optimization cycle for tanker design using CO and response surfaces with trust region updates.	102
Figure 5.13	Compatibility constraint value versus RS update cycle for trust region using RS approximation	103
Figure 5.14	Initial and final design point for tanker design with collaborative optimization using response surfaces matches result from single level optimization	104
Figure 6.1	Value of subproblem optimal objective function J^* , Eqn. 6-1, vs. W_o and R_o	108
Figure 6.2	RS fit of J^* based on points a, b, c, d, e, and f from Figure 6.1	110

Figure 6.3	RS fit of J^* based on evaluations and gradients at points a, c, and e from Figure 6.1	112
Figure 6.4	RS fit of J^* based on evaluations and gradients, but no redundant information, at points a, c, and e from Figure 6.1	113
Figure 6.5	RS fit of J^* based fitting the optimal subproblem design vectors $\{x^*\}$ and $\{y^*\}$ evaluated at points a, c and e from Figure 6.1	115
Figure 6.6	RS fit of J^* based fitting the optimal subproblem design vectors $\{x^*\}$ and $\{y^*\}$ evaluated only at points a and e from Figure 6.1	117
Figure 7.1	Design variables defining tailless UAV	121
Figure 7.2	Structure of analyses in tailless UAV problem	123
Figure 7.3	Convergence behavior of basic collaborative optimization implementation	127
Figure 7.4	Range vs. trust region cycle	128
Figure 7.5	Optimal subproblem objective function values vs. trust region cycle	128
Figure 7.6	Spanwise twist vs. trust region cycle	129
Figure 7.7	Section spanwise C_l distribution	129
Figure 7.8	HSCT top view	130
Figure 7.9	A502 wing panel model and computed upper surface pressure distribution	132
Figure 7.10	Computed results and design variables for each of the disciplinary design problems	133
Figure 7.11	FESMDO FEM structural representation of HSCT wing	134
Figure 7.12	Use of response surfaces in HSCT CO design problem	136
Figure 7.13	Compatibility constraint value with design cycle	137
Figure 7.14	Initial cruise altitude from the three subproblems	138
Figure 7.15	Takeoff gross weight (TOGW) from the three subproblems	138
Figure 7.16	Initial and final upper surface pressure distributions	139

Chapter 1

Toward a New Design Process

On February 23, 1927, Charles Lindbergh walked into Ryan Airlines, Inc. of San Diego, California and ordered an airplane. He, chief designer Donald Hall, and the construction foreman, Claude Ryan, sketched out the design of a new aircraft in 3 days. On May 21, 1927, the Ryan NYP, dubbed the Spirit of St. Louis, took off for Paris and the history books^[1].

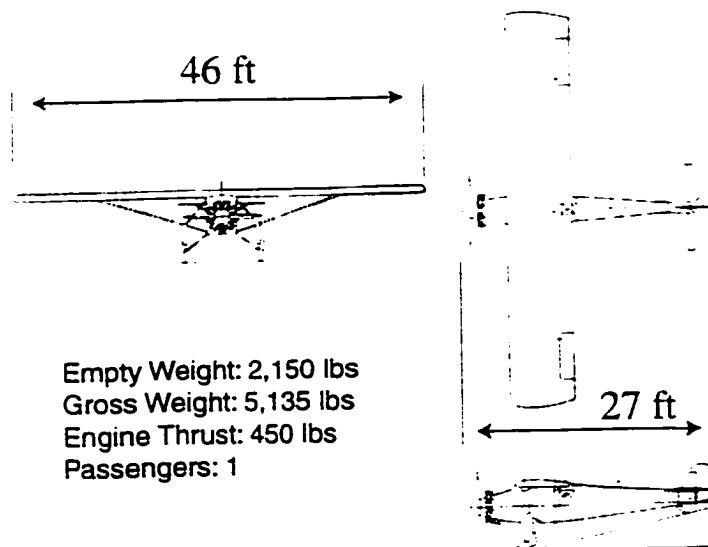


Figure 1.1 The Ryan NYP was designed and built in 3 months

On October 16, 1990, United Airlines formally requested that the Boeing Airplane Company build a new widebody transport. Using advanced design techniques

and automation. more than 10,000 engineers spent 44 months designing United Airlines flight 921, a Boeing 777 that left Heathrow Airport on June 7, 1995^[2].

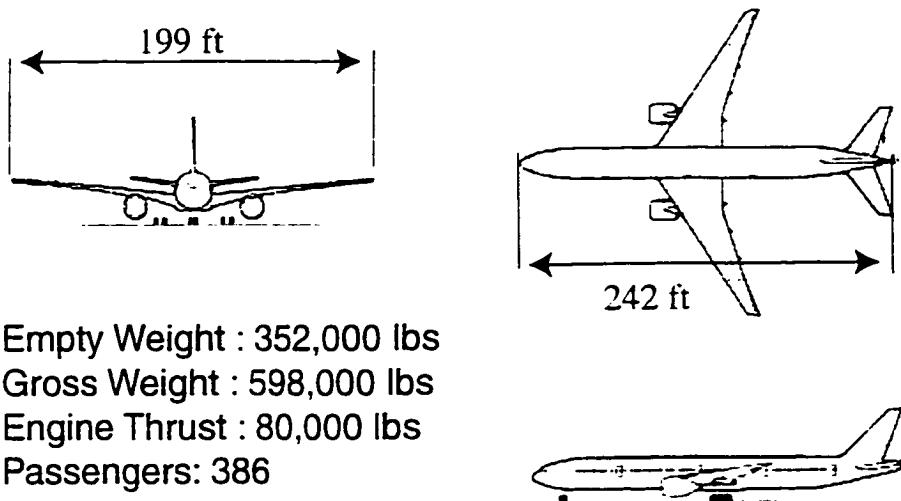


Figure 1.2 The Boeing 777-300

The 777 is much more complex than the Ryan NYP; and so is the design process that created it. This increase in complexity has been driven by two forces:

1. the need to manage the millions of physical pieces that make up a complex machine like an airplane, and to anticipate the interaction of these pieces.
2. the advent of exacting, but computationally expensive, analysis techniques for predicting physical behavior and the desire to exploit this ability to improve the design.

Components in a complex machine can interact in unanticipated, and sometimes embarrassing, ways. The first model of the Chevy Monza was released to consumers in 1975. Only then was it discovered that engine compartment clearance in the Chevy prevented the spark plugs from being changed. Access to the plugs was only possible by releasing the engine mounts and hoisting the block a few inches out of the compartment^[3].

To avoid similar difficulties in its new aircraft development, Boeing brought to the 777 design an advanced modeling system called CATIA, for Computer graphics Aided Three dimensional Interactive Application. Each part and system was scrupulously modeled in a computer database allowing anyone to visualize how their system affected the systems of everyone else. This design management tool was able to alert engineers to the fact that the original placement of the 10 foot diameter engine blocked access to the fuel port on the underside of the wing. Previously, this sort of system interaction was only caught on the machine shop floor.

But clearance, accessibility, and physical interference is only the simplest and most straight forward kind of interaction between components. The physics of different systems of a complex machine can adversely or positively affect other components. Many of these interactions have been well understood for a long time. After all, Ryan, Hall, and Lindbergh put a horizontal tail on the Ryan NYP because they knew it needed one for trimmed flight. Other interactions, such as the propeller whirl-mode flutter on the Lockheed Electra, were discovered only after disaster^[4]. The development of physically sophisticated models has allowed for exacting prediction of the behavior of aircraft systems but has also driven distributed specialization of expertise. The small team that built the NYP has been replaced with literally thousands of designers grouped together in fields of specialization. Within these disciplinary groupings reside knowledge and computational resources that allow designers to thoroughly model the physical performance of aircraft systems prior to construction. And while technology such as CATIA has been incorporated into the management of the physical components of an aircraft, the design process that defines the shape of those systems has changed only by becoming larger and more difficult to manage.

1.1 The Design Process

The design of complex machines, such as aircraft, is traditionally divided into three phases, conceptual, preliminary, and detailed^{[5][9]}.

During the conceptual design phase the major characteristics of the aircraft are open for definition. The aircraft may have a delta or simply swept wing, with two or four aft-mounted or under-wing engines, and a conventional or v-tail empennage. These decisions are often made by senior designers, on the basis of accumulated experience, very preliminary analysis, and market requirements. Analysis tends to be limited to parametric studies based on quick running, low fidelity models of the relevant physics. At the end of this stage the rough outline of the aircraft is fixed. In the case of the 777, the conceptual design was completed on the basis of customer feedback from the 747 and 767, and so the conceptual design process was completed before the final decision to develop the airplane was even made.

Preliminary design involves many substages of development. At the beginning of the preliminary design phase only the rough outline of the aircraft is fixed. By the end of the process all the most influential design variable values, in terms of overall performance, are specified. In between, the disciplinary teams interact to progressively specify the design through trade studies and the use of increasingly sophisticated analysis tools—eventually including CFD and FEM analysis, wind tunnel tests, and mock-up construction.

A tool such as CATIA is primarily used during the detailed design phase. The shapes of door latches, the path of ventilation ducts and electrical conduits, the details of the overhead luggage rack, are all specified during this phase. The variables that have a first order effect on the airplane performance however have already been fixed, at the end of the preliminary design process or at the beginning of the detailed phase.

The greatest freedom to exploit potential trade-offs between aircraft subsystems for the optimization of the design occurs earliest in the design process. However, the capacity to utilize this freedom is limited by the fact that the analysis tools brought to bear at this stage are the most limited^[6]. More detailed analysis can contradict the findings of earlier, simpler, analysis. Ideally, detailed analysis should be used as early in the design process as possible.

Working against this goal is the difficulty in managing the enormous amount of detail used to describe each of the various systems of a complex machine like an aircraft. These details can not be simply ignored and yet the controlling authority of the preliminary design process cannot explicitly coordinate thousands of design variables. The preliminary design of the 777 was performed by 3000 people. But most of the crucial design compromises were negotiated by 25 people who gathered weekly in a room in Renton, Washington. Each of them, representing 100+ engineers working in their own specialty, got two hours, every two and a half months, to present their group's work^[2]. The capacity for making detailed physical trades in this environment is severely limited.

The 777 could be successfully designed in this way because there is 40 years of industry experience developing aircraft that look like the 777. The interaction of different systems is relatively well understood and statistical data exist that can be leaned on heavily in making preliminary trade-offs without resorting to highly detailed models of the aircraft^[9]. However, for new aircraft, such as the Mach 2.4 High Speed Civil Transport, no similar industry experience exists and unexpected trade-offs may have an important effect on vehicle performance.

Direct optimization techniques have long been used within disciplinary teams to improve disciplinary designs. However, for a variety of reasons including the distributed design environment, described in the next section, and the large number of variables that define the design space, the direct implementation of optimization to the overall design process has never been successfully accomplished. Collaborative optimization is a design architecture that was developed with the requirements of real world design applications in mind. The resulting methodology maps easily onto the current design process structure.

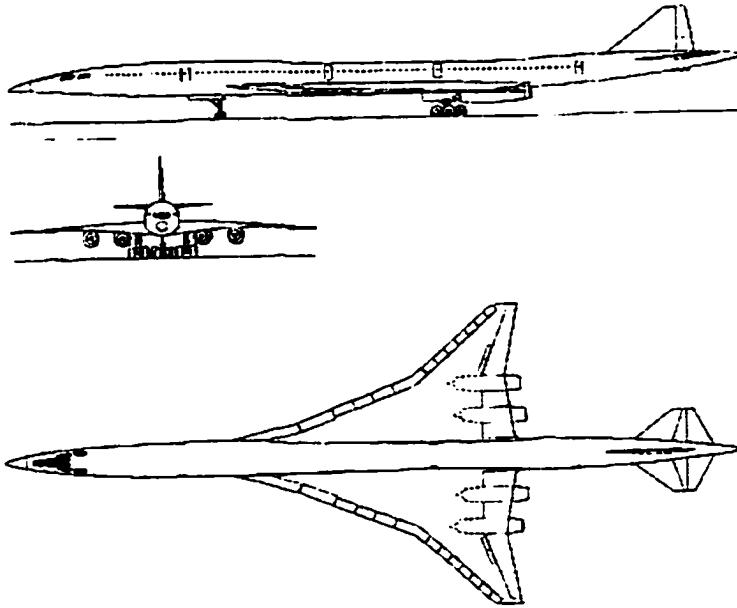


Figure 1.3 Three view of HSCT concept

1.2 The Distributed Design Environment

Complex design environments are not limited to aircraft design. For example, consider the design of the Mars 2001 probe. The goal of this government sponsored project is to build an unmanned probe to land on Mars. The conceptual design was already completed by October of 1997. Starting in October, a six month effort further refined the baseline design. Many parameters, including the Mars entry trajectory, heat shield thickness, equipment packaging, and total vehicle mass, were determined during this phase by individuals at NASA-Langley, NASA-Ames, Lockheed Martin, and the Jet Propulsion Laboratory.

The Mars 2001 design team consisted of approximately 100 people, working at four geographically separate locations, performing tasks from project management to atmospheric reentry analysis. There were thirteen distinct teams of individuals working with discipline specific analysis tools. Figure 1.4 illustrates the dependency

of these teams on each other in a design structure matrix. In this schematic, each box represents an analysis team. Input values enter the box from above and below and outputs exit to the left and right. A dot on a line indicates that some information is passed from one discipline to another. If the analyses are executed in the order they appear on the diagonal, dots above the diagonal represent feed forward connections while dots below the diagonal represent feedback.

In practice, the analyses of Figure 1.4 are considered part of larger disciplinary design teams. A pair of analyses may be part of a particular design team because they iterate closely with each other (aerodynamics & trajectory). They require similar disciplinary expertise (electronics & communication). The same people are involved in different tasks (structures and packaging). They are geographically proximal, or for organizational reasons (mission design and payload packaging). In this design effort there were three distinct disciplinary design problems: Atmospheric Flight Dynamics, Thermal Protection System sizing, and Vehicle Design.

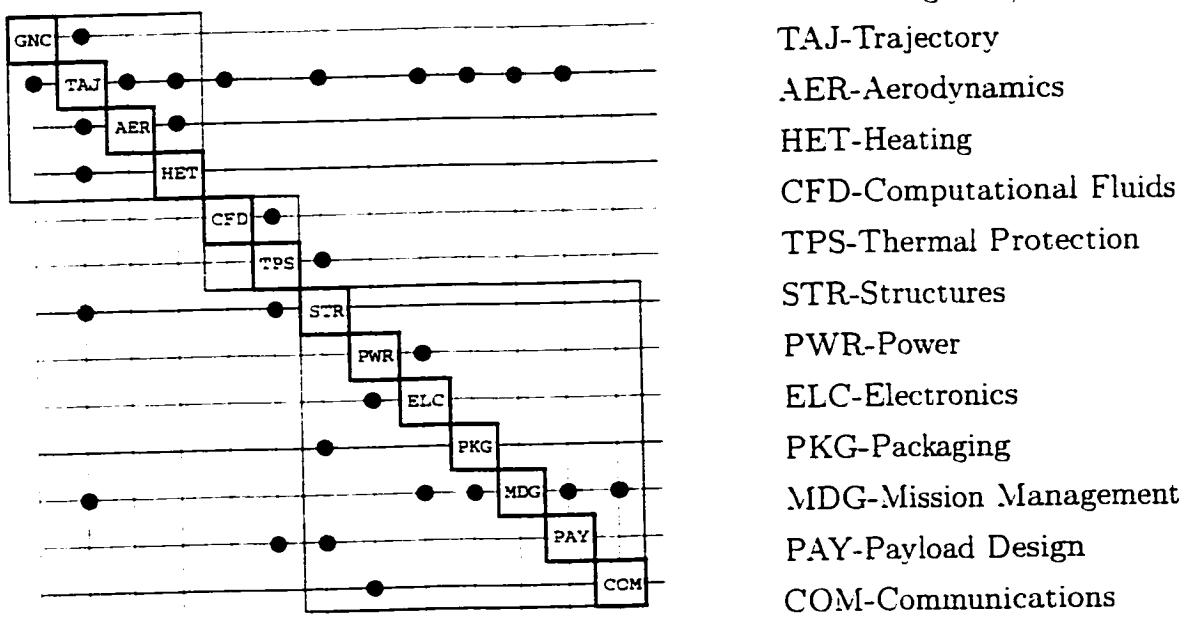


Figure 1.4 Disciplinary design dependency diagram

The Atmospheric Flight Dynamics discipline combines aerodynamics, dynamics, and controls analysis to predict and design the Mars probe entry trajectory. Using a specified entry state vector, desired landing location, the vehicle center of gravity location, mass, moment of inertia, and thruster location and power, the AFD team runs numerical simulations of the vehicle aerodynamics and dynamics to compute the trajectory and resulting deceleration, vibration, ΔV requirements, and target uncertainties. The local design effort is directed toward development of the guidance control algorithm. The people performing this analysis are located at JPL, NASA-Ames, NASA-Johnson, and Lockheed/Martin.

The Thermal Protection System (TPS) discipline designs the thermal shell for the vehicle that protects it during Mars entry. Points along the trajectory, computed by the AFD team, are evaluated using the Navier Stokes equations using CFD. The TPS team is restricted to designing the constant thickness TPS shell that satisfies payload temperature, structure temperature, and TPS material constraints. Manufacturing constraints specify the TPS shell material and specify the thickness distribution. TPS sizing work is being performed both at Lockheed/Martin and NASA-Ames.

Finally, the Vehicle Layout discipline defines the details of the internal configuration of the vehicle (i.e. position of equipment, structural design, electronic equipment, etc.). Also included in the vehicle layout problem is the design of the communications system, power supply, payload thermal protection, and maneuver propulsion system. This design problem, in itself, involves scores of design variables and constraints. Much of this effort is the joint responsibility of the JPL project management office and the people at Lockheed/Martin in Colorado.

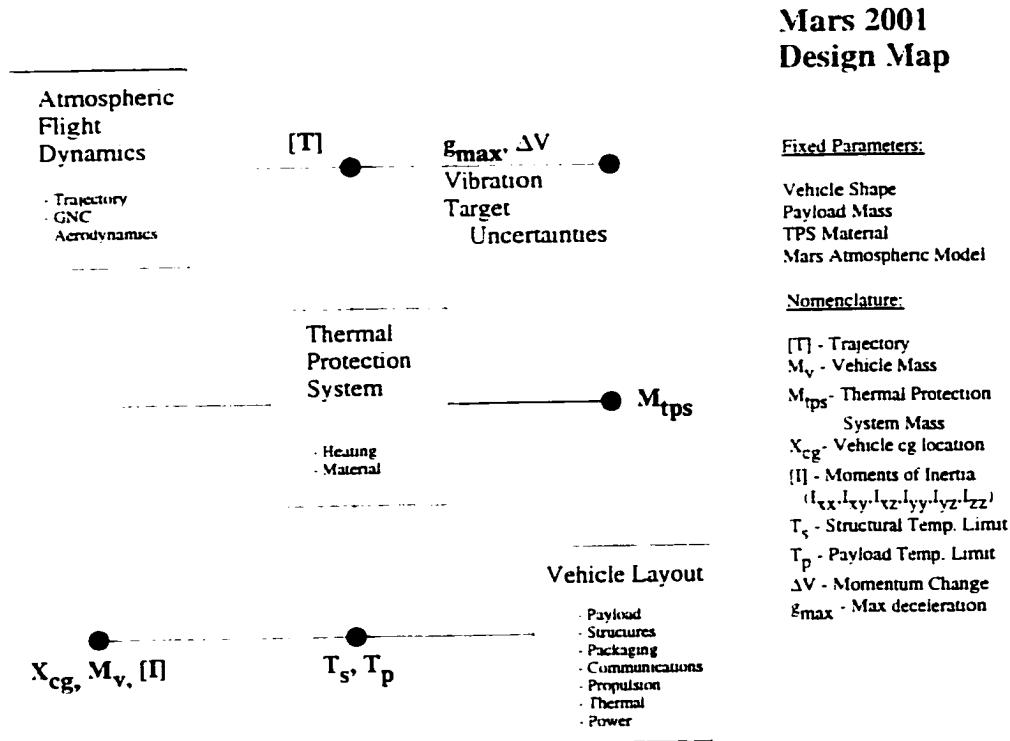


Figure 1.5 Macro-scale design dependency diagram

1.3 How the Design Proceeds

The design map shown in Figure 1.4 is fundamentally sequential and iterative. In practice however these disciplinary teams work in parallel. Iterative dependence is removed by specifying the value of the interdisciplinary variables. The design cycle follows the flowchart shown in Figure 1.6.

For example, the baseline design from the pre-phase A design cycle specifies the initial interdisciplinary variable values for the phase A design cycle. As the details of the subproblem designs are refined the previously feasible design may become infeasible and require modification of the interdisciplinary design variable values.

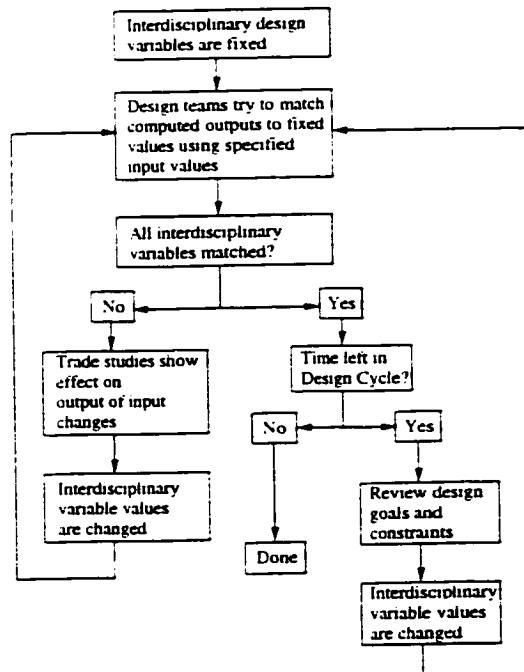


Figure 1.6 Design flowchart

The decision making structure for variable value changes is *ad hoc*. Sometimes it is based on trade studies, sometimes experience, sometimes it is arbitrary. These decisions are made through weekly teleconference calls, much like the preliminary design meetings of the 777 described in Section 1.3.

Additionally, in the current design approach the design tasks of the disciplines are not fixed. For one period of time AFD may calculate trajectories and g loadings and the structures people will compensate by modifying their design. At other times the controlling authority may fix the 'g' loading maximums and the AFD team is tasked with modifying their trajectories to meet these constraints. When and how the sub-design tasks are formulated is also an *ad hoc* decision.

1.4 Designing the Design Process

The problems of the previous section are not limited to aircraft design. As the example of the Chevy Monza illustrates, even automobiles have become complex enough to require the need for sophisticated design tools^{[10][11]}. Similarly, the integrated circuit market^{[12][13][14]}, naval architecture^{[15][16]}, and launch vehicle design^{[20][35][63]}, are all areas of where the demands for design management have motivated the development of design process design.

Since the mid-80s researchers have been developing techniques and building architectures for all phases of the design process. Each phase has different requirements for design development. In conceptual design, where discrete choices and large design spaces are typical, integrated visualization and rapid trade-study software^{[5][21][32]} and genetic search algorithms^{[18][19][20]} have found useful application. For detailed design, where the primary need is to manage the interactions of a large number of distinct physical design components, and to deal with issues of manufacturability, accessibility, and clearance, tools such as CATIA have been developed^[2].

Concurrent engineering approaches aid in the design process at all levels. Research in this area has focussed on aiding design teams in incorporating nontraditional design concerns, such as manufacturability, environmental impact, and aesthetics, as well as the more traditional goals of interdisciplinary dispute resolution and optimization^{[22][23]}. Frameworks for facilitating team design have been developed by a number of researchers in different fields. Next-Cut provides a coordinating data base and intelligent agent system for highlighting incompatibilities between different disciplinary subsystems and manufacturability requirements^[24]. A key feature of this approach is the directed invocation of analysis processes based on the requirement for information. This feature is also present in the quasi-procedural method that underlies the preliminary aircraft design tool PASS where it has been shown to substantially reduce computational cost and design cycle times^[40].

Alternate frameworks have been directed toward improving the preliminary design structure. Simply mapping the design process with a design dependency

matrix, such as Figure 1.4, allows for the intelligent scheduling of analyses to remove unnecessary feedback and iteration^{[7][8][9]}. Examples of design scheduling tools include AGENDA^[26] and DeMAiD^[27]. Tools for automatic differentiation, such as ADIFOR, have enabled complex analytic codes to provide analytically computed gradients for sensitivity analysis^[28]. However, as described earlier, in preliminary design the central challenge is performing design optimization in a distributed environment made up of distinct disciplinary design teams with individual solution strategies, locally defined variables and constraints, potentially costly computational analyses, and interdisciplinary coupling. The nature of this sort of design environment precludes the use of direct iterative optimization methods, such as those used in many local subsystem design problems.

In response to this a variety of multidisciplinary optimization methods have been developed that attempt to enable a more formalized design optimization process in the preliminary design phase. These are related to one another in their use of a coordinating optimization problem to make progress toward optimal designs. The methods differ from each other in how they handle local feasibility, interdisciplinary compatibility, and local design autonomy. Chapter 2 places collaborative optimization in the context of other methods for performing multidisciplinary design optimization.

1.5 Collaborative Optimization As a Design Architecture

Collaborative optimization provides a formulation of the subproblem design task and a rational way of guiding the changes in the variables used to describe the design. Its form is compatible with the current design structure in industry and thus can be readily implemented in real-world design environments.

The current design process, shown in Figure 1.5 and Figure 1.6, is very similar to collaborative optimization. However, the current approach is informal and *ad hoc*. Collaborative optimization formalizes this process by doing three things:

1. Makes the local designer's goal the minimization of the discrepancy between local values of interdisciplinary variables and the value specified by the controlling authority, while generating locally feasible designs.
2. Gives local control of input variable values to the disciplinary design team.
3. Provides the system designer with information on how to change the interdisciplinary variable values.

The basic form of collaborative optimization has been shown to obtain the correct optimum for a variety of design problems^{[54][65]}. In this thesis five design problems are solved, adding to the breadth of problems successfully solved using the basic formulation of collaborative optimization. In these problems, previous observations concerning sensitivity of performance to implementation details, coupling breadth and strength, and problem decomposition are confirmed and elaborated upon. These results and observations are summarized in Chapters 2 and 3. Since each evaluation of the subproblem design is a complete iterative optimization, the basic formulation places a heavy emphasis on minimizing the number of interdisciplinary design variables and using a fast converging optimizer at the system level. Chapter 4 demonstrates an approach to handling high bandwidth coupling that reduces the number of interdisciplinary variables. Chapter 5 introduces a modification to the basic architecture that uses response surfaces to reduce computational cost by incorporating approximate models of the subproblem design results. The system level optimization problem uses the response surfaces, in lieu of executing the subproblem optimizations, to predict the effect of changing target variable values. A design point is specified, based on the models, about which a new set of response surfaces is generated. The size and shape of the response surface domain is governed by an algorithm using ideas from trust region update methods. This thesis demonstrates the first use of response surfaces within the collaborative optimization architecture and the use of response surfaces to represent design results, rather than analyses.

A challenge in the use of response surfaces in general is that even a model of quadratic order requires on $O(n^2)$ function evaluations to solve for the unknown model coefficients. In collaborative optimization, where it is expected that several response surfaces will be generated during the course of the overall system optimiza-

tion, this challenge is even more acute. Chapter 6 shows how a quadratic model for a collaborative subproblem may be generated in $O(n)$ subproblem design optimizations and how special information inherent in each subproblem optimization may be exploited to reduce the number of required subproblem optimizations by a further 50%. These techniques enable the practical application of response surfaces within the collaborative optimization architecture.

The modified collaborative optimization algorithm with response surface estimation is used in Chapter 7 for the design of a tailless UAV and a Mach 2.5 HSCT. In Chapter 6 a similar algorithm is used to specify the optimal design for an ocean-going oil tanker. These results demonstrate the successful implementation and performance advantages of the modified method and motivate its continuing development and application in an industrial setting.

Chapter 2

Multidisciplinary Design Optimization

The previous chapter discussed the need for design optimization methods tailored to the distributed disciplinary specialization found in modern design. Collaborative optimization is only one of the newest approaches addressing this need. In this chapter, several other MDO methods are described and a relatively simple multidisciplinary design problem is solved using a few of these methods. The results illustrate the advantages and disadvantages of the various approaches and emphasize the importance of correctly accounting for cross-disciplinary effects. This chapter demonstrates the implementation of collaborative optimization and the following chapter examines its applicability, performance, and convergence behavior.

2.1 Nomenclature

Min.	=	Minimize
w.r.t.	=	With respect to design variables
s.t.	=	Subject to constraints
$\{x_g\}$	=	Vector of global variables
$\{x_g\}_i$	=	Portion of global variable vector used by i^{th} analysis
$\{x_l\}$	=	Vector of all strictly local variables
$\{x_l\}_i$	=	Variables used only by the i^{th} analysis
$\{y\}_i$	=	Parameters computed by the i^{th} analysis
$\{x_{\text{aux}}\}$	=	Vector of auxiliary variables

$\{x_{aux}\}_i$	Portion of auxiliary variable vector used by i^{th} analysis
$\{z\}$	System level target variable
θ_i	Incidence at i^{th} spanwise station
τ_i	Spar thickness at i^{th} spanwise station
b	Span
L/D	Lift to drag ratio
R	Range
w_{total}	Total weight
w_{fixed}	Fixed weight
w_s	Spar weight
AR	Aspect ratio
t/c	Thickness to chord ratio
ρ	Structural material density
σ_{max}	Maximum allowable stress
{l}	Lift distribution
$g(\{x_{aux}\}_i, \{y\}_i)$	= Compatibility constraint between $\{x_{aux}\}$ and $\{y\}$
J	Collaborative optimization subproblem objective function
\cdot	Superscript identifies OBD auxiliary variable
$_o$	Subscript identifies CO target variable

2.2 Approaches to MDO

2.2.1 Single Level Design Optimization

The most straightforward use of optimization in design is to combine the disciplinary analyses, such as those shown in Figure 1.4, into a single integrated analysis. This integrated analysis becomes a black box, a computational process that requires certain inputs and produces computed results. Some of these results are constrained,

others comprise the merit function to be improved. Interdisciplinary compatibility is accomplished within the integrated set of analyses, typically through iteration.

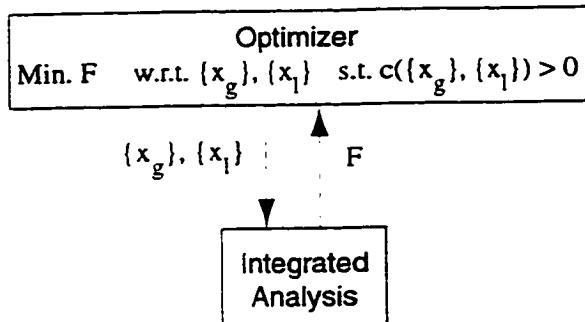


Figure 2.1 Single-level approach to design optimization

This approach is conceptually very simple and analogous to the way in which optimization is applied within disciplines to their analysis tools. However, this architecture removes all design authority from the disciplines and concentrates that authority with the system optimizer. This presents potentially serious implementation problems.

First, in detailed design problems the number of local design variables can be very large. The single level approach requires the system level controlling authority to make decisions concerning the value of every variable that defines the object. Similarly, every computed constraint must be satisfied by the optimizer. This is a huge analysis problem. For example, a single level optimization of an aircraft design may require the controlling authority to specify the thicknesses of hundreds of stringers, the x, y, and z ordinates of spanwise airfoil sections, and satisfaction of climb constraints and accessibility to cargo doors for luggage handlers.

Second, parallel execution of analyses is not possible. Since interdisciplinary compatibility is obtained through iteration, each analysis block in the process must await the solution of earlier analyses before it may proceed.

Third, for many physical systems, fixed point iteration, to obtain interdisciplinary compatibility within the integrated analysis, is divergent. In addition, iteration that includes complex analyses can easily become excessively time consuming.

Fourth, gradient information becomes useless unless it is available from each disciplinary analysis. A disciplinary analysis may be able to provide first order information about its computed results; but this information will not help the system level optimizer define a new search direction unless gradient information is available from each disciplinary analysis.

And finally, local solution strategies cannot be integrated. Because each discipline performs only analysis, attempts to use locally appropriate design strategies, such as an inverse design or a genetic algorithm, are not able to be accommodated within the rules of the architecture.

2.2.2 Design by Sequential Optimization

Sequential optimization is an extension of the fundamental single level optimization approach that provides a greater degree of disciplinary autonomy and reduces the communications requirements between disciplines. In sequential optimization each discipline attempts to develop a local design that improves a locally defined merit function. Usually this merit function is a quantity computed by the discipline that is thought to have an understood beneficial effect on the overall system objective.

This approach recognizes that most of the variables that define a complex design are strictly local to a discipline. This is, in fact, one of the characteristics that define disciplinary groupings. In sequential optimization each discipline has sole control over their strictly local variables and constraints. By adjusting their local variables they try to satisfy these constraints. The approach also facilitates local solution strategies and allows disciplinary designers to decide on local measures of merit.

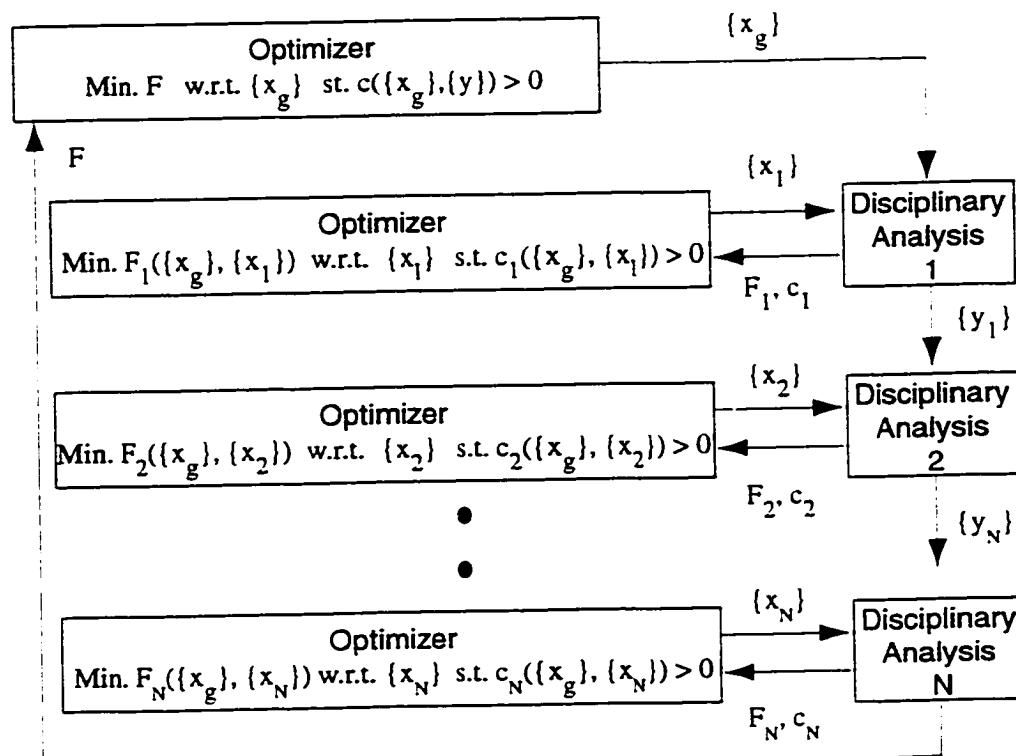


Figure 2.2 Sequential optimization

The method still suffers from several of the drawbacks of single level optimization. Like single level optimization, it is fundamentally serial in execution requiring earlier disciplinary optimizations to converge before subsequent disciplines can begin their design analysis. Also, interdisciplinary compatibility is enforced by iteration and, just as in single level optimization, this can easily become computationally prohibitively expensive and non-convergent.

Finally, as will be seen in Section 2.3.4, definition of the merit function in a disciplinary design is not always obvious. Except in special problems, this approach will lead to incorrect solutions.

2.2.3 Optimizer-Based Decomposition (OBD)

Research in decomposition analysis has led to a class of alternative formulations known by several names: simultaneous analysis and design^{[41][42]}, all-at-once^[43], or optimizer-based decomposition^{[44][26][45]}. As sketched in Figure 2.3, this formulation allows the N analysis-blocks to be executed in parallel.

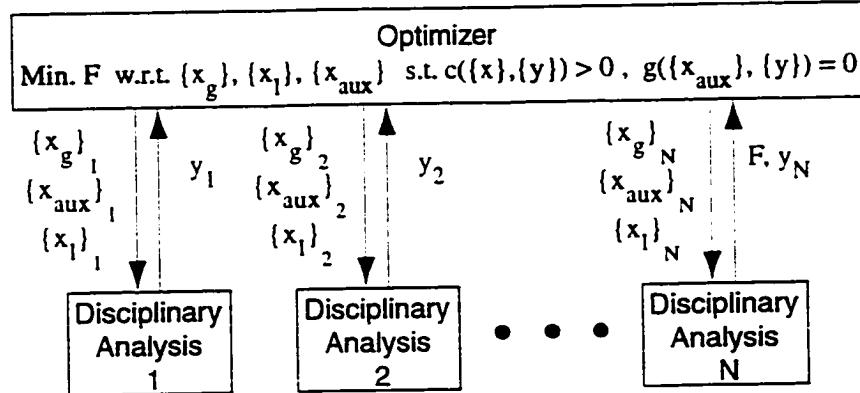


Figure 2.3 Optimizer based decomposition (OBD) architecture.

Here, each analysis-block is responsible for evaluating its own set of the originally partitioned constraints, c_i . For each interdisciplinarity variable, $\{x_{aux}\}$, shown in Figure 2.3, an equality constraint and a design variable is added to the optimization problem set^[26]. When satisfied, these equality constraints, g , require that the value of a variable computed in analysis block j match the value of the equivalent variable input to analysis block i . Within the optimization problem, these coupling variables, $\{x_{aux}\}$, are included in the design variable vector along with both the disciplinary, $\{x_l\}$, and interdisciplinary inputs, $\{x_g\}$. In comparison to the standard formulation, the requirement of producing a compatible multidisciplinary model is removed from the analyses, becoming an additional task of the optimizer. In this way, interdisciplinarity feasibility is only required at the solution, where $g_i=0$. In some cases, these formulative changes have been shown to produce computational savings through the removal of implicit iteration loops from the original analysis block^{[18][46][47][48]}.

The approach of Figure 2.3 retains some of the single-level approaches' shortcomings. The analysis groups are removed from the design decision-process, acting

simply as function evaluators. As with the single-level approach, the optimizer is responsible for the satisfaction of all design decisions regardless of how local they are. The number of constraints in this problem is increased by the degree of the interdisciplinary coupling.

2.2.4 Discipline-Feasible Constraint Methods

In disciplinary feasible constraint methods the individual analysis blocks are required to return a design candidate that satisfies the disciplinary governing equations as well as the domain-specific constraints but need not be interdisciplinary compatible. A system-level optimizer is utilized to ensure interdisciplinary compatibility at the overall solution and minimize the objective function.

Concurrent Subspace Optimization (CSSO) is an example of a discipline feasible constraint method, introduced in [49] and modified in [50], [51], [52], and [53]. Multiple subspace optimization problems are driven by a system-level optimizer which insures that, in the end, all subproblem constraints are satisfied. Interdisciplinary variable values are fixed by the system level optimizer but the value of computed results and constraints is approximated by each subproblem and used in defining local designs that both satisfy the local governing equations and reduce the residual of constraint violations. Responsibility for ultimate satisfaction of local constraints is partitioned among the subproblems. This approach has been successfully applied to numerous design problems^{[52][53]}, and convergence difficulties from [53] have led to improvements reported in [87].

2.2.5 Collaborative Optimization

Collaborative optimization is another discipline feasible constraint method that promotes disciplinary autonomy while providing a coordinating mechanism for obtaining interdisciplinary compatibility^{[35][36][37]}. The natural decomposition of the design problem along disciplinary boundaries, as shown in Figure 2.4, is preserved. As with CSSO, interdisciplinary incompatibilities may exist between computed results, but CO also allows incompatibilities to exist between different local values

of the same interdisciplinary variable. A principal feature of the collaborative optimization approach is that the subproblems are not required to use a single value for shared interdisciplinary variables. Each group is given control over its own set of local design variables and is charged with satisfying its own domain-specific constraints.

Interdisciplinary compatibility is achieved through a system level optimizer that specifies a set of target values for each interdisciplinary variable. The subproblems are executed as optimization problems with the goal of minimizing the discrepancy between their parameters and these system level specified target values. The system level optimizer's goal is to minimize an objective function of the target variables while satisfying its compatibility constraints. The compatibility constraints are the same function that each subproblem design is tasked with minimizing.

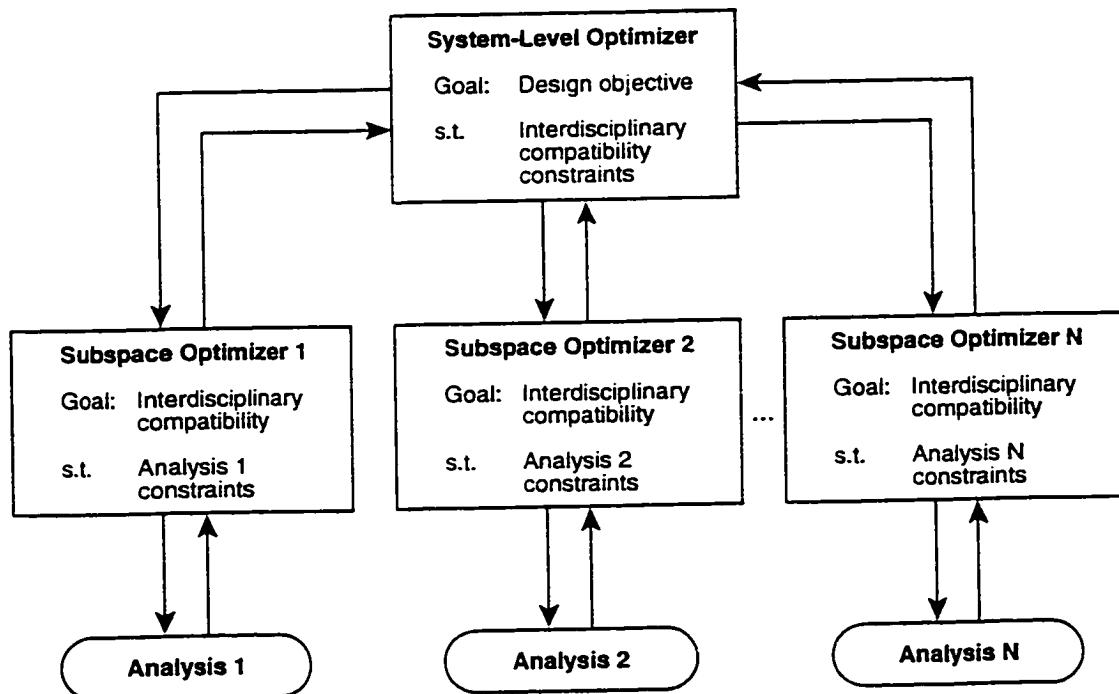


Figure 2.4 The basic collaborative optimization architecture.

Collaborative optimization keeps information relevant to a single subproblem at that subproblem level. Not all variables and computed results in a design problem need a system level target—only those parameters that are explicitly shared. In practice many variables are needed by only a single discipline. These variables are strictly local and do not need to be passed to the system level problem or other subproblems. This reduces the amount of required information transfer between the system level and subproblem level. For instance, the GSE method^[39] requires subproblems to report to the system level optimizer the gradients of all local constraints and parameters with respect to all design variables. In CO, local gradients are not required by the system level. This reduces the amount of communication required to the set of global design variables and the compatibility constraint values.

Although the system level compatibility constraints must eventually be reduced to zero, they may be violated during the course of the optimization. Non-zero compatibility constraints imply a discrepancy between the values used by different subproblems for the same variable. Such discrepancies occur whenever the “target” set of variables is inconsistent or when there are constraints in the subproblems that prevent matching the subproblem parameter to the target specified by the system-level optimizer. For instance, a very high L/D target may be incompatible with a very small aspect ratio target; or stress constraints in the subproblem may limit the acceptable span.

The system level optimizer provides a vector of target variable values, z , to the subproblems which execute their own optimizations. The subproblems minimize the discrepancy between their own local variables, $\{x\}$, and computed state variables, $\{y\}$, and the system level targets, $\{z\}$. The system-level optimization problem may be expressed as follows:

$$\begin{aligned}
 &\text{minimize:} && z_n \\
 &\text{with respect to:} && \{z\} = \{z_1 \dots z_n\} \\
 &\text{subject to:} && J_i^* = 0 \quad \forall \text{subproblems}_i
 \end{aligned} \tag{Eqn. 2-1}$$

where the i^{th} subproblem optimization is given by:

$$\begin{aligned}
 \text{minimize: } J_i &= \sum_{j=1}^m (x_j - z_j)^2 + \sum_{j=m+1}^n (y_j - z_j)^2 \\
 \text{with respect to: } \{x\} &= \{x_1 \dots x_m\}, \{x_{m+1} \dots x_n\} \\
 \text{subject to: } c_i(x, y) &\geq 0 \quad i = 1, k
 \end{aligned} \tag{Eqn. 2-2}$$

A key feature in the development of collaborative optimization was the recognition that the system level targets could be treated as fixed parameters in the subproblem optimization. This observation allowed the development of analytic expressions for the variation in the subproblem optimal objective function, J^* , as a function of the system level target vector, $\{z\}$. That is, the variation in the optimal objective as a function of a fixed parameter, p_j , is given by^{[57][58][59][60]}:

$$\frac{dF^*}{dp_j} = \frac{\partial F^*}{\partial p_j} - \lambda \frac{\partial}{\partial p_j} c(\{x^*\}, \{p\}) \tag{Eqn. 2-3}$$

This result is useful in discerning the effect that fixed parameters in an optimization problem have on the optimal result; just as the Lagrange multiplier itself is a measure of the effect that variation in the constraint has on the objective value. However, the corresponding expression for a subproblem optimization at the optimum is given by:

$$\frac{dJ^*}{dz_j} = \frac{\partial J^*}{\partial z_j} - \lambda \frac{\partial}{\partial z_j} c(\{x^*\}, \{p\}) \tag{Eqn. 2-4}$$

where J is given by Eqn. 2-2, and since the local constraints $\{c\}$ are only a function of local parameters $\{p\}$ and variables $\{z\}$ (the “parameters” z never appear in the local constraints by definition), then:

$$\frac{dJ^*}{dz_j} = \frac{\partial J^*}{\partial z_j} = -2(x_j - z_j) \quad \text{or} \quad \frac{dJ^*}{dz_j} = \frac{\partial J^*}{\partial z_j} = -2(y_j - z_j) \tag{Eqn. 2-5}$$

The validity of this post-optimal variation has been established through multiple derivations and computationally, through finite differencing of local subproblem solutions^{[36][38][54]}.

This architecture is well suited for application in real world design problems because it preserves the disciplinary organization that currently exists, limits the amount of communication to only the interdisciplinary variable values, and allows local solution techniques. The fundamental idea behind the development of the collaborative optimization architecture is that disciplinary experts should participate in the design decision process while not having to fully address local changes imposed by the other groups within the system. Explicit knowledge of the other groups' constraints or design variables is not required. This decentralized decision strategy is not only a practical approach to design, but also allows for the use of existing disciplinary analyses without major modification. This is not a trivial advantage, as the practical acceptance of many MDO techniques is limited by their implementation overhead requirements^[54].

2.2.6 Summary

Distributed architectures, such as the optimizer-based decomposition and disciplinary feasible constraint methods detailed earlier have numerous organizational and computational advantages over the standard approach. Organizational advantages include: (1) a more natural fit to the current disciplinary expertise structure found in most design organizations, (2) empowerment of the disciplinary experts in the design decision process (through subspace optimization) and (3) the flexibility to efficiently alter a portion of the design analyses without having to repose the complete problem. Similarly, computational advantages of a distributed system include: (1) a reduction in the integration and communication requirements, (2) a parallel optimization architecture which is readily operable on a suite of heterogeneous platforms and (3) removal of iteration loops.

In aerospace design, most optimization studies performed today rely on some form of integrated analysis and optimization method. While this brute-force approach may be the simplest to comprehend, it also has several performance drawbacks. It requires analysis integration which is not trivial. In addition, for the many problems with feedback between analyses the repetitive function evaluations

required for convergence can become excessively expensive—assuming that fixed point iteration is even convergent. On the other hand, while an OBD approach may provide the most efficient means toward a solution, this solution strategy may be difficult to implement, has large communications requirements, and leaves the domain-specific analyses with the role of function evaluation only. By maintaining several of the advantages of the OBD approach, while incorporating subspace optimization to augment the role of the disciplinary expert, the disciplinary constraint feasible methods (like CO) have the most to offer the engineering community towards the solution of large-scale, practical design problems.

The rules that define collaborative optimization are well suited for implementation in many of the frameworks developed in industry for performing design optimization. Tools like Next-Cut^[24] and ICM^[25] (Interdisciplinary Communications Medium) are used in design projects to facilitate interdisciplinary communication and provide a rational framework for design discussion. A hallmark of these tools is the attempt to preserve disciplinary design autonomy while guiding changes that lead to better design. The local autonomy provided by collaborative optimization is a direct analogy to this mechanism.

2.3 Wing Design Example

In this section a simple design problem is solved using single-level, OBD, sequential, and collaborative optimization. The design generated using CO matches the benchmark design found via single-level optimization. Sequential optimization, however, failed to find this optimal wing design; erroneously reporting a less successful design as the optimum.

2.3.1 Design Problem Description

The wing design problem is to specify the span, thickness and twist distributions for a wing that will maximize a performance function. The computed parameters for a wing that will maximize a performance function. The computed parameters for a wing that will maximize a performance function.

ters required for the evaluation of this function are the lift-to-drag ratio and the wing weight.

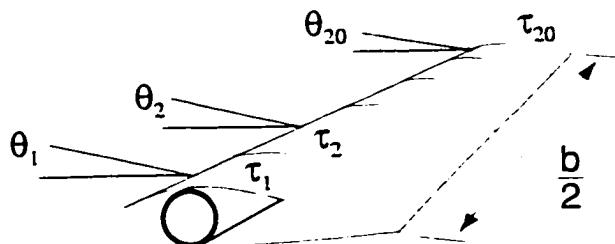


Figure 2.5 Wing design problem

Table 2.1 specifies the values of the fixed parameters used in this problem. The wing structure is represented as shown in Figure 2.5, with a tubular spar extending over the span of the wing. The tube diameter matches the wing quarter chord thickness at each of twenty spanwise stations. The wall thickness of the tube is specified for each of these twenty elements. Spar weight is computed by integrating the local spar weight over the span. The structural analysis computes the total wing weight and the normal stresses for each of the wing elements. The stresses at each section are calculated from the cross sectional moment of inertia and bending moment. The bending moment is computed by integrating the 1-g aerodynamic loading over the span. The objective function is a performance factor of the same form as the Breguet range equation: $R = k \frac{L}{D} \ln \left(\frac{W_{\text{total}}}{W_{\text{fixed}} + W_{\text{spar}}} \right)$

An incidence is specified at each of the spanwise stations. These incidences, along with the wing span and root angle of attack, are inputs to the aerodynamics analysis. A vortex lattice code is used to compute the inviscid drag of the wing, the spanwise lift distribution, and the total lift. Parasite drag is computed as the sum of a constant drag area, f_{other} , and a wing area dependent term, C_{D_p} . These two drag components and the total lift are used to compute the lift-to-drag ratio.

Parameter	Value	Parameter	Value
W_{total}	100,000 lbs.	σ_{Max}	60 ksi
W_{fixed}	70,000 lbs.	f_{other}	10 ft ²
AR	8.6	C_{D_p}	0.01
taper	0.32	n_g	3.0
t/c	0.12	ρ_{air}	0.101 lb/in ³
Sweep	30 deg.	Mach	0.82

TABLE 2.1 Wing design problem fixed parameters

2.3.2 Single Level Implementation

Figure 2.6 shows the use of single level optimization to solve this problem. The analyses are integrated and require the specification of the span, spanwise incidence, and thicknesses. The integrated analysis computes a range and stress vector and the optimizer adjusts the inputs to maximize the range.

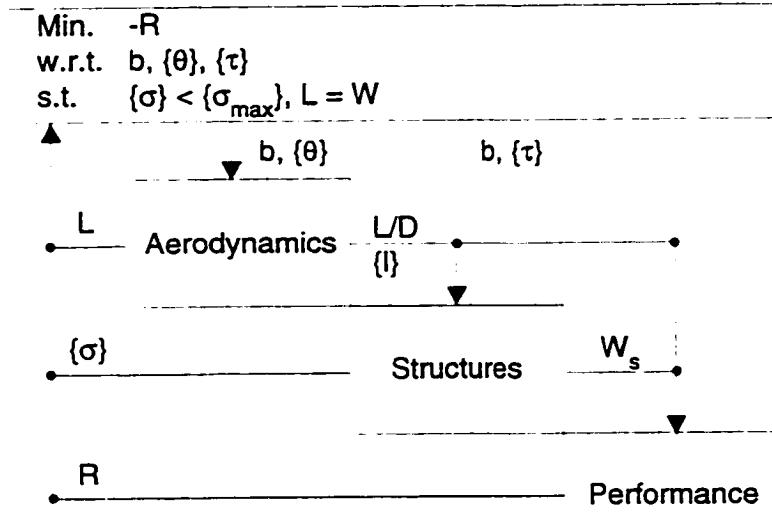


Figure 2.6 Single-level optimization

Note that the entire problem is expressed in forty-one design variables: twenty twists, twenty thicknesses, and the wing span. Intermediate values, such as the lift distribution, are computed and passed to subsequent analyses. The lift distribution

is computed for a single cruise loading case. There is no feedback in this simple problem as the wing is assumed to be built to a jig shape that will offset the effect of elastic twist and result in the prescribed incidence distribution. However, there are twenty constraints that limit the allowable stress and a constraint to ensure that the resultant lift equals the fixed weight.

2.3.3 Optimizer Based Decomposition (OBD) Implementation

The wing design problem implemented using the OBD approach is shown in Figure 2.7. Here, the explicit interdisciplinary coupling is removed through the addition of auxiliary variables for lift to drag ratio, lift distribution, and spar weight. The addition of compatibility constraints ensure that these design variables equal their corresponding computed quantities at the solution. The final values of design variables, computed parameters, and the performance function, R, for the OBD optimization is identical to the single-level solution and establishes the baseline result shown in Figure 2.10 and Figure 2.11.

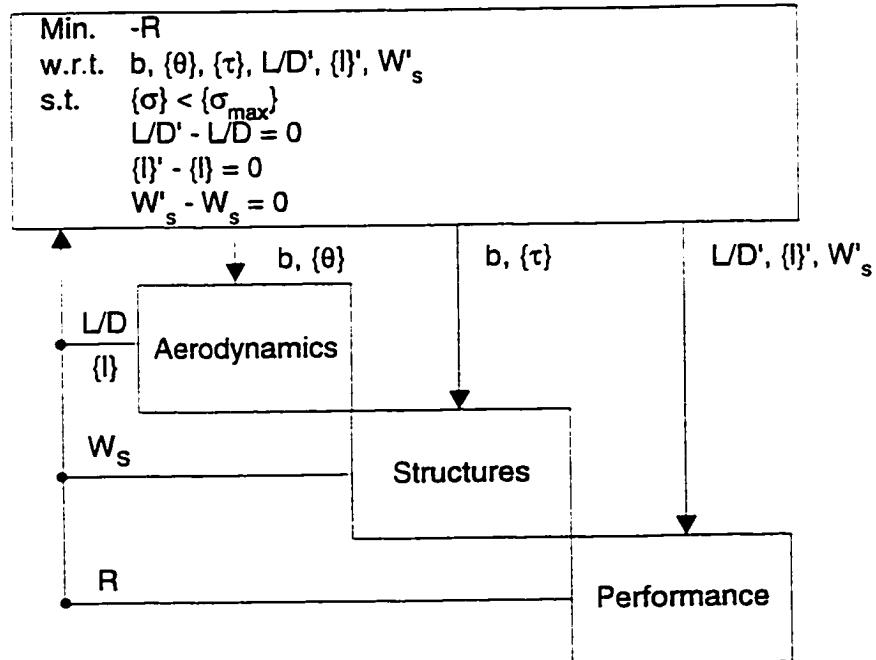


Figure 2.7 Wing design by optimization based decomposition

2.3.4 Design by Sequential Optimization

From the expression for performance given at the end of Section 2.3.1, it is clear that increasing L/D and decreasing spar weight benefit the performance factor R. It therefore would seem beneficial if the aerodynamics discipline finds a cruise twist distribution that maximizes L/D and the structures discipline manipulates the twenty thicknesses to minimize weight. This arrangement is attractive in industrial applications because many analysis tools are setup to perform discipline-specific optimization. For example, a structures FEM code may permit the user to minimize structural weight or an aerodynamics package may minimize drag. Sequential optimization is the serial solution of disciplinary optimization problems, perhaps in the context of a larger-scale iteration.

Figure 2.8 shows the wing design problem formulated as a sequential optimization. Because many of the variables, such as twist and thickness, are only needed in the subproblem optimizations, this approach dramatically reduces the size of the system level problem. The system level problem becomes a one variable system-level optimization executing two twenty-variable sub-optimization problems. The goal of the system level optimizer is to find the value of span that maximizes the performance index R. Organizing the design problem sequentially also moves all the constraints to the subproblem level. The twenty stress constraints become part of the structures subproblem and the lift constraint becomes part of the aerodynamics subproblem.

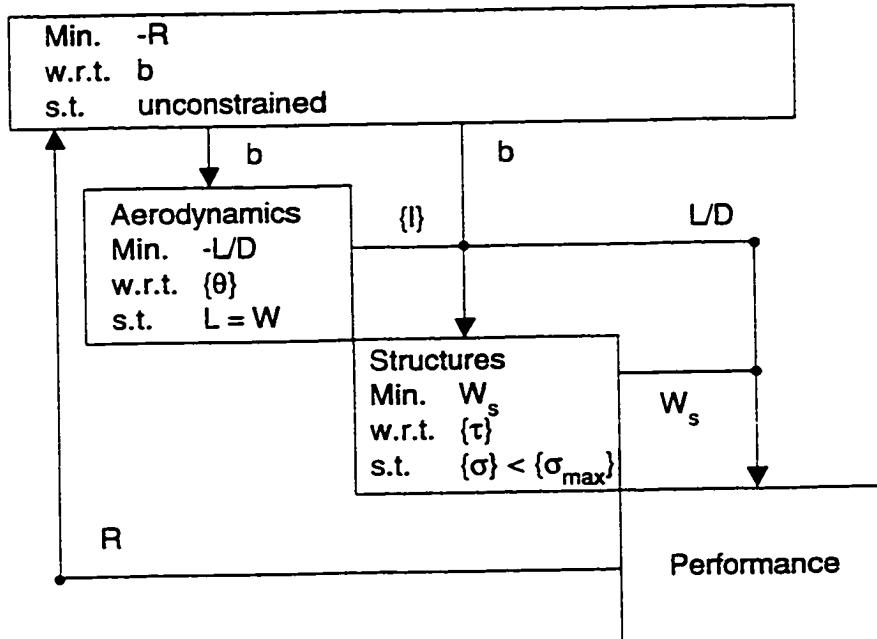


Figure 2.8 Wing design by sequential optimization

2.3.5 Collaborative Optimization

Figure 2.9 shows the wing design problem in a collaborative optimization format. A system level optimizer coordinates the decomposed design problem. This system level optimizer operates on four design variables including target span, target wing weight, and target L/D. The optimizer maximizes the performance factor while satisfying the compatibility constraints $J_1 = 0$ and $J_2 = 0$. Compatibility constraint gradient information is obtained analytically using post-optimal sensitivities^[38].

The analyses used in the subproblems remain the same as used in the sequential optimization. However, the form of the subproblem optimizations is changed. The details are shown in the boxes of Figure 2.9. The subspace optimization problems no longer minimize a single discipline-related quantity such as drag or weight. Rather, the subproblem optimizers adjust their design variables to minimize the difference

between local variables and corresponding system level targets (i.e. components of the discrepancy functions J_1 and J_2). The local objective functions are the sum of the squared differences between local design variables or computed results and corresponding system level targets:

$$J_1 = (\text{Span} - \text{Target Span})^2 + (\text{Lift Distribution} - \text{Target Lift Distribution})^2 + (\text{L/D} - \text{Target L/D})^2$$

Eqn. 2-1

$$J_2 = (\text{Span} - \text{Target Span})^2 + (\text{Lift Distribution} - \text{Target Lift Distribution})^2 + (\text{Spar Weight} - \text{Target Spar Weight})^2$$

As with sequential optimization, many variables are required only by a single discipline and so remain design variables in only one subproblem. Variables that are required by more than one discipline become design variables at the system and the subproblem level. Thus, the application of CO does increase the dimensionality of the subproblems. The aerodynamics subproblem now specifies not only the twenty values of wing incidence, but also its own value of span. The structures subproblem becomes dimensionally larger with the addition of local variables to control span and the lift distribution. As with local variables, local constraints remain strictly at the subproblem level (e.g. the twenty structures stress constraints).

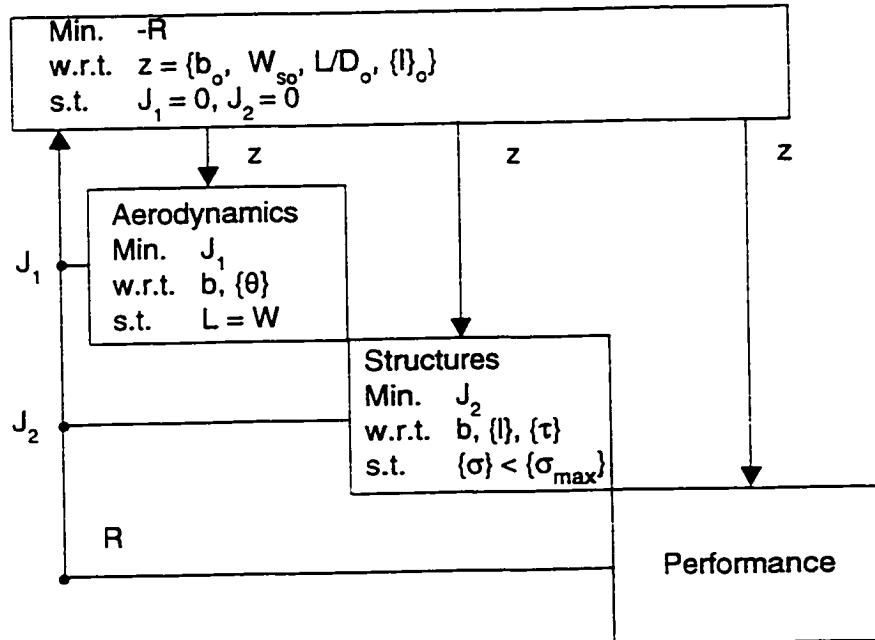


Figure 2.9 Wing design by collaborative optimization

2.3.6 Summary of Wing Design Results

The wing design problem described above was solved using single-level, sequential, and collaborative optimization. This section compares the solutions found by each of these methods. As seen in Figure 2.10, the value of the performance factor, R , obtained using sequential optimization is about two percent lower than that obtained from single-level optimization. Although the lift-to-drag ratio is higher in the sequential case the spar weight is much greater. Nearly the same values of computed results and performance factor are obtained using collaborative and single-level optimization. OBD obtains the same solution as single-level.

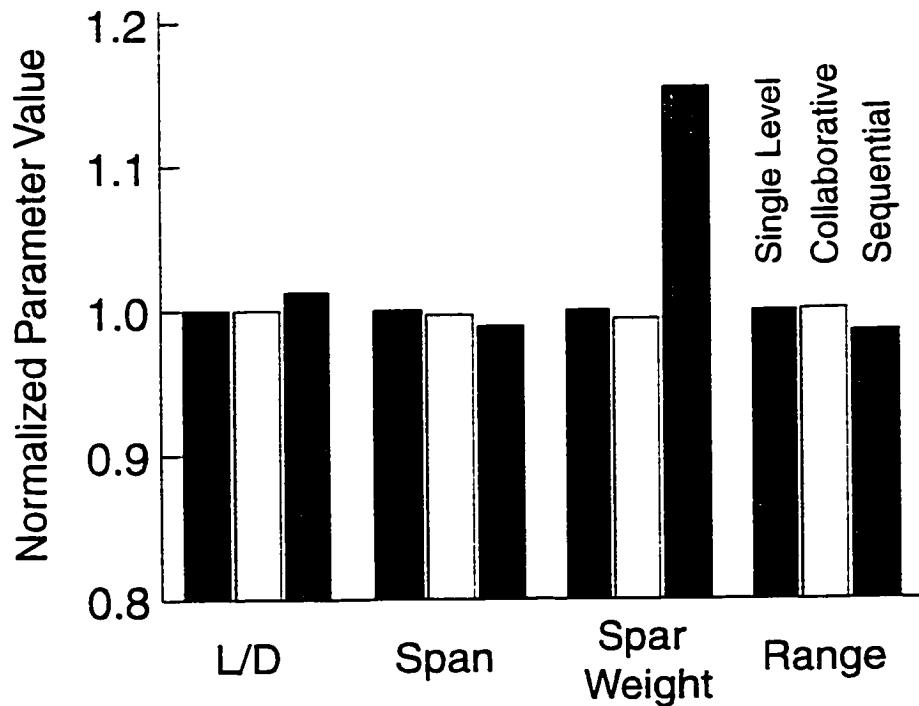


Figure 2.10 Optimal values of wing design parameters

The difference between the single-level and sequential optima occurs because the sequential architecture does not allow for trade-offs between disciplines except through the system level variable of the span. The aerodynamics optimization in the sequential problem maximizes L/D and provides the resultant lift distribution directly to the structures subproblem optimization, which then calculates stresses and thicknesses for minimum weight. But, because structures has no control over the lift distribution, it can not convey the advantage to overall performance that accrues when a slight drag penalty is accepted in exchange. Such a distribution decreases spanwise bending moments and total wing weight with very small drag penalties.

Each of the subproblems in the sequential optimization problem performed as expected. The aerodynamics subproblem designed a wing with a high L/D. By comparison, the optimum lift distribution found by the single-level optimization is far

from elliptic, with the centroid of lift much further inboard, as shown in Figure 2.11. Both designs are fully-stressed solutions, but since the single-level optimizer's lift distribution results in a smaller root bending moment, it is able to design a lighter wing with the same stress levels.

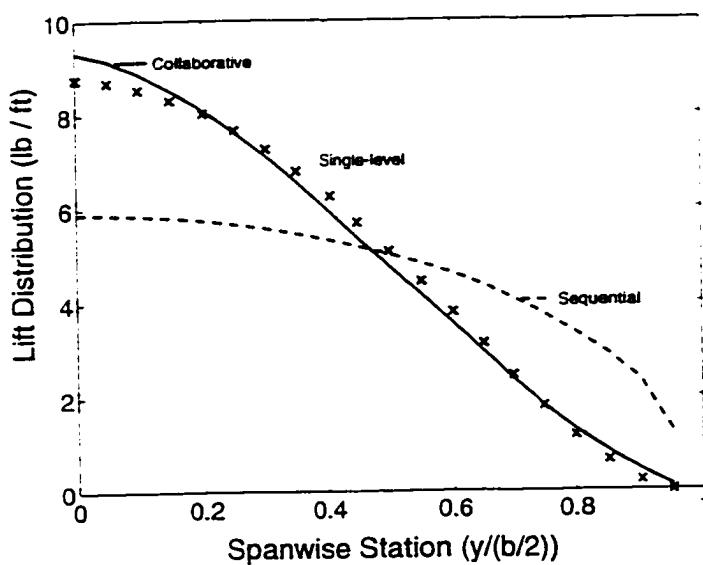


Figure 2.11 Lift distribution

The single-level optimization was able to trade drag for weight because it computed the sensitivity of the thickness and twist in terms of the performance function, R . In sequential optimization, these sensitivities were ignored. The correct solution, found by the single-level strategy, was obtained at the expense of a more complex analysis structure. The advantages of disciplinary autonomy have been lost and the entire problem must be solved as a single, large design task.

Collaborative optimization also finds the correct optimum. As seen in Figure 2.10, it successfully found the design trade-off between increased aerodynamic drag and structural weight, evidenced by its lower L/D and lower wing weight as compared with the sequential results. Figure 2.10 shows that the lift distribution also closely matches the results obtained from single-level optimization.

These results show that, despite its initial appeal, sequential optimization is an unacceptable approach to many design problems. To obtain correct solutions an optimization architecture must properly account for these cross-disciplinary effects.

A key advantage of CO over sequential or single level optimization is the ability to execute the disciplinary design problems in parallel. Figure 2.12 illustrates graphically how the wing design problem was solved using several processors in parallel. The system level optimization and range analysis was performed on an IBM RS/6000 computer, while the aerodynamics and structures designs were performed on two different SUN SparcStations.

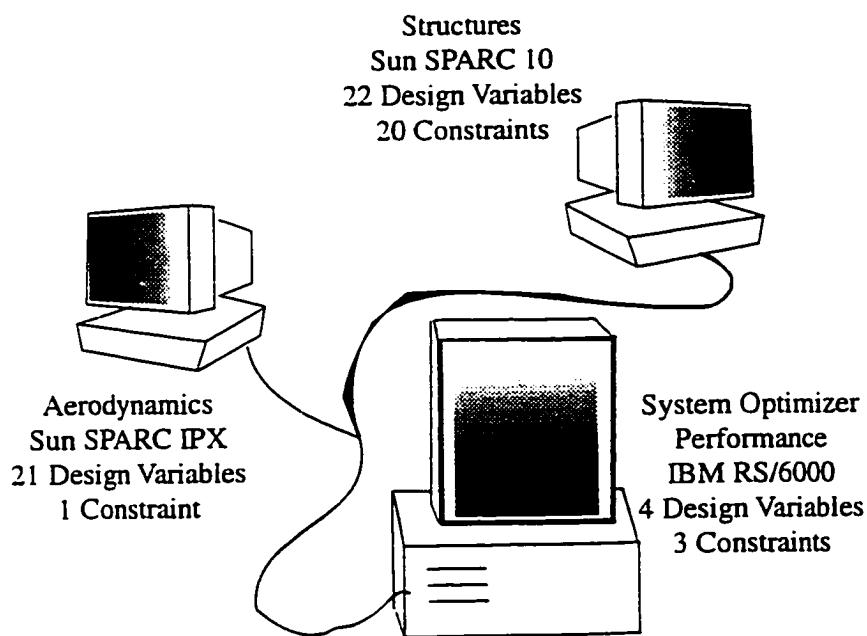


Figure 2.12 Wing design was executed collaboratively on three parallel processors

2.4 Summary

This chapter has examined some of the methods for applying formalized optimization to a multidisciplinary problem. Collaborative optimization is an attractive method that preserves local design decomposition and autonomy, keeps local design variables and constraints local, minimizes the amount of required interdisciplinary communication, and facilitates parallel analysis, all while providing a mechanism that correctly accounts for interdisciplinary coupling. These characteristics are highlighted in the solution of the wing design problem of Section 2.3. In the following chapter the characteristic performance of collaborative optimization is examined, as well as the theoretical performance based on the number of required function evaluations. However, the primary motivation for using collaborative optimization does not lie in the performance of the method on simple problems but in the advantages of the method when applied in a real world environment.

Chapter 3

Performance Properties of Collaborative Optimization

This chapter discusses the applicability, performance, and convergence properties of the conventional implementation of collaborative optimization. The applicability of collaborative optimization is evaluated in terms of the strength and breadth of the interdisciplinary coupling and the ratio of variables that are local to a single discipline to those that are shared between disciplines. The collaborative decomposition of design problem into subproblem optimizations is shown to have benefits or liabilities that are problem dependent. Performance and convergence behavior of CO is examined on a sample aircraft design problem which also highlights the method's sensitivity of performance to implementation details. The results from these sections motivate modifications to the basic method, such as the bandwidth reduction techniques of Chapter 4 and the introduction of response surfaces in Chapters 5, 6, and 7.

3.1 Theoretical Estimate of Performance

In the following sections estimates are made of the number of function evaluations required to find an optimal solution for problems with different interdisciplinary coupling strengths and bandwidths.

3.1.1 Comparison Assumptions

The required number of function evaluations will vary depending on the optimization algorithm used and the nonlinearity of the problem. For purposes of comparison it is assumed that all gradient information is accurately computed using a first order forward difference method, the functional dependencies $y_i = f(x)$ are quadratic and that the optimizer is a sequential quasi-newton solver.

These assumptions allow the approximation of the maximum number of function evaluations required to find a stationary point because, when they hold, one expects that:

1. The number of analysis evaluations required to obtain a gradient, and thus a line search direction, is one plus the number of inputs to the analysis.
2. A minimum can be found along this line by evaluating a single additional design point.
3. The maximum number of line searches required to find an optimum is equal to the total number of design variables.

3.1.2 Nomenclature

Figure 3.1 illustrates the relationship between an optimizer and a group of independent analyses in a multidisciplinary optimization problem. Each discipline requires a vector of input values $\{x\}_i$ and computes an output vector $\{y\}_i$. The disciplinary input vector is partitioned into two classes of variables: local input variables, $\{x_l\}_i$, that are used only by the i^{th} discipline, and global variables, $\{x_g\}_i$, that are used by at least one other discipline. The vector of design variables adjusted by the optimizer is made up of the union of these global and local input variables; $\{x\} = \{x_g\} \cup \{x_l\}$ where $\{x_g\} = \{x_g\}_1 \cup \{x_g\}_2 \cup \dots \cup \{x_g\}_N$ and $\{x_l\} = \{x_l\}_1 \cup \{x_l\}_2 \cup \dots \cup \{x_l\}_N$ where $\{x_l\}_i \cap \{x_l\}_j = \emptyset$ for all $i, j = 1, N$ but $\{x_g\}_i \cap \{x_g\}_j \neq \emptyset$ for some (i, j) pairs of analyses.

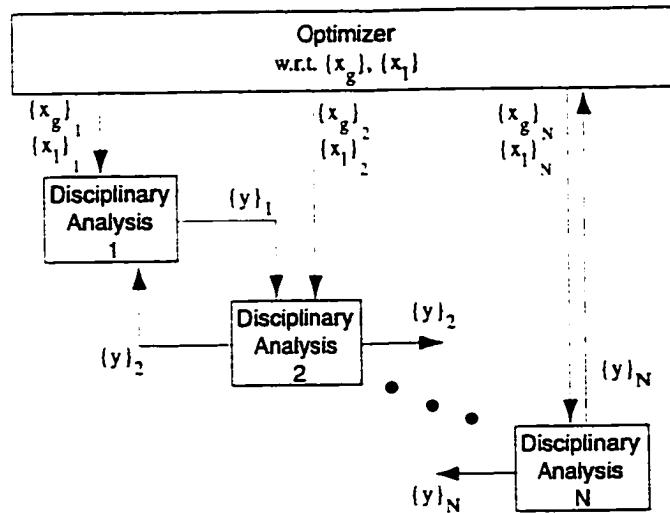


Figure 3.1 Single level optimization.

- $\{x_g\}$ = Vector of global variables
- $\{x_g\}_i$ = Portion of global variable vector used by i^{th} analysis
- $\{x_l\}$ = Vector of all strictly local variables
- $\{x_l\}_i$ = Variables used only by the i^{th} analysis
- $\{y\}_i$ = Parameters computed by the i^{th} analysis
- $\{x_{\text{aux}}\}$ = Vector of auxiliary variables (see Section 3.1.4)
- $\{x_{\text{aux}}\}_i$ = Portion of auxiliary variable vector used by i^{th} analysis
- z = System level target variable (see Section 3.1.5)

The quantity of each type of variable is given by:

- n_l_i = Number of strictly local input variables used by the i^{th} analysis
- n_l = Total number of local variables: used only by a single analysis = $\sum_1^N n_{l_i}$
- n_{g_i} = Size of subset of $\{x_g\}$ used by the i^{th} analysis
- n_g = Total number of global variables
- n_{aux_i} = Size of subset of $\{x_{\text{aux}}\}$ that corresponds to y_i from the i^{th} analysis
- n_{aux} = Total number of auxiliary variables = $\sum_1^N n_{\text{aux}_i}$
- N = Total number of disciplines

$p =$ Number of iterations required to converge feedback loops

3.1.3 Single Level Optimization

Figure 3.1 illustrates the standard multidisciplinary approach described in Section 2.2.1. The values of the global and local design variables are manipulated by the system level optimizer to improve the objective function. The total size of the system level design variable vector is, $n_g + n_l$. Feedback loops may require p executions to obtain a converged value of $\{y\}_i$ through iteration. Thus the maximum number of times an analysis must be executed to obtain one computed parameter vector $\{y\}_i$ and gradient $\nabla\{y\}$ is:

$$p((n_g + n_l) + 1) \quad \text{Eqn. 3-1}$$

The maximum number of line searches is equal to the number of system level design variables, $n_l + n_g$, so the maximum number of analysis executions required to obtain an optimal solution is:

$$p[(n_g + n_l)^2 + n_g + n_l] \quad \text{Eqn. 3-2}$$

3.1.4 Optimizer Based Decomposition (OBD)

OBD, introduced in Section 2.2.3, partitions analyses to permit concurrent execution. Intermediate variables are no longer passed between analyses, but rather between analyses and an optimizer. Auxiliary variables are introduced and constraints are added requiring that the computed variable values $\{y\}_i$ equal the auxiliary variable values $\{x_{aux}\}_i$. The size of the system level optimization problem increases by the number of these shared variables n_{aux} .

The system level design variable vector is composed of three kinds of subvectors, $x_g, x_{aux}, x_l \in x_g$. The vector $\{x_{aux}\}$ is also partitioned into subvectors, $\{x_{aux}\}_i$, corresponding to the variables used as inputs to the i^{th} analysis. The quantity of each type of variable (global, local, and auxiliary), for the i^{th} subproblem, is n_{g_i} , n_{l_i} , and n_{aux_i} .

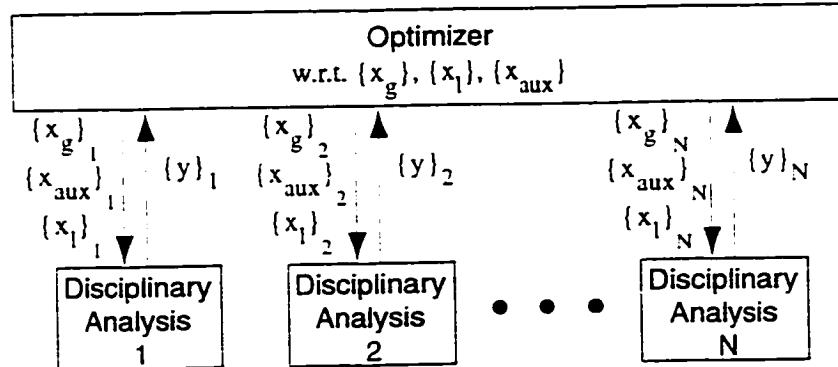


Figure 3.2 Optimizer based decomposition

For the i^{th} analysis, computation of an initial point and gradient requires $(n_{g_i} + n_{aux_i} + n_{l_i} + 1)$ function evaluations. If the optimization requires the maximum number of line searches then the total number of function evaluations to determine a stationary point is given by:

$$(n_{g_i} + n_{aux_i} + n_{l_i})(n_{g_i} + n_{aux_i} + n_{l_i} + 1) \quad \text{Eqn. 3-3}$$

3.1.5 Collaborative Optimization

The collaborative formulation poses each sub-analysis as an optimization whose goal is to match a set of system level targets. Computed quantities are not passed directly between disciplines; instead auxiliary variables, $\{z_{aux}\}$, are introduced at the system level that are used as target values for the computed quantities. The number of these auxiliary variables is equal to the number of computed parameters that are used as inputs to other disciplines.

The size of this target vector is equal to the number of global and auxiliary design variables. Each CO disciplinary analysis is an optimization in $n_{g_i} + n_{aux_i} + n_{l_i}$ variables.

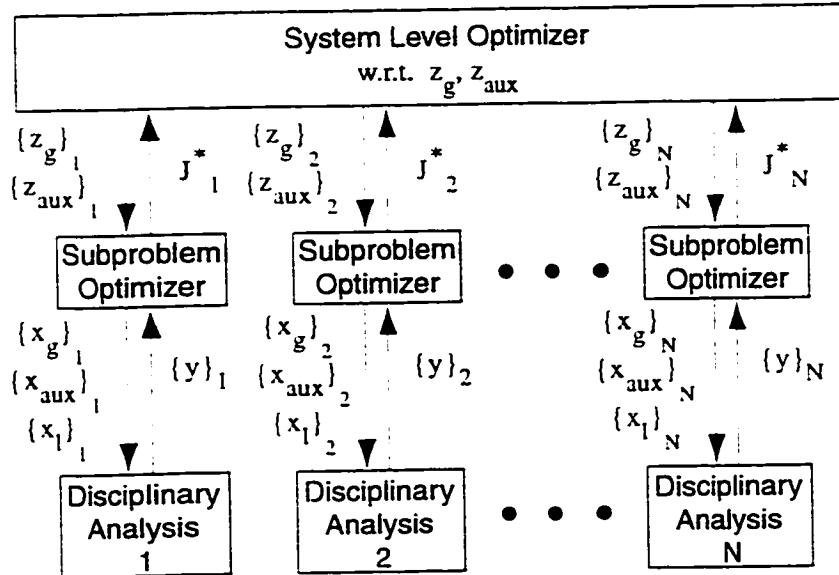


Figure 3.3 Collaborative implementation

Thus, the maximum number of function evaluations of the i^{th} subproblem analysis required to compute a single design point in the system optimization is:

$$(n_{g_i} + n_{aux_i} + n_{l_i})(n_{g_i} + n_{aux_i} + n_{l_i} + 1) \quad \text{Eqn. 3-4}$$

If gradient information is obtained using post-optimality sensitivities^{[54][57][58]}, then no further subproblem optimization, and thus no more function evaluations, are required. The maximum number of line searches at the system level is equal to the number of system level design variables: $n_g + n_{aux}$. Thus the total number of required function evaluations of the i^{th} analysis to find a stationary point at the system level is given by:

$$(n_g + n_{aux})(n_{g_i} + n_{aux_i} + n_{l_i})^2 + (n_{g_i} + n_{aux_i} + n_{l_i}) \quad \text{Eqn. 3-5}$$

3.2 Effect of Coupling Bandwidth on CO Applicability

Coupling bandwidth is a measure of the number of parameters shared between disciplines. In Figure 3.1 it can be measured as the sum of n_{aux} and n_g . It is the measure not only of the number of computed parameters that are used as inputs in other disciplines but also the number of shared variables.

For each additional parameter added to this vector the number of required analysis evaluations increases by as much as $[(n_{g_i} + n_{aux_i} + n_{l_i})^2 + (n_{g_i} + n_{aux_i} + n_{l_i})]$. Each global or auxiliary variable must be included in the system level design variable vector and so problems of larger bandwidth become substantially more computationally expensive. In practical problems this means that CO is best suited for problems with few interdisciplinary coupling variables relative to strictly local subproblem variables.

Disciplinary boundaries in practical design problems tend to be drawn in such a way that this is often true. There are exceptional cases where very high bandwidth cross disciplinary interactions exists; such as in aeroelasticity. Methods for accommodating high bandwidth information are discussed in Chapter 4.

In [64], several simple analytic problems were solved using collaborative optimization. The relative coupling between the problems could be adjusted by varying a single scalar parameter. Results from that study, summarized in Section 3.2.2, demonstrated that the performance of CO is insensitive to the strength of the interdisciplinary coupling. However, if the coupling is very weak the optimal solution may be insensitive to whether the coupling variable is included as a system level target or not. Since CO performance is affected by each additional component of the system level target design vector, very weak interdisciplinary coupling variables might be eliminated from the system level optimization problem.

3.2.1 Comparison of OBD and CO

The required number of evaluations of the i^{th} disciplinary analysis to reach a solution is given by:

$$\begin{aligned} F_{OBD_i} &= (A + n_l)B_i \\ F_{CO_i} &= AB_i(B_i - 1) \end{aligned} \quad \text{Eqn. 3-1}$$

where.

$$\begin{aligned} A &= n_g + n_{aux} \\ B_i &= n_{g_i} + n_{aux_i} + n_{l_i} + 1 \end{aligned} \quad \text{Eqn. 3-2}$$

From these relations it is possible show the effect of the relative proportion of the three types of variables, described in Section 3.1.2, on the required number of analysis executions to reach a stationary point. Taking the ratio of Eqn. 3-1:

$$\frac{F_{OBD_i}}{F_{CO_i}} = \frac{(A + n_l)B_i}{AB_i(B_i - 1)} = \frac{(A + n_l)}{A(B_i - 1)} \quad \text{Eqn. 3-3}$$

and recognizing that OBD will be functionally more expensive when this ratio is greater than one we obtain:

$$\begin{aligned} \frac{F_{OBD_i}}{F_{CO_i}} &= \frac{(A + n_l)}{A(B_i - 1)} > 1 \\ (A + n_l) &> A(B_i - 1) \\ n_l &> (n_g + n_{aux})(n_{g_i} + n_{aux_i} + n_{l_i} - 1) \end{aligned} \quad \text{Eqn. 3-4}$$

If we assume equally sized subproblems, where each problem requires the same number of global, auxiliary, and strictly local inputs, then:

$$\begin{aligned} n_l &= Nn_{l_i} \\ n_g &= Nn_{g_i} \\ n_{aux} &= Nn_{aux_i} \end{aligned} \quad \text{Eqn. 3-5}$$

and, Eqn. 3-4 can be rewritten as:

$$Nn_{l_i} > (Nn_{g_i} + Nn_{aux_i})(n_{g_i} + n_{aux_i} + n_{l_i} - 1) \quad \text{Eqn. 3-6}$$

which simplifies to:

$$\begin{aligned} (n_{g_i} + n_{aux_i}) + n_{l_i} &> (n_{g_i} + n_{aux_i})(n_{g_i} + n_{aux_i} + n_{l_i}) \\ 1 &> (n_{g_i} + n_{aux_i}) \end{aligned} \quad \text{Eqn. 3-7}$$

Eqn. 3-7 states that, for a MDO problem made up of identically sized analyses, subject to the assumptions of Section 3.1.1, the OBD approach is more efficient than CO, regardless of the actual number of local variables. This is shown for an arbitrary three subproblem optimization problem in Table 3.1.

Variable Type	Subproblem #1	Subproblem #2	Subproblem #3	Total	Total
n_g	2	2	2	n_g	6
n_{aux}	2	2	2	n_{aux}	6
n_l	100	100	100	n_s	300
$\frac{F_{OBD}}{F_{CO}}$	0.25	0.25	0.25	$\frac{F_{OBD}}{F_{CO}}$	0.25

TABLE 3.1 Equally sized disciplinary design problem.

The question of what happens when the subproblems are not identically sized is more difficult to answer in general terms. Table 3.2 represents subproblems of minimum cross disciplinary communication (having n_g and n_{aux} values equal to one), but large numbers of local variables. In particular, one subproblem is much larger, having more local variables, than the other two. In this case, OBD is more efficient

than CO in solving this one subproblem, but less efficient in solving the other two. This is because the large number of local variables used by the third subproblem results in a relatively larger number of line searches to find a global optimum than in CO. As a result, the first two subproblems are executed more often. In terms of the real-time computation expense, if the first two subproblems required more time to solve than the third subproblem then the fact that CO requires fewer of these computations would result in real time computational savings. This is a key case in which collaborative optimization is computationally advantageous. That occurs in cases with loose coupling between disciplines and one or more relatively large subproblems that run faster than the other smaller subproblems. An example of such a problem is a design process that couples a Navier Stokes aerodynamic analysis defined by a few variables, that takes days to run, with a trajectory analysis solved using a collocation method. With a collocation solver the trajectory is split into many smaller pieces where the governing equations of motion are solved subject to compatibility constraints at the segments' endpoints—resulting in a design problem with thousands of variables and constraints but very fast running analytic analyses^[66].

Variable Type	Subproblem #1	Subproblem #2	Subproblem #3	Total	Total
n_g	1	1	1	n_g	3
n_{aux}	1	1	1	n_{aux}	3
n_l	100	100	10000	n_s	10200
$\frac{F_{OBD}}{F_{CO}}$	16.7	16.7	.17	$\frac{F_{OBD}}{F_{CO}}$	11.18

TABLE 3.2 One large disciplinary design problem

The third case, shown in Table 3.3, demonstrates the limitations of this case. When the cross disciplinary coupling becomes slightly broader, the relative benefit of using CO decreases. Here the number of auxiliary and local variables increases substantially. The result is that the first two subproblems take 3 times longer to solve using OBD than CO; but the third problems is solved 29 times faster. Thus,

the first two subproblems would have to execute ten times slower than the third to make CO more efficient than OBD in real-time.

Variable Type	Subproblem	Subproblem	Subproblem	Total	Total
	#1	#2	#3	n _g	n _s
n _g	2	3	1	n _g	6
n _{aux}	15	6	2	n _{aux}	23
n _s	100	100	10000	n _s	10200
$\frac{F_{OBD}}{F_{CO}}$	3.01	3.24	0.035	$\frac{F_{OBD}}{F_{CO}}$	2.10

TABLE 3.3 One large and two medium sized disciplines

3.2.2 Performance Insensitive to Coupling Strength

Simple analytic problems have been developed to challenge discipline feasible methods like CSSO and CO^[53]. In [64], these problems were implemented using CO with successful results. The first example problem has two design variables, and a quadratic objective. The two constraints are linear, and the parameter β is used to modify their slopes, thereby shifting the location of the constrained optimum. Five different starting points are used to ensure that convergence is consistent.

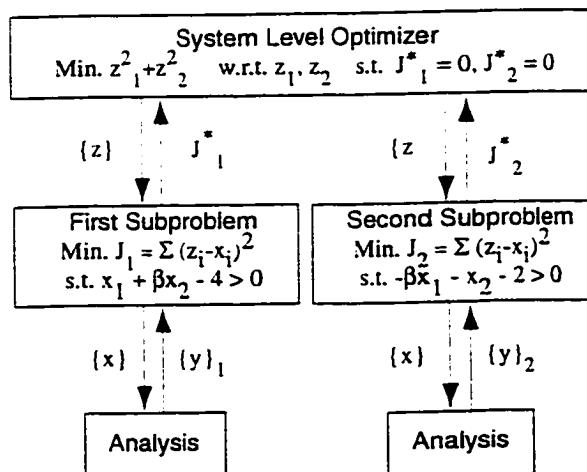


Figure 3.4 Collaborative formulation for sample problem from [53].

Successful convergence was achieved for all values of β , from five different starting points. The numbers of system-level iterations was less in all cases than the 71 required to solve the same problem using nonhierarchical decomposition. When there is no coupling between subproblems, the methods have similar performance. The coupling strength was increased from $\beta=0$ to $\beta=1$ with virtually no increase in required function evaluations.

3.2.3 Summary of Appropriate Problem Qualities

For very inexpensive problems with no feedbacks or with fast, accurately converging iterative loops, simple single level optimization may be most appropriate. Such situations are very common as academic exercises but very rare in real world design problems. For problems with many computationally intensive analyses the parallel execution of analysis tasks possible in methods like CO and OBD become attractive.

For a sample problem such as that described in Section 3.1.1 CO will be more efficient than OBD when relatively slow disciplinary problems in few variables are combined with fast running, large disciplines. OBD is more efficient on problems of equal size regardless of the actual number of local variables. However, design solution with inverse methods and problems of disparate size and solution speed, such as collocation solvers mixed with Navier Stokes codes, may actually be less expensive to solve collaboratively than with OBD.

Comparing Eqn. 3-3 and Eqn. 3-5 we see that the number of line searches required for a CO solution will be smaller than an OBD solution. Thus, if the local optimal solution could be obtained without a subproblem optimization then the CO implementation would be substantially less expensive. Inverse design techniques, such as full stressed structural design, are a class of subproblem design for which no gradient based optimization is required.

In many ways this comparison between OBD and CO is inappropriate. For OBD and CO should not be applied to the same types of problems. The results of this

section only confirm that if a design problem can be implemented in an OBD architecture, it should be. However, for large, detailed design problems, it is impractical to use the OBD architecture. There are too many local constraints understood locally, to combine them blindly with an optimizer. CO keeps the results closest to people with the expertise to judge the results.

The assessment of the last few sections is based on estimated number of disciplinary analyses. In real world design environments the functional efficiency of a method plays only a part in the assessment of its applicability. In practical design problems issues such as the cultural acceptance of the new design method, ease of use, and robustness, may be more important. CO provides an unprecedented degree of disciplinary autonomy: local design variables are defined locally and are not communicated to other disciplines or higher authority.

3.3 Artifacts of Decomposition

The decomposition of the design problem can lead the system level optimizer to select poor search directions. In other cases the decomposition can avoid local minima that a single level optimization will not. Consider the optimization problem defined by:

$$\begin{aligned}
 & \text{minimize:} && y \\
 & \text{with respect to:} && x, y \\
 & \text{subject to:} && g = y^4 - x
 \end{aligned} \tag{Eqn. 3-1}$$

The topology for this problem is shown in Figure 3.5. For the point shown ($x = 4, y = -0.5$) the gradient of the objective function f and constraint g are as shown. If this were posed as a subproblem however, tasked with matching the system level target shown by T , then the subproblem optimal solution might be as shown in the figure. In such a case, the gradient of the system level discrepancy constraint J would be as shown, leading to a resultant search direction at the system level very different, and not as good as, that suggested by a single level optimization. This is

because single level has a better model of the constraint and objective functions than the collaborative system level optimizer.

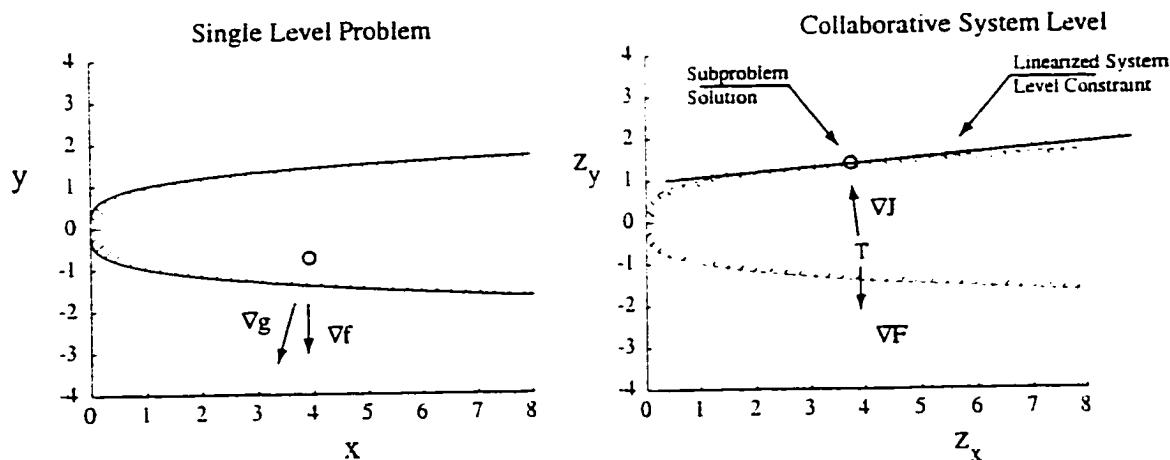


Figure 3.5 Decomposition can lead to poor search directions.

On the other hand, the collaborative decomposition of the design problem can have a beneficial effect on convergence to a global optimum. Consider the example defined by Eqn. 3-2 and illustrated by Figure 3.6.

$$\begin{aligned}
 & \text{minimize:} && x_2 \\
 & \text{with respect to:} && x_1, x_2 \\
 & \text{subject to:} && \\
 & g_1 = x_1 > -8 && \text{Eqn. 3-2} \\
 & g_2 = x_1^2 - x_2 - 10 > 0 \\
 & g_3 = x_1 - x_2 < 0
 \end{aligned}$$

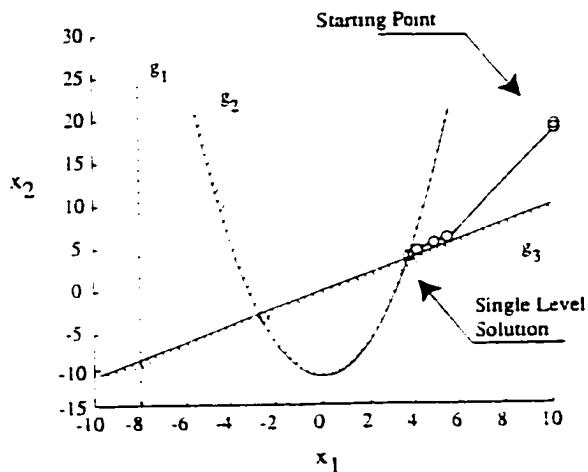


Figure 3.6 Solution to single level problem with two constraints is local optimum.

Here the interaction of a nonlinear and linear constraint creates a local minimum that may trap an optimizer. A decomposition that separates these two constraints results in two subproblems that, at this same point, has a system-level search direction for improvement.

$$\begin{aligned}
 & \text{minimize:} && z_{x_2} \\
 & \text{with respect to:} && z_{x_1}, z_{x_2} \\
 & \text{subject to:} && J_1 = 0 \\
 & && J_2 = 0
 \end{aligned} \tag{Eqn. 3-3}$$

where the subproblems are defined by:

$$\begin{aligned}
 & \text{minimize:} && J_1 = (x_1 - z_{x_1})^2 + (x_2 - z_{x_2})^2 \\
 & \text{with respect to:} && x_1, x_2 \\
 & \text{subject to:} && g_2 = x_1^2 - x_2 - 10 > 0
 \end{aligned} \tag{Eqn. 3-4}$$

and.

$$\begin{aligned}
 \text{minimize: } & J_2 = (x_1 - z_{x_1})^2 + (x_2 - z_{x_2})^2 \\
 \text{with respect to: } & x_1, x_2 \\
 \text{subject to: } & g_1 = x_1 > -8 \\
 & g_3 = x_1 - x_2 < 0
 \end{aligned} \tag{Eqn. 3-5}$$

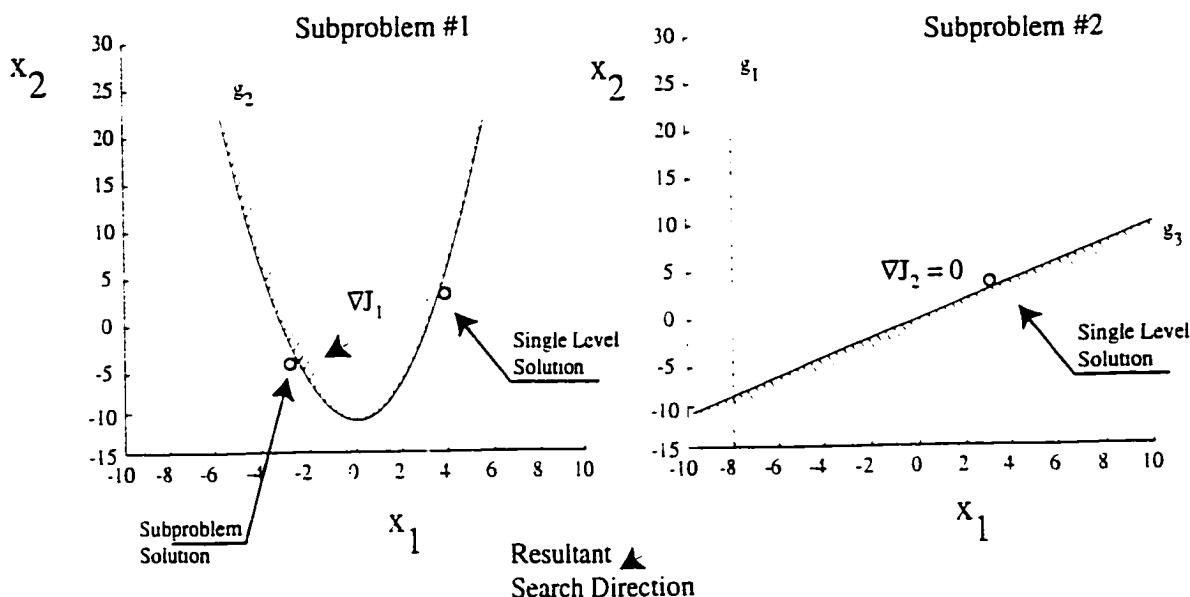


Figure 3.7 Decomposition of constraints allows movement toward global optimum

This problem was solved collaboratively and with a single level optimization. The solution paths for each approach are shown in Figure 3.8. Note that the single level solution becomes stuck at the non-global optimum solution—the intersection of the two constraints. It finds an objective function value of 3.701571 at $(x_1=3.71, x_2=3.71)$ in 9 major iterations. If the CO implementation groups the g_1 and g_3 constraints together into one subproblem and the g_2 constraint in the second subproblem than the solution path shown in Figure 3.8 is the collaborative result. Note that

CO converges to a better optimal solution ($x_1 = -8.0$, $x_2 = -8.0$) than single level optimization in 21 major iterations.

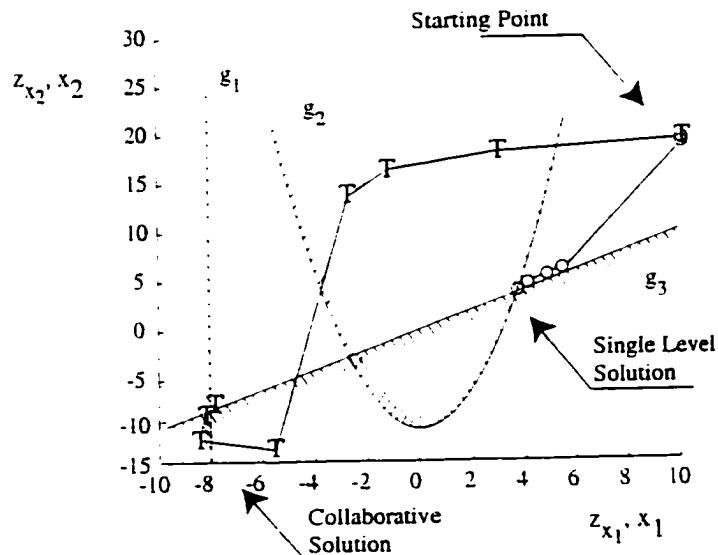


Figure 3.8 CO finds a better design point for Eqn. 3-2 than single level optimization

3.4 Performance on Sample Problem

Only a few of the multi-level architectures have been formally proven to be convergent^[61]. Lacking such proof, confidence in the robustness of the CO approach is developed by examining the performance of the method on problems. In this thesis there are several examples of successful application of CO to design problems. As mentioned in the Chapter 1, CO has been applied to a variety of other design problems^{[54][62][63]} and problems specifically designed to cause trouble for hierarchical methods like CO^[64]. In this section the results of a simple medium range transport sizing problem is reported. The problem is solved using the conventional implementation of CO and demonstrates the importance of implementation details in CO performance, a comparison between OBD and CO on a real design problem and an example of typical convergence behavior. The effect of changing subproblem dimensionality is also demonstrated.

3.4.1 Description of the Aircraft Sizing Problem

In this problem the wing planform is designed for a medium sized transport aircraft. Free design variables include the distributions of material thickness and twist along the wing, as well as its aspect ratio and area. The goal is to produce a design with maximum range evaluated at two flight conditions with static aeroelastic effects included. Figure 3.9 is a block diagram illustrating the analysis dependencies in the aircraft sizing problem. Lift distributions and inviscid drag are computed using a vortex lattice method for two flight conditions; level flight and gust. The total drag is represented as the sum of inviscid drag, compressibility drag, viscous lift-dependent drag, and parasite drag. The analysis takes planform area, aspect ratio, aeroelastic twist, and jig twist as inputs. The jig twist is specified independently at several spanwise stations. The incidence of a particular panel section is the sum of the jig twist at that panel and the aeroelastic twist computed by the structures analysis. The resulting lift distribution is represented by two Fourier coefficients as outlined in [65] and Section 4.2.4.

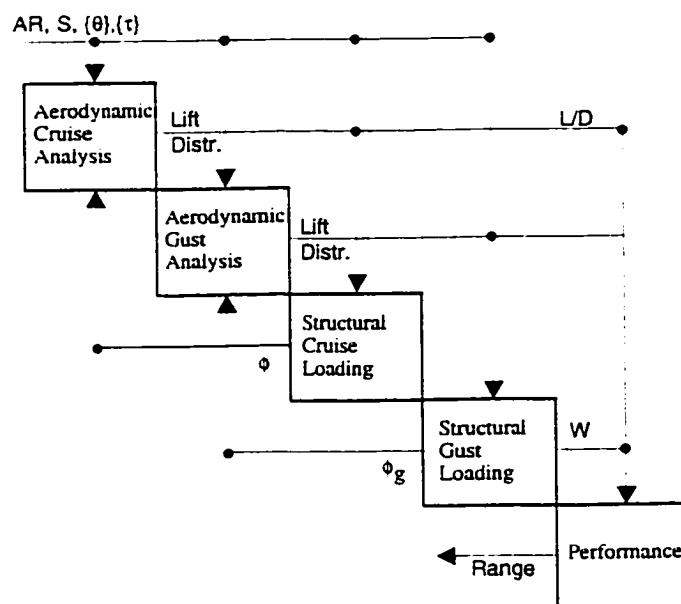


Figure 3.9 Aircraft sizing problem

The structural analysis is also performed for each loading case. The wing span and chord distribution are computed from the aspect ratio and planform area. The load carrying wing is modeled as a wing box stretching from 1/4 to 3/4 chord and equal in height to the specified wing thickness at that spanwise location. The wing is divided into a number of spanwise elements, and the skin thickness of the upper and lower wing box surfaces are specified for each element.

The twist due to bending of this swept wing is computed by passing the stiffness and loading distribution into a finite element code. The code represents the wing as a series of beam elements and computes the twist due to bending deflections in each of the loading cases. These twists are used by the aerodynamics analysis in the computation of the lift distribution.

3.4.2 Optimizer Based Decomposition Solution

Figure 3.10 shows the OBD implementation of this problem. There are a total of 45 design variables: twenty jig twists, twenty thicknesses, planform area, aspect ratio, wing weight, twist due to gust loading, and twist due to cruise loading. The results from this implementation were used to establish a baseline answer and for comparison with CO. In this case parallel processors were not used to solve the analysis. Thus, OBD was implemented to simply remove iterative feedback loops rather than to completely decoupled the analyses.

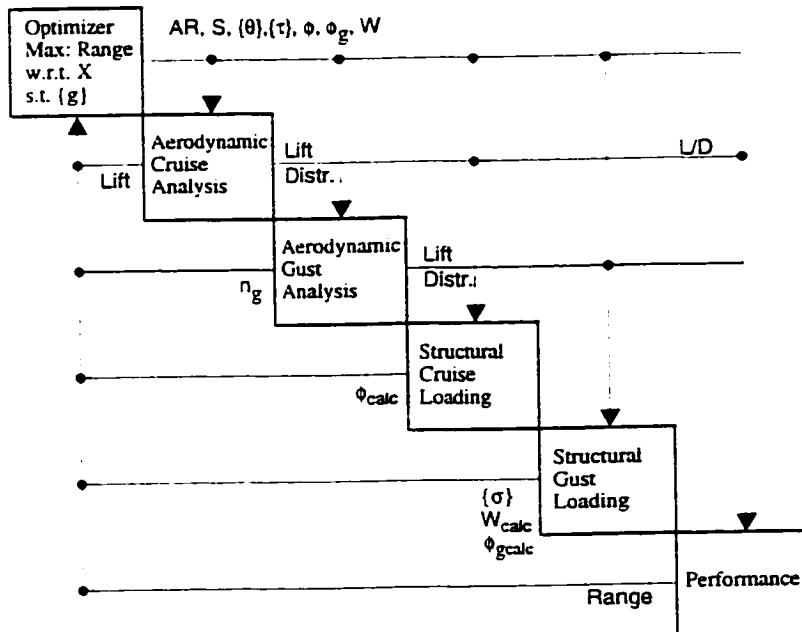


Figure 3.10 OBD implementation of aircraft sizing problem

The gust and cruise aeroelastic twists and wing weight are designated design variables to remove iteration from within the analysis. The computed values from the structural analysis are not used directly by the aerodynamic discipline. Rather, design variables representing these quantities are introduced and used for the aerodynamics calculations. A constraint is added to ensure that the design variable value used by aerodynamics equals the computed result from structures. The resultant analysis structure of Figure 3.10 has no iterative loops.

3.4.3 Collaborative Optimization

The architecture of the aircraft sizing problem solved using collaborative optimization is shown in Figure 3.11. All shared inputs and outputs between disciplines in Figure 3.9 are replaced by targets specified by the system level optimizer. There are

nine of these variables. Neither subproblem requires all nine elements of the design vector.

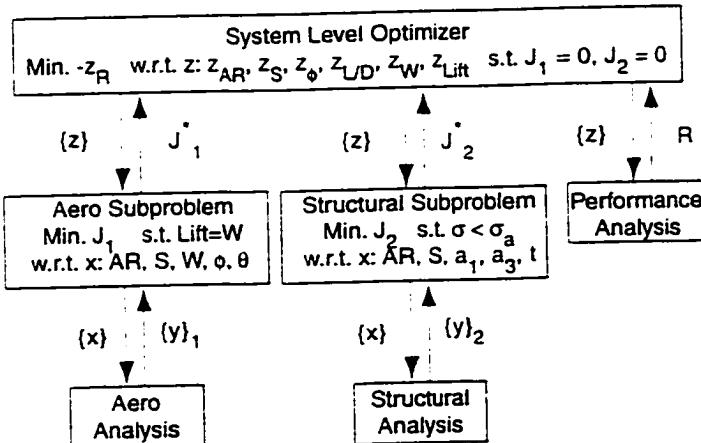


Figure 3.11 Collaborative solution of aircraft sizing problem

In addition there are two system level compatibility constraints. The objective is the range computed as a simple analytic function of the target lift to drag ratio and target weight fraction.

3.4.4 Aircraft Design Results

Figure 3.12 and Figure 3.13 illustrates the iteration history of two parameters for OBD and collaborative optimization. A comparison of the design variable values and computed results at the optimum is shown in Figure 3.14. The objective function improves threefold from an initial value of 860 miles to a final value of 3020 miles. Aspect ratio is reduced by twenty-eight percent and wing area is decreased by forty percent. The results shown are for converged solutions. The small discrepancy between the final variable and computed values results from two factors. First, to reduce the dimensionality of the system level target design variable vector, a reduced basis approach was used to represent the lift distribution. This issue will be discussed more in Chapter 4. Secondly, the subproblem optimizations are executed to a specified tolerance and thus allow for slight interdisciplinary discrepancies. These discrepancies may be removed by tightening the tolerances, at the expense of increasing the required number of function evaluations.

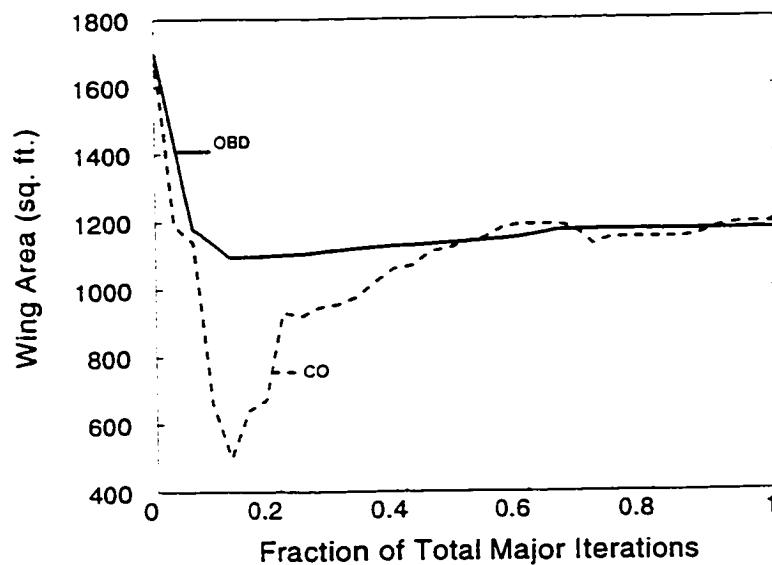


Figure 3.12 Variation of wing area in CO and OBD design cycles

Both methods find virtually the same optimal value of the objective function. CO determined a range of 3026 miles, eight tenths of one percent from the OBD result of 3003 miles. The other values of design parameters are all within one percent of each other at the optimum.

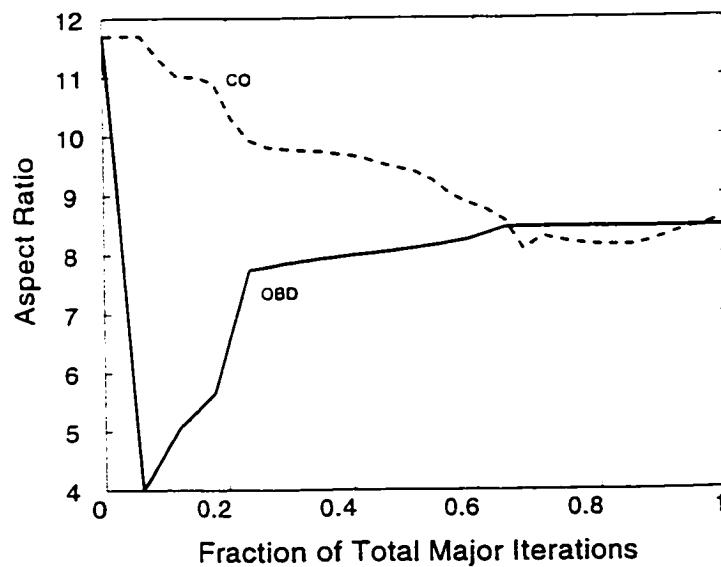


Figure 3.13 Variation of aspect ratio in CO and OBD design cycles

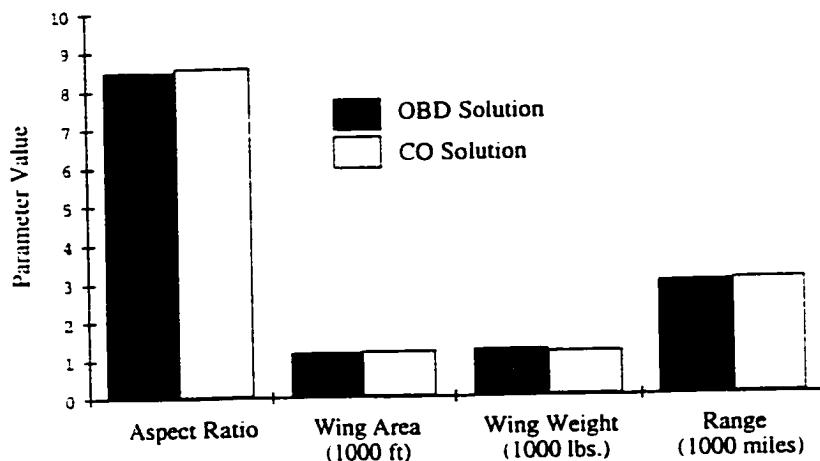


Figure 3.14 Design Variables and Computed Results

Figure 3.15 shows the spanwise distribution of skin thickness for the OBD and collaborative optimal solutions. The constant thickness sections outboard are due to minimum gauge constraints. These results are important because they show that the independent structural design discipline found virtually the same values of local thickness variables as the OBD implementation.

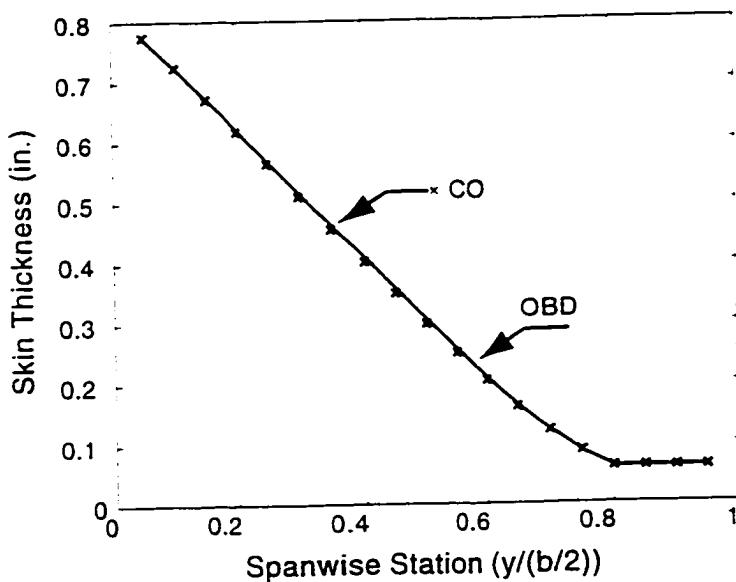


Figure 3.15 Spanwise thickness distribution for OBD and CO solutions

3.4.5 Implementation Details Critical to Performance

The performance of CO on problems can be substantially affected by details of its numerical implementation. Thus far, all the problems solved using CO have used the SQP algorithm NPSOL for both the system and subproblem level optimizations. During each subproblem optimization approximate information is collected to generate the Hessian and Lagrange multipliers. This information is lost if the subproblem optimization is simply restarted with the new target point and an identity matrix for the Hessian. NPSOL provides the option of performing a warm start where this information is stored and used during the next major iteration. Use of information from previous subproblem optimizations should improve overall efficiency^[54]. In Table 3.4 we see the number of subproblem analysis evaluations used to find a global optimum for the airplane design problem of the previous section. Shown is the effect of simply starting the subproblem from the previous solution; reducing the number of analyses by 50%. Full warm starting reduces the number of function evaluations by 72%.

Implementation	Number of Major Iterations	Number of Analysis Calls
Baseline	36	154152
Previous Point	27	75928
Full Warm Start	21	43312

TABLE 3.4 Effect of SQP implementation on CO efficiency

The accuracy of the gradients obtained using post-optimality information is also critical to the successful and rapid convergence of CO. The accuracy of the derivative require that the subproblem design be a local optimum that satisfies the KKT conditions. The degree of accuracy in the design point must be high. Table 3.5 shows the effect in the accuracy of the post-optimal gradients as a function of the convergence tolerance of the NPSOL. The value shown is for $\frac{\partial J_1}{\partial a_{1,g}}$ from the aerodynamics subproblem of Section 3.4.3.

Convergence Tolerance	Feasibility Tolerance	Post-Optimal Sensitivity	Error (%)
10^{-1}	10^{-1}	0.704427908	67.556
10^{-2}	10^{-1}	0.956798226	55.934
10^{-2}	10^{-2}	2.111217845	2.765
10^{-3}	10^{-2}	2.17004927	0.056
10^{-5}	10^{-4}	2.170566821	0.032
10^{-7}	10^{-6}	2.171320201	.002
10^{-9}	10^{-7}	2.171266448	0.0

TABLE 3.5 Effect of optimality tolerances on derivative accuracy

The convergence tolerance roughly corresponds to the number of accurate significant digits in the solution vector and objective function value^[97]. Thus, the convergence tolerance on the subproblems must always be tighter than the feasibility tolerance for the system level optimizer.

3.4.6 Performance of Example Airplane Problem

The table below compares the number of function evaluations for the aerodynamics analysis solved with each of the three methods discussed in this chapter. The single-level optimization required the greatest number of analysis evaluations to find a solution, in large part because of the fixed iteration between aerodynamics and structures that took many cycles to converge. Though the solution was found in 25 major iterations, the analysis was executed 85,100 times. CO converged in substantially fewer major iterations than OBD but, since each CO major iteration includes at least one complete subproblem optimization, the number of analysis calls was five times greater. This is on the same order of functional difference predicted in Section 3.2.1.

Method	Number of Major Iterations	Number of Analysis Calls
Single Level	25	85100
CO	21	43312
OBD	36	8510

TABLE 3.6 Computational expense of airplane design problem

3.4.7 Convergence Behavior

The conventional implementation of collaborative optimization exhibits characteristic convergence behavior shown in Figure 3.16. The objective function overshoots the known optimal solution and is drawn back toward the correct answer as the compatibility between the subproblems is obtained. This behavior is evident also in Figure 3.8, in all the conventional implementation problems of this thesis, and was observed in [54]. The reason for the overshoot is clear by examining the system level optimization formulation. The gradient of the objective is always equal to one while the constraint values vary between zero at compatible design points and non-zero at other points. Thus, even if the problem were started at the optimal solution the system level optimizer would propose a new point with a much better

objective function value and, upon finding the compatibility constraints violated, would reduce the objective function target until it returned to the optimal point.

The compatibility constraints take on a zero value and zero gradient at all compatible points. This is, of course, the case at the problem optima. This can cause trouble for SQP methods like NPSOL because they rely on first order information about the constraints to determine search directions, step sizes, and to generate the approximate Hessian. The singularity at the solution is overcome by NPSOL through adjustment of internal penalty parameters that decrease the step size. However, this also results in very small step sizes and relatively slow convergence in the region of the optimal solution. This is also shown in Figure 3.16, where the optimal solution is found in the first 50% of the major iterations but the optimization continues to struggle to satisfy the compatibility tolerances.

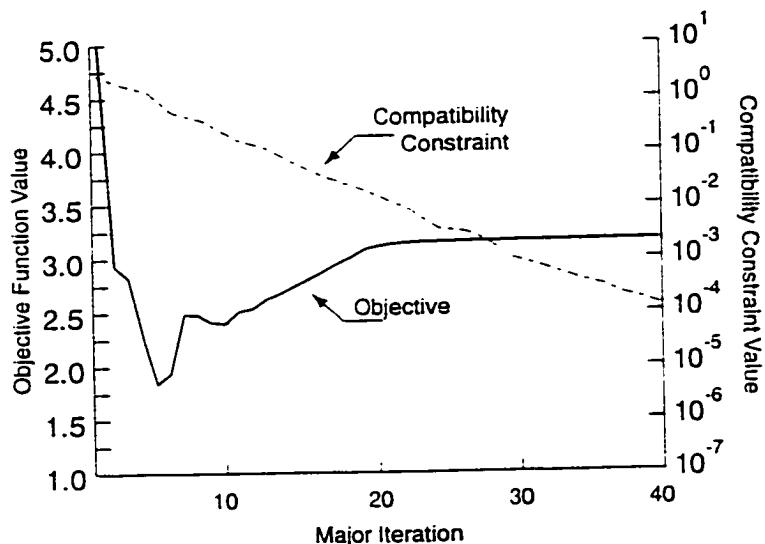


Figure 3.16 Variation in objective function and discrepancy constraints with major iteration.

3.4.8 Effect of Increasing Subspace Dimensionality

As discussed theoretically in Section 3.2.3, the number of collaborative system level iterations should be insensitive to the number of strictly local design variables. By varying the number of panels used to model the structures of the wing it was possible to observe the effects of increasing subspace dimensionality on the system level performance (Figure 3.17). As the number of panels is increased from two to seventy the number of collaborative system level iterations remains relatively constant. This contrasts with the steadily increasing number of OBD optimizer iterations. This does not imply that the collaborative solution is computationally cheaper as the number of variables is increased: each system iteration still requires the sub-spaces to do a complete optimization. However, it does show that the convergence rate of the system level optimization is relatively insensitive to changes in the number of design variables at the subproblem level.

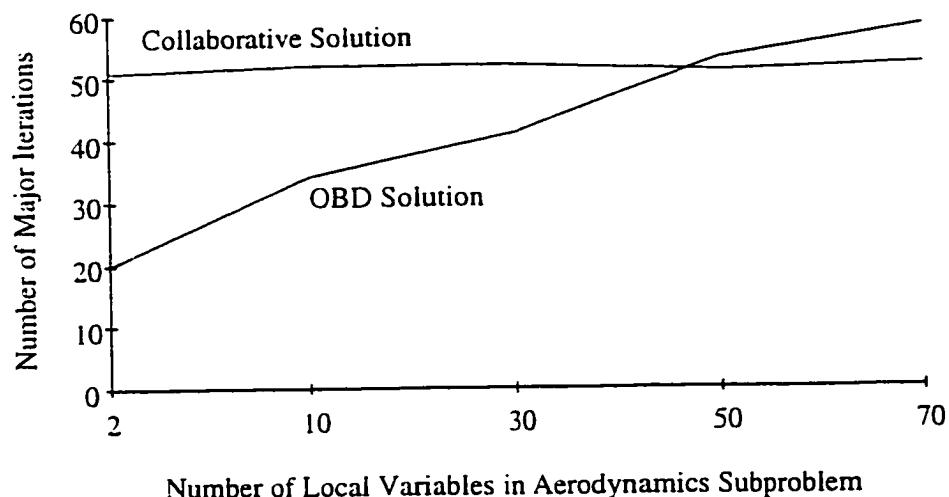


Figure 3.17 Variation in number of major iterations with local variables.

3.5 Summary

In an comparison based strictly on the number of function evaluations there is only a narrow class of problems in which collaborative optimization is superior to optimizer based decomposition. Problems where a slow running analysis described in a few variables is combined with a fast running analysis in many variables, and problems amenable to inverse design, are two classes of design problems where CO may be superior in terms of required function evaluations to find an optimum. However, the primary motivation for using CO is that real world design problems are too large and the expertise too diffuse to combine in an OBD architecture. Collaborative optimization provides an unprecedeted level of disciplinary autonomy and keeps design choices close to the disciplinary level.

The conventional implementation of collaborative optimization is effective in finding the solution to MDO problems. As evidenced in this chapter the hierarchical decomposition can sometimes be harmful but also helpful. But the conventional method has been used successfully to solve many problems, including the wing design problem of Chapter 2 and the aircraft design problem shown in this chapter. Emphasized here is the need to implement CO correctly, ensuring the proper relative values of optimizer convergence tolerances and the use of warm starting techniques to avoid unnecessary function evaluations.

The challenge for further development of collaborative optimization is to address some of the characteristics of the method. First, as described in Section 3.1.5 the number of function evaluations varies with the cube of the number of system level design variables. Though CO will always be most applicable to narrow bandwidth problems there are simple techniques, discussed in Chapter 4, for reducing bandwidth of interdisciplinary coupling. One promising response to the computational expensive subproblem optimization is the use of approximate methods. The use of response surfaces to represent very expensive analyses has been gaining attention in the last few years. The idea presented in the following chapters is to generate low order models of the subproblem optimal designs and use these models in lieu of the expensive full analysis to suggest new design points.

Chapter 4

Reducing High Bandwidth Coupling

The previous chapter showed that collaborative optimization is best suited for problems with few interdisciplinary variables relative to the total number of design variables. This tends to be the way large real world multidisciplinary problems are decomposed. However, special cases of high bandwidth coupling are able to be successfully reformulated as low bandwidth problems without changing the overall decomposition of the problem.

4.1 Special Case of High Bandwidth Coupling

Consider the wing design problem of Figure 3.9. The lift distribution was defined at twenty spanwise stations. The resulting lift distribution was then used by the structures discipline to determine a structural design that satisfied direct stress constraints due to bending. In the single level optimization method of Section 3.4.1 the lift distribution vector was passed directly from the aerodynamics to the structures discipline. In a collaborative implementation, a system level target design variable must be introduced for each element of this vector. Doing so substantially increases the size of the system level optimization problem. This is an example of a problem with relatively low interdisciplinary coupling except for the single high bandwidth coupling between aerodynamics and structures. As described in Section 3.1.5, the number of system level major iterations varies in proportion to the number of system level design variables. Because each system level iteration requires on the order

of n^2 function evaluations of the contributing analysis, the computational expense of the CO method can be reduced by as much as $3n^2 - 3n + 1$ for each variable eliminated from the system level problem. This provides clear motivation for minimizing the size of the system level variable set.

The aeroelastic coupling described here represents a special case of high bandwidth coupling. First, the coupling parameters represent measures of the same kind. For instance, all the elements of the lift distribution are expressed in units of force per unit length, and each element of the twist distribution is measured in identical units of displacement. Second, the value varies smoothly, continuously, and as a function of geometric position. These qualities may be exploited to reduce the bandwidth of the coupling while effectively communicating the relevant information. This is distinct from vectors of information whose elements are unrelated to one another. This chapter is specifically not about reducing the bandwidth of coupling in general but rather only that coupling which is of a particular kind.

4.2 Reduced Basis Modeling

In many instances, methods for reducing high bandwidth information to a smaller set of parameters are obvious. For example, the scalar value for the longitudinal position of the center of gravity of an aircraft is, in fact, a low bandwidth representation of the center of gravity locations of each of the various aircraft subcomponents. Using the center of gravity location of the overall aircraft in lieu of all the subcomponents accurately communicates the necessary information required by the aerodynamics group to compute trim using a reduced number of parameters. However, as will be shown, reduced basis models for other kinds of high bandwidth interactions, such as the loading and twist coupling in aeroelasticity; are less obvious.

4.2.1 Measure of Merit

Coupling bandwidth is a measure of the number of parameters shared between disciplines—not the amount of information. It is possible to convey the relevant information in a smaller set of parameters. The twin goals of a successful reduced basis approach are:

1. Reduction in the order of the coupling vector
2. Accurate communication of relevant information

In other words, we wish to use only the number of parameters needed to model the relevant information accurately and nothing more. Of course, the measure of what is relevant and accurate will depend on the particular problem and in some cases is not obvious.

4.2.2 Model Requirements Define Appropriate Basis Method

Imagine we have the same airplane design dependency diagram as pictured in Figure 3.9. Only now the analyses used in the aerodynamics and structures disciplines are a full potential panel code and finite element method.

Finite element structural analysis and aerodynamic panel methods model the wing with a grid of elements. Figure 4.1 shows a sample of the aerodynamic grid of a wing generated by an industry standard panel method^{[118][119]}. Pressure values are computed at the center of n_c chordwise and n_s spanwise panels.

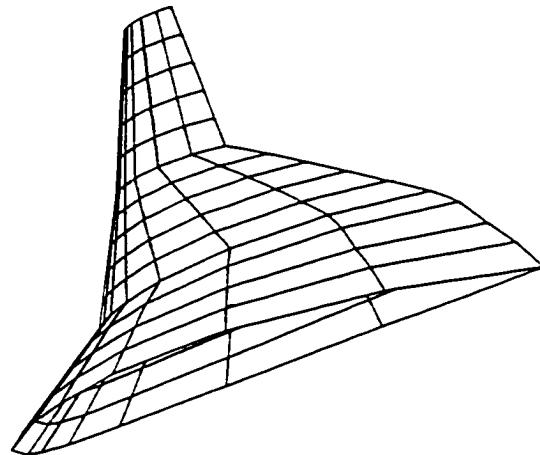


Figure 4.1 A502 Panel Model of Wing

Similarly, Figure 4.2 shows the finite element model of the same wing modeled using an industry standard FEM code^{[116][117]}. This particular code requires that loads are applied at each panel corner point. This is also where the FEM code computes deflections.

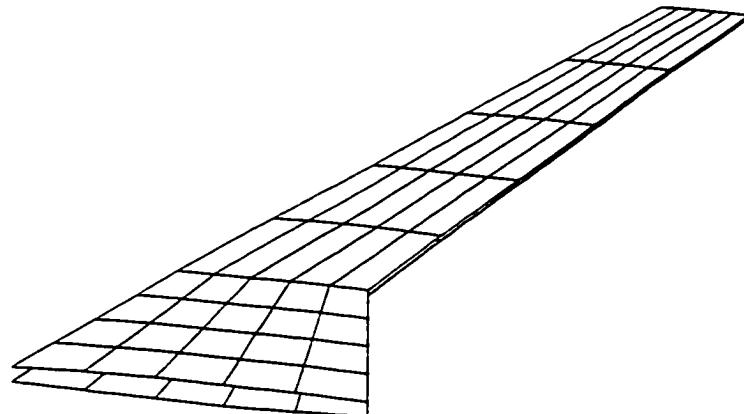


Figure 4.2 FESMDO structural model of generic wing

Absent a reduced basis modeling approach, the chordwise and spanwise matrix of pressure distributions creates a $n_c * n_s$ number of parameters to be passed between

the aerodynamics and structural analysis. This is an example of a very high bandwidth interdisciplinary coupling. It is important to remember that even in a single level approach, where the loading is directly passed between analyses, the different grid topologies required transforming the panel centric pressure values to corner point loadings.

4.2.3 Modeling a 2-D Pressure Distribution

The coupling of a full panel method and FEM code, described in the previous section, represents a very high bandwidth case. One approach to reducing the order of the problem is by first modeling the chordwise force distribution and then the spanwise distribution.

The chordwise force distribution used in the structural analysis must accurately represent the torque at that section. The structural box has a different chord than the aerodynamic section, thus we would expect there to be differences in the shape; what is important is that the torque, which plays a first order role in load induced twist is accurate, and the distribution is close enough to ensure first order accuracy in the sizing of individual structural elements. Thus, a bi-linear, or in the case of a supersonic section, a linear, force distribution may suffice. In highly swept wings, where the majority of torque results from the outboard force distribution, the particular chordwise basis function chosen is even less important. For the same reason, accurate modeling of the spanwise information is very important.

This assessment of the quality of the basis function is problem dependent. In cases where the lift distribution looks nothing like the assumed model, stalled sections for instance, the assumed basis function may be very poor. The definition of the basis function must be made with understanding of the expected shapes and if the computed values differ greatly it must be caught by the disciplinary experts.

4.2.4 Fitting the Spanwise Force Distribution

In the aircraft design problem represented by Figure 4.3 the spanwise lift distribution is computed at twenty spanwise locations. This lift distribution may be represented using the first two odd terms of a Fourier series ($n=1, n=3$):

$$l = \kappa(A_1 \sin \theta + A_3 \sin 3\theta) \quad \text{Eqn. 4-1}$$

We choose the odd terms because we know our lift distribution is symmetric. Figure 4.3 shows the optimal lift distribution defined by the solution of the panel code at twenty spanwise stations and the lift distribution based on the use of these two odd Fourier coefficients.

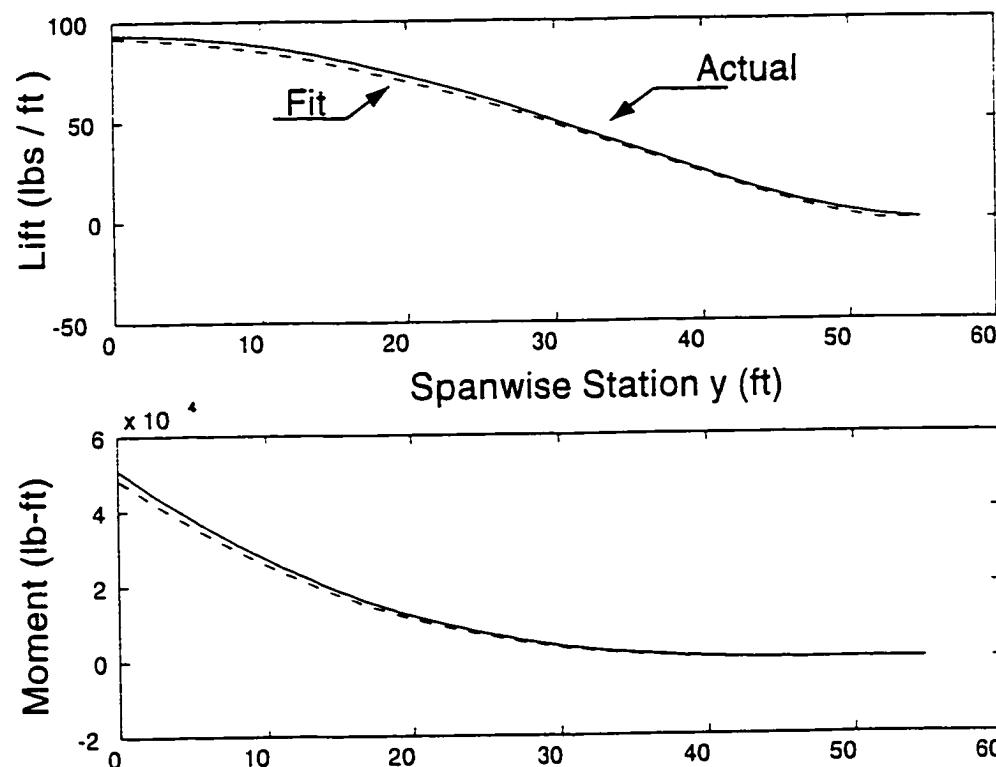


Figure 4.3 Lift distribution modeled with two Fourier coefficients

Fitting the spanwise force distribution with two Fourier coefficients yields a good model of the force. However, small errors accumulate and result in the off-set error for the integrated bending moment, seen in Figure 4.4. The quality of the reduced basis model for this particular problem is determined by how well it models the moment. Two fourier coefficients are insufficient to model a relatively complex spanwise lift distribution like that shown in Figure 4.4.

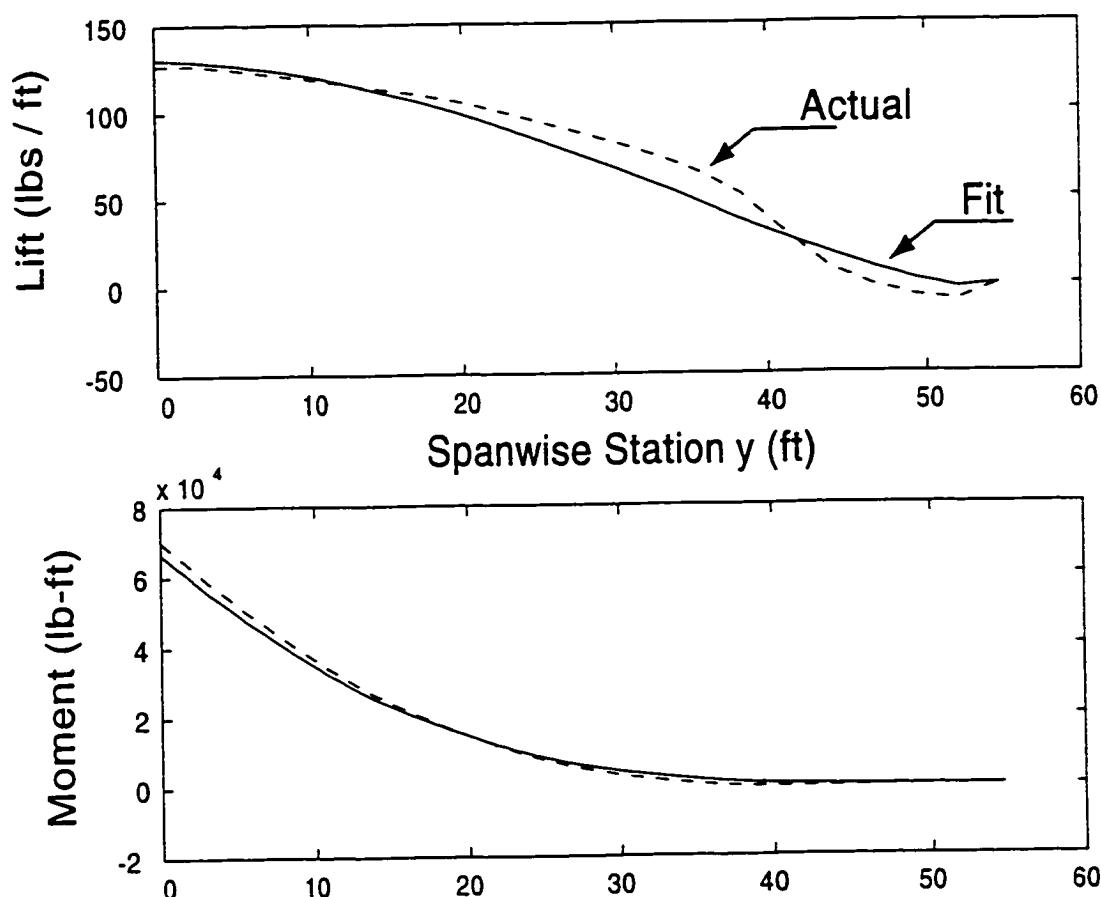


Figure 4.4 Approximated force and resulting integrated moment based on two Fourier coefficient model of lift distribution from 2-D panel code

4.2.5 Fitting the Spanwise Moment Distribution

Since we are interested in accurately reproducing the moment distribution, an alternate approach is to fit the spanwise bending moment directly. The force distribution may then be obtained by taking the second derivative of the moment. Several different basis functions are evaluated here for modeling the moment distribution.

If a cubic polynomial is used as a basis function, the aerodynamics discipline must integrate its computed force distribution and determine the bending moment at four spanwise locations. The moments at these knot points are then used to solve for the four unknown coefficients of the cubic polynomial. These four values then are used in lieu of the force distribution to describe the wing loading. As is shown in Figure 4.2, this reduced basis model very accurately models the moment distribution.

However, plotted below the moment distribution is the corresponding force distribution—the second derivative of a cubic is a straight line. This is a poor match to the actual force distribution which results in a poorly reproduced chordwise pressure distribution and a detrimental effect on the fidelity of the inboard torque values.

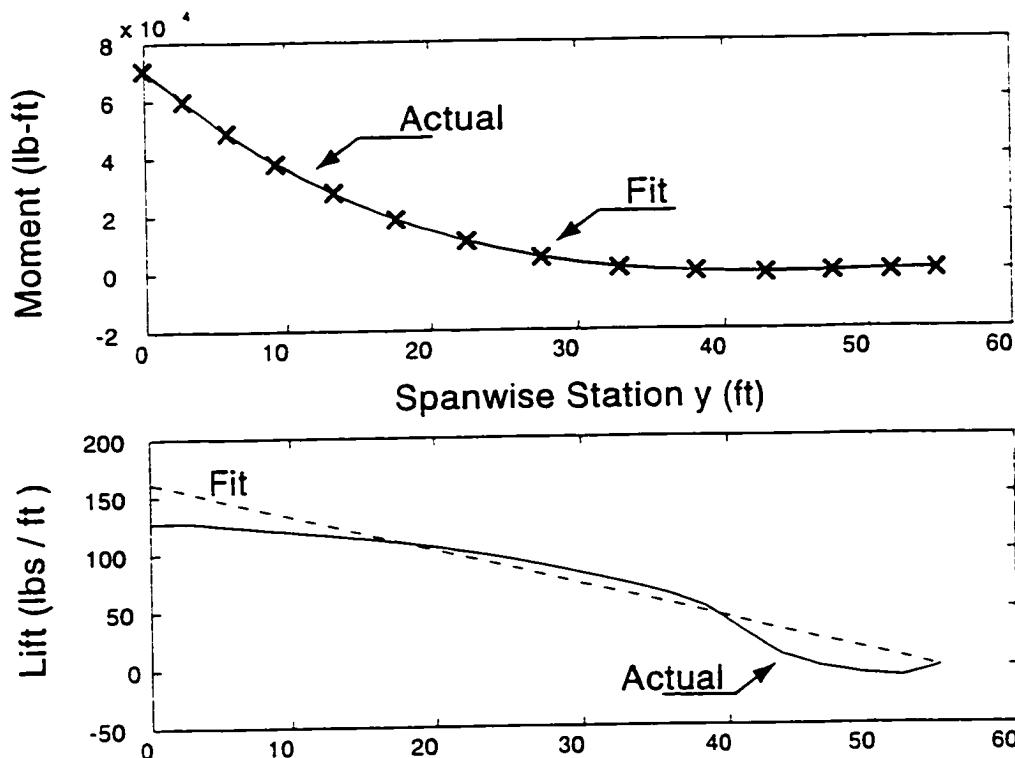


Figure 4.5 Cubic polynomial model of spanwise moment distribution and corresponding force distribution (d^2F/dx^2) and actual moment and force

A spline is a better candidate than a cubic polynomial because it is a piecewise cubic polynomial fit between each of the knot points. Thus, the second derivative is a piecewise linear force distribution that will more closely model the actual force distribution. The spline of the moment, shown in Figure 4.6, uses run-out boundary conditions that corresponds to the known physics of the lift distribution, the moment is zero at the tip, the slopes must match at the knot points, and the force at the root is known.

As shown in Figure 4.6 the spline basis function models the moment very accurately and the corresponding force distribution tracks the force distribution more closely than the cubic. A spline has the nice quality of passing directly through the knot points, thus if a knot point is placed at the root, it guarantees a model that

will correctly capture the often critical root bending moment. Additionally, the four parameters used to define the spline are not physically meaningless coefficients but rather the value of the moment at four spanwise locations.

Fitting the moment with a spline defined by a few knot points can result in a model that is not reproducible with strictly distributed forces. For example, integrating the force distribution shown in Figure 4.6 results in a moment distribution with considerable error. This is because the second derivative fails to capture concentrated forces that would be necessary to create the moment distribution. These concentrated forces could be added by looking at the first derivative and comparing it to the integrated force. However, concentrated loadings are not physical in a wing loading problem and beg for a better model.

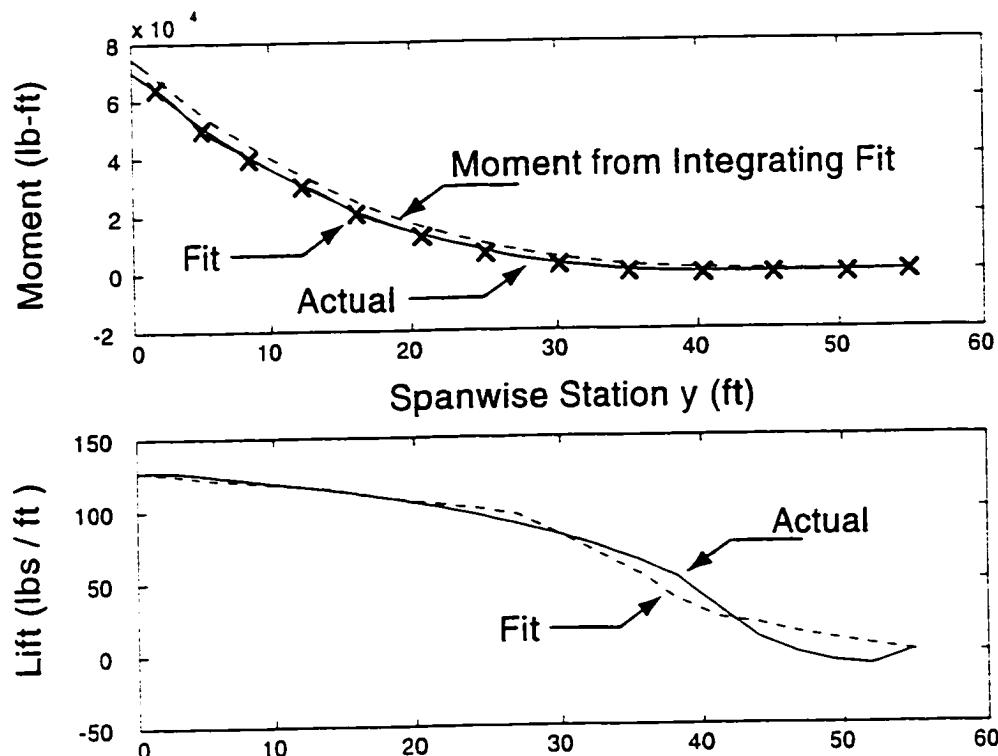


Figure 4.6 Spline model of spanwise moment distribution and corresponding force distribution (d^2F/dx^2) and actual moment and force

Attempts to correct the spline basis model by adjusting the boundary conditions can lead to some even more non-physical models. Consider for instance the use of a monotonic spline. The boundary conditions are adjusted so that the spline keeps the same sign of the 1st derivative between knot points, thus avoiding the concentrated tip loadings. But, to include this additional boundary condition the second derivative of the spline is discontinuous. As is shown in Figure 4.7, though the fit of the moment is quite good with a monotonic spline, the corresponding force distribution is wildly unrealistic.

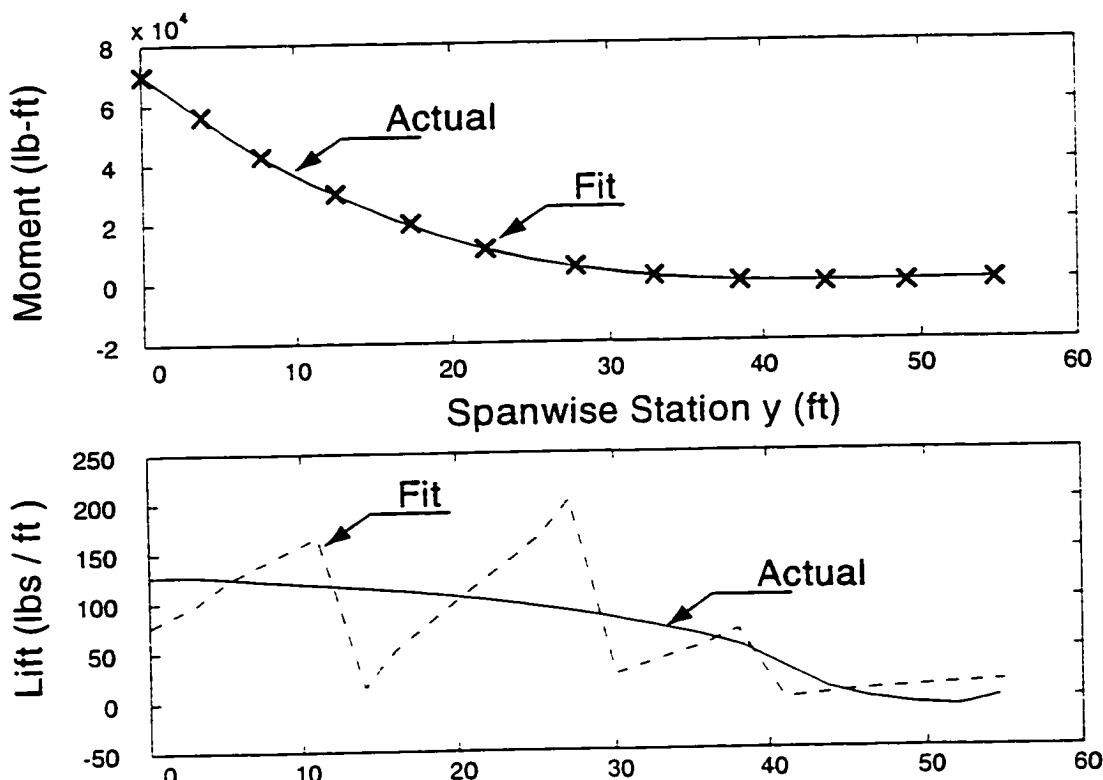


Figure 4.7 Monotonic spline model of spanwise moment distribution and corresponding force distribution (d^2F/dx^2) and actual moment and force

In short, this is an effective technique for modeling of the moments, but not for computing torque. This, in fact, is a fine method for application to the problem in Section 2.3.1; where the simple beam model used for stress analysis could have used the bending moment to determine the stress directly. Again, this emphasizes the point that the form the inputs are transformed into is the best measure of the merit of a basis function. In this case the moments would do fine.

4.2.6 Incompatible Models of Torque, Force, and Moment

The previous sections have shown difficulty in using a reduced basis model to obtain the force from a moment representation or vice versa. An alternate idea is to

independently model the moment, torque, and force. However, because the models will necessarily have errors it is virtually certain that for any particular spanwise location the value of moment, torque, and lift will be incompatible. The errors seen above result from attempting to get these forces from the moments, or in fitting the forces without regard to the fidelity with which their integral represents the actual moment distribution. This suggests the method of the next section.

4.2.7 Optimal Fit with Assumed Force Functional Form

The preferred fitting approach is to react to the observation that we are trying to develop the best model of the moment and torque distribution given a particular force distribution. The idea is to develop a force distribution model but evaluate the coefficients of that model so that the resulting error in the spanwise bending moment is minimized. In this way we obtain the a consistent force, torque, moment model while assuring that we obtain the best possible model given the constraint of the form of the force basis functions.

One such approach is to develop a piecewise cubic fit to the moment, much like the spline, by fitting a piecewise linear force distribution defined by the force per unit span at four spanwise knot points. This means finding the f values that solve the following problem.

$$\begin{aligned}
 & \text{minimize:} && \text{Error} \\
 & \text{with respect to:} && f_1, f_2, f_3, f_4 \\
 & \text{Error} = \sum_{i=1}^{N} (M_{\text{actual}} - M_{\text{computed}})_i^2
 \end{aligned} \tag{Eqn. 4-2}$$

This forms a linear system of equations that can be solved directly for the force values but by evaluating the moment at many spanwise locations. The resulting fit is the best possible piecewise linear force distribution for modeling the moments and because of the assumed force shape we are guaranteed a certain character of the force distribution.

In addition, this method lets all the spanwise moment information be used. The spline and cubic fits introduced earlier were developed by matching values at just a few spanwise locations. By Eqn. 4-2, the force distribution is made by comparing the computed moment at every spanwise location computed in the aerodynamics panel model with the generated fit.

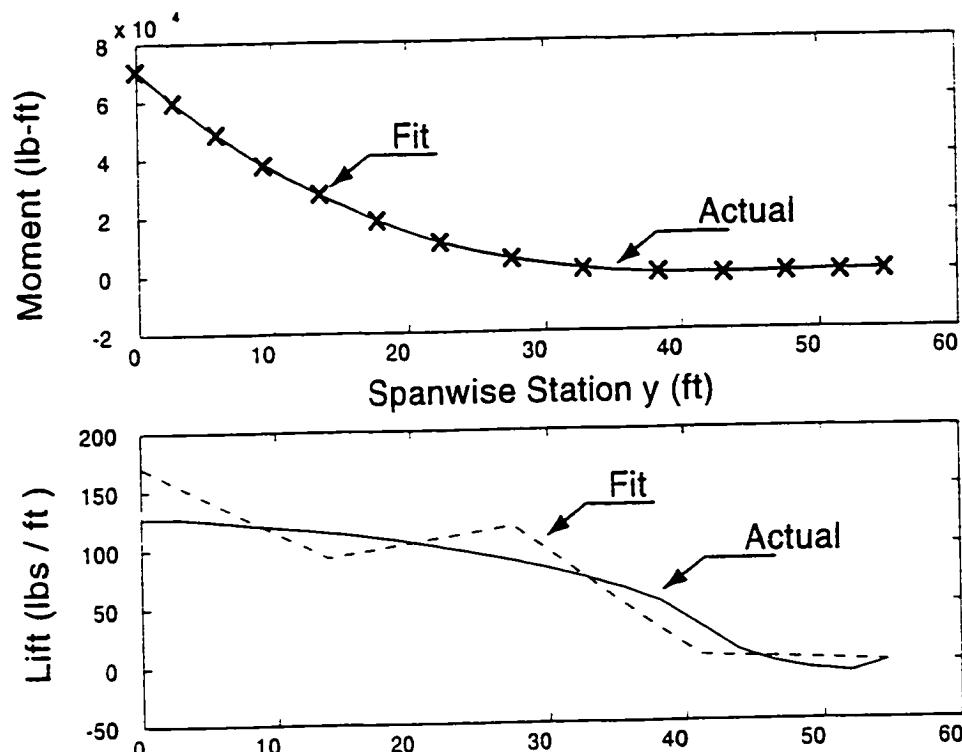


Figure 4.8 Piecewise linear model of force to satisfy Eqn. 4-2 and actual moment and force

The choice of a piecewise linear force distribution is somewhat arbitrary. For example, we could have chosen to model the force using Fourier coefficients; but unlike the previous application we would again evaluate the quality of the fit not in terms of matching the force distribution, but instead in matching the moment distribution.

$$\begin{aligned}
 & \text{minimize:} && \text{Error} \\
 & \text{with respect to:} && A_1, A_3, A_5, A_7 \\
 & \text{where} && F = A_1 \sin \theta + A_3 \sin 3\theta + A_5 \sin 5\theta + A_7 \sin 7\theta \quad \text{Eqn. 4-3} \\
 & \text{Error} = \sum_{i=1}^N (M_{\text{actual}} - M_{\text{computed}})_i^2
 \end{aligned}$$

The resulting match of the forces and moments is even better:

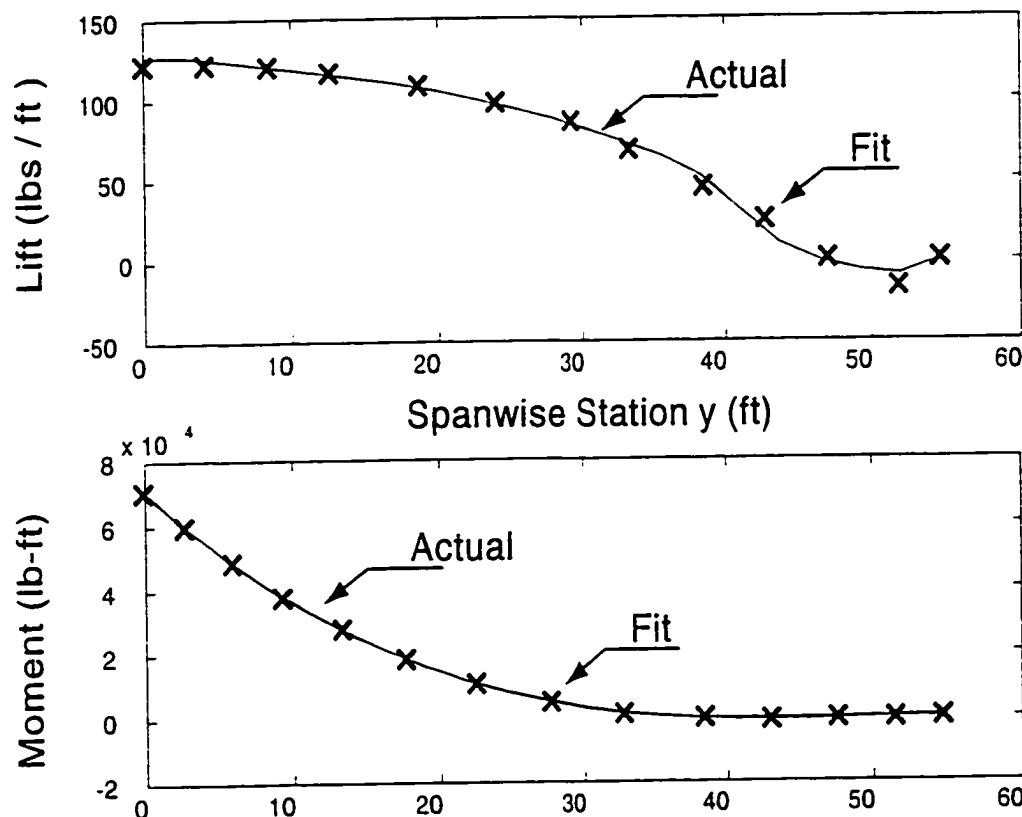


Figure 4.9 Fourier model of force to satisfy Eqn. 4-2 and actual moment and force

Though the use of fourier coefficients in this way does a very good job, it has the drawback of using parameters with little direct physical meaning.

4.2.8 Coefficient Meaning

The previous section shows that the best reduced basis formulation for this particular problem is the Fourier force series based on a measure of the quality of the resulting integrated force distribution. Fidelity in the quality of the reduced basis formulation is a primary goal but one that should be balanced with the meaning of the representation. The third and fourth Fourier coefficients, or the coefficients of a cubic polynomial have no inherent physical meaning. Compare this to the clear physical representation of the knot points in a spline (which are the spanwise moments), the first Fourier coefficient (which corresponds to total lift), or the four spanwise force values that are used in the least squares approach of the previous section. These physically meaningful quantities are useful in practical design where disciplinary designers wish to communicate a qualitative change in the parameter being modeled, or establish limits on these coefficients that relate directly to physical constraints (such as on maximum section lift coefficient or thickness). Changing the value of a spline knot point or one of the ‘f’ values of Eqn. 4-2 is like asking for unloading of the wingtips or increasing the camber. No such clear physical analog exists for these other formulations.

4.3 Summary of Approaches

The previous section demonstrates various approaches to generating a reduced basis model of high bandwidth information. The quality of one method versus another depends on how well the reduced basis method communicates the relevant information. Figure 4.10 summarizes the approaches evaluated in terms of a qualitative assessment of the error in the torque and moment. Clearly, the best approach is to transfer the high bandwidth information directly. But if a reduced basis model is used then the best strategy depends on which of these two parameters is of interest and what the input requirements of the analysis tools are.

For instance, in the simple problem of Section 2.3.1; the wing skin was sized using direct bending stress only by a formula that was a direct function of the moment. Thus, methods that correctly represented the moment distribution but not

the force distribution, like the monotonic spline, or cubic polynomial would be acceptable. However, for the case considered in Chapter 7 where the tools are a full panel code and FEM described in Section 7.3, the best approach is using the least squared representation of the forces: whether by a piecewise linear basis model or a Fourier series model.

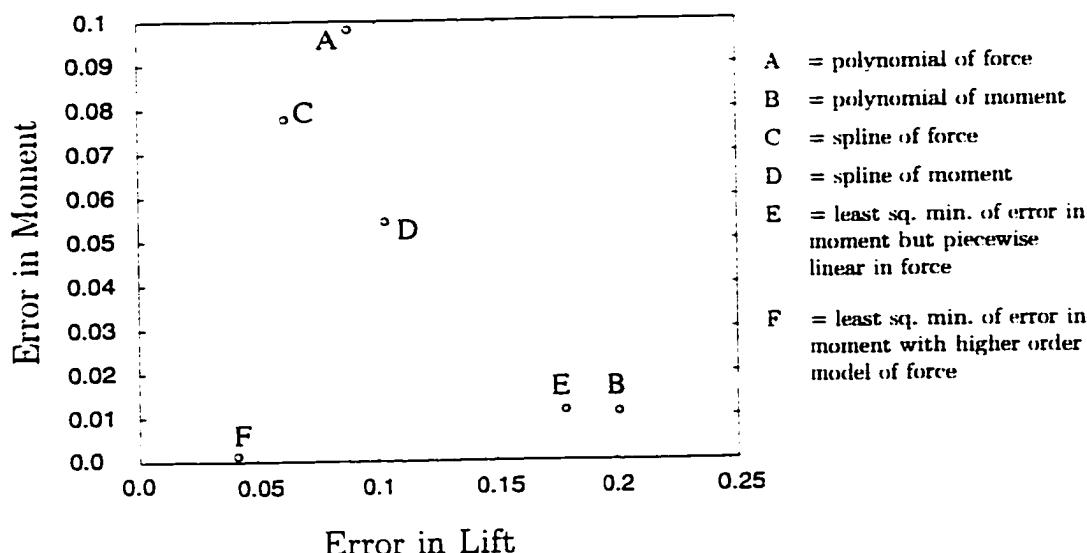


Figure 4.10 Normalized sum squared error of reduced basis strategies for given loading

Chapter 5

Using Response Surfaces in Collaborative Optimization

This chapter motivates and describes the use of response surface estimation with collaborative optimization. Response surfaces are used here in a novel way, not to model the functional analysis, but rather the disciplinary subproblem optimization results. The formulation of collaborative optimization described here represents a unique modification to the basic method and is demonstrated to improve its convergence performance and applicability. In subsequent chapters, this first implementation of response surfaces is developed further by improving the response surface generation techniques and applying the method to several example design problems.

5.1 Response Surfaces in Design

Many analysis codes are also extremely time consuming to execute. For example, [74] reports that one evaluation of a rotor blade aeroelastic simulation takes six hours to solve on a dedicated Cray Y-MP. In such cases, computation of first order sensitivity information via finite differences becomes impractical and one is reliant on derivative free search methods. Because of the expense in evaluating the function the convergence rate of the method selected becomes extremely important. Popular methods such as Nelder-Mead's simplex are not useful in such cases because, though robust, they have relatively slow convergence^[75].

A second challenge in the use of numerical simulation in design optimization are artifacts of the numerical simulation on the digital computer. Discretization of models, errors in precision, and incomplete convergence of iterative solutions can lead to errors in computed results. These errors differ from physical experiments in that the error is not random but purely deterministic and repeatable given the same inputs^{[82][83]}. However, the result is high frequency oscillations in the computed result. The practical effect of this phenomena is that an optimizer can become trapped in local minima of the problem; missing the global trends in the computed parameters^{[76][77]}.

A response to both of these observations concerning numerical simulation is the introduction of an approximate function as a substitute for the full analysis. This is an old practice in engineering applications^[78]. The use of approximation methods has been applied within several MDO methodologies. Response surfaces were used extensively as surrogate models in designing an HSCT in a variable complexity algorithm^[31]. Use of neural net based approximations has helped improve the efficiency of the CSSO algorithm^{[85][87][86]}.

Response surfaces refer to a class of low order mathematical models that predict the response of an unknown functional relationship. The functional relationship is generated on the bases of a set of computed results correlated with different input variable values. Response surfaces have several properties that make them attractive for use with optimization, in general, and CO in particular. They are:

- *computationally very inexpensive to evaluate*
- *represent noisy analysis with an inherently smooth model*
- *a natural way of implementing coarse-grained parallelization*

Because the response surface itself is computationally inexpensive to evaluate, the need for fast converging optimization algorithms is less important than the algorithm's robustness^[88]. The smoothing effect can have an important utility in avoiding locally optimal results that belie the general trend of data. Finally, where multiple processors exist, the response surface may be generated in parallel by solving the analysis for each different input set of variable values. In real-time then, a

response surface model of a large domain may be created in the same time as the analysis at a single point.

An obvious disadvantage of response surface use is that analytic approximations of unknown functions may be inaccurate. When used in lieu of actual analysis, the fidelity of the response surface fit must be considered. As will be seen in subsequent sections, when incorporated into collaborative optimization architecture, the response surfaces will be evaluated relative to the actual function being modeled and new response surfaces will be generated as necessary.

5.2 Using Response Surfaces in CO

In collaborative optimization, multiple processors can be used to execute the subproblem optimization for different target design variable vectors. If sufficient computational resources exist, a full response surface may be generated in the time required for a single subproblem optimization. The conventional CO formulation (i.e. the analysis coupled directly with an optimizer) requires that a particular subproblem optimization be executed serially^[64].

Once generated, the response surfaces are inexpensive to evaluate, and the slow convergence behavior described in Chapter 3, becomes less computationally expensive. The extra evaluations resulting from poor conditioning of the Jacobian, are performed on the relatively inexpensive response surface model, rather than on the subproblem optimization itself.

Finally, because the response surface is a model over a relatively large domain of the design space, errors in the points used to define the model have a smaller effect on model quality than on the post-optimal sensitivities at any particular point. This allows the subproblem optimizations to be converged to a slightly looser tolerance than is required in the basic formulation.

5.2.1 Modeling the Subproblem Design

Response surfaces are typically used to model disciplinary analyses^[70]. Figure 5.1(b) illustrates using response surfaces in this way within a CO method. A response surface of the subproblem analysis is generated by solving the analysis for a design vectors, $\{x\}$, and obtaining a corresponding set of computed state variables, $\{y\}$. This process is repeated for as many different vectors $\{x\}$ as are needed to solve for the unknown coefficients of the response surface model. The subproblem optimizer uses the response surface approximation, in lieu of the analysis, to minimize the subproblem objective function (e.g. system level compatibility constraint), J . This approximated value of J^* is passed back to the system level optimizer.

Use of response surfaces in this manner presents difficulties for highly nonlinear problems or for problems of high dimensionality. To generate even a relatively low order quadratic response surface requires $O(n^2)$ function evaluations:

$$F(x) = C_0 + \sum_i^n C_i x_i + \sum_{1 \leq i \leq j \leq n}^n C_{ij} x_i x_j \quad \text{Eqn. 5-1}$$

Where the number of unknown coefficients C is equal to $(n^2+3n+2)/2$. Typical design analyses often have hundreds of variables, resulting in an impractical number of required function evaluations^[68]. This problem becomes even more acute if the response surface model is to be regenerated at a new design point.

An alternate approach is demonstrated in Figure 5.1(c). The response surface is generated by solving the subproblem optimization for a target variable design vector, $\{z\}$, and obtaining a corresponding subproblem objective function value J_i^* . The subproblem is evaluated for as many different vectors of $\{z\}$ are required to solved for the unknown coefficients of the response surface model of J_i^* . The system-level optimizer then uses this response surface, in lieu of the subproblem optimization, to approximate J_i^* as a function of the target vector, $\{z\}$.

Collaborative optimization problems are already divided along disciplinary boundaries that require relatively little interdisciplinary communication. The width

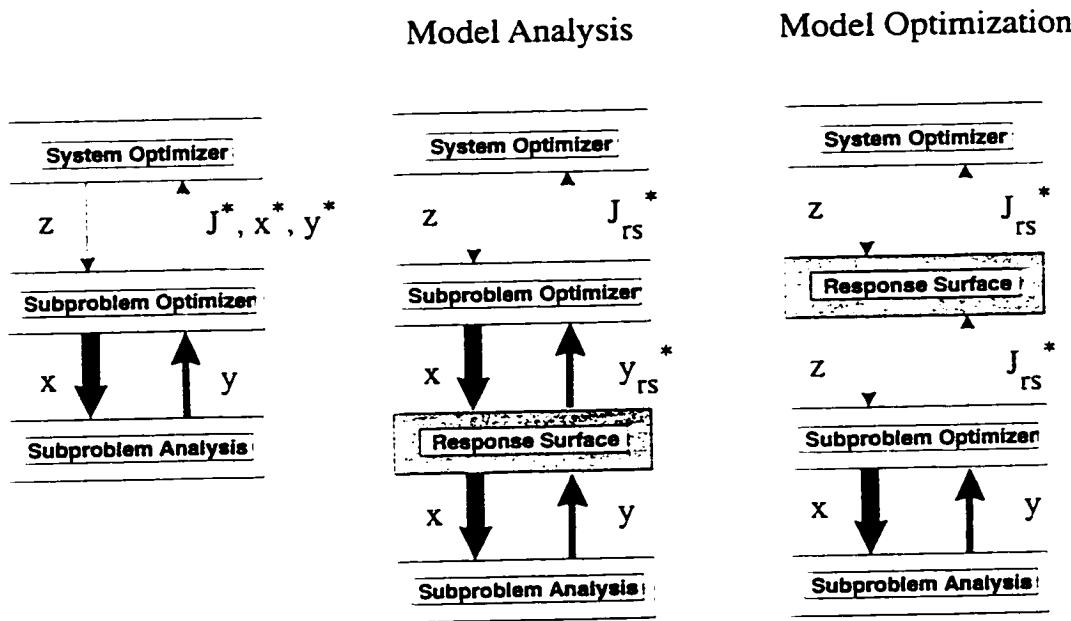


Figure 5.1 (a) Standard collaborative implementation, (b) RS models analysis, (c) RS models optimization

of the arrows in Figure 5.1 suggest the relative size of problems well suited for CO. Although the design subproblems may include hundreds of design variables $\{x\}$, they typically include a much smaller number of interdisciplinary variables $\{z\}$. Thus, by using response surfaces as in Figure 5.1(c) the dimension of the fit is more manageable.

5.2.2 Model Quality

Nonlinearities in the model being represented will cause error between the value predicted by the response surface model and the actual value of the analysis. In some applications, successful computation of nonlinear solutions was found using a single response surface^[76]. However, in general it is anticipated that the response surface needs to be refined as the design point changes and moves away from the original region. Shown in Figure 5.2 is the actual variation in the subproblem objective function taken from one of the design subproblems of the sample problem of Section 5.3. Also shown on this figure is a plot of the variation in the predicted value of J^* based on a quadratic model of the subspace. This plot demonstrates the

quality of the quadratic response surface model for the given problem; and the limitations of using the model for extrapolation outside the domain in which it was developed.

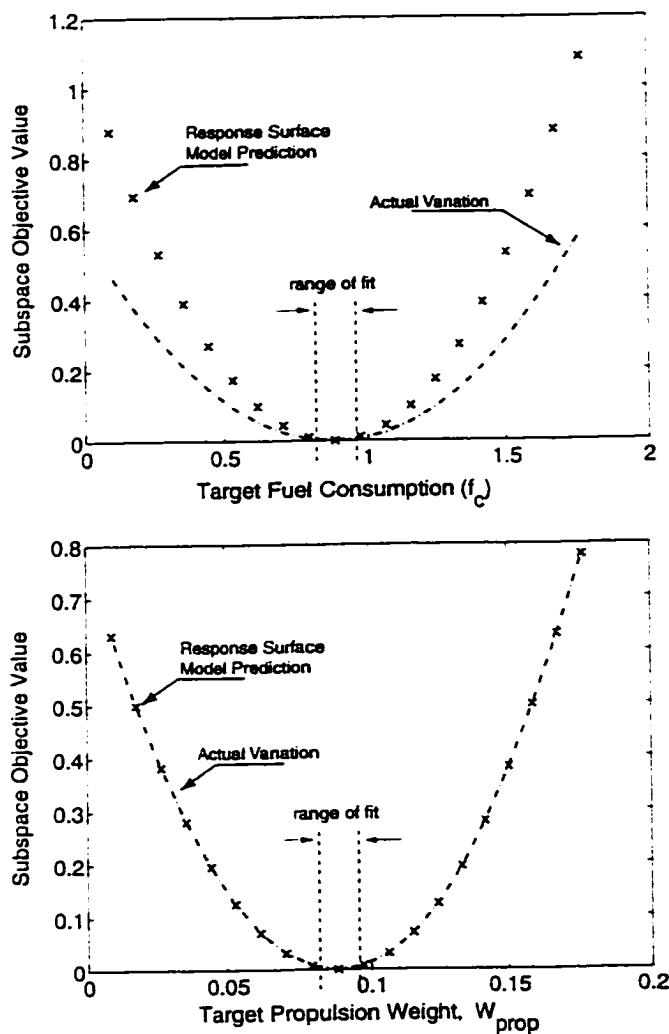


Figure 5.2 Variation in subproblem function and quadratic response surface model, from ship problem (Section 5.3) propulsive design.

In general, the error of the response surface decreases within this domain as it becomes smaller. What is needed then is an algorithm that controls the size of the domain of the response surfaces, controls over what range they may be used to

model the subproblem optimizations, and defines how they should be refined. Such an algorithm, borrowing ideas from trust regions, is described in the following section.

5.2.3 Management of Response Surface Updating

Trust regions operate in a single optimization iteration much as move limits; which serve to restrict the extent to which the optimizer may adjust one or more design variables^[83]. Move limits however tend to be specified in an *ad hoc* manner by the person directing the optimization. Trust regions refer to the class of optimization algorithms that use a systematic methodology for specifying changes in the size of the move limits^[34]. General convergence has been proven for some methods based on trust region approaches^[92].

Trust regions are well suited for use with response surfaces because one of the intrinsic steps in generating a response surface is the evaluation of a set of points in the vicinity of the current design point. The fidelity of the resulting response surface is a function of the distance between these points and the non-linearity of the actual function being modeled. By using the trust region dimension to specify the distance between these points we create a systematic method of adjusting the fidelity of the response surface model to account for poor predictive behavior in regions of large non-linearity.

A general trust region algorithm to minimize the function F is as follows.

- a. Specify an initial design vector $\{z\}_k$
- b. Define a region of trust Δ
- c. Initialize a set of m design vectors $\{z\}_i$ within the trust region: $\|\{z\}_i - \{z\}_k\| < \Delta$
- d. Evaluate the analysis for each vector
- e. Solve for the unknown coefficients of the model to generate $f_{rs}(\{z\})$
- f. Find the minimizer, $\{z^*\}$, of the model $f_{rs}(\{z\})$
- g. Evaluate the actual analysis at $f(\{z^*\})$
- h. Compare $f(\{z^*\})$ with $f_{rs}(\{z\})$ and determine if $\{z^*\}$ is an acceptable point
- i. Adjust trust region size and/or initial point ($\{z\}_{k+1} = \{z^*\}$) and repeat from c

Trust region algorithms generally differ from one another in how they perform items (c), (h), and (i) in the above algorithm. Design of experiments (DOE) methods define a procedure for item (c). Often the challenge is to sample a large dimensionality space with a relatively small number of function evaluations while minimizing the effect of experimental noise. Computer simulations, such as those we are concerned with here, are less sensitive to this kind of noise but DOE methods are still useful in ensuring that evaluated points are linearly independent for a particular form of RS model^{[69][82][83]}. As will be shown in the next chapter however, special features of the collaborative architecture may be used to reduce the number of points required to generate a quadratic response surface model to the point that linearly independent sets of data are easily obtained.

Many methods exist for determining the quality of a predicted minimizer and adjusting the trust region size. Most are based on a measure of the accuracy of the response surface fit in predicting the variation in the actual function^{[70][71][72][73]}. Three update approaches were considered and differ from each other in the way steps (h) and (i) are performed. The first accepts the new point only if the actual objective function value at this point is better. That is, if $f(\{z^*\}) < f(\{z\}_k)$ where z^* was found on the basis of the response surface model and $\{z\}_k$ was the point around which the model was built (as per step a), then accept $\{z^*\}$ as the new point, $\{z\}_{k+1} = \{z^*\}$, and repeat the process. If $f(z^*) > f(z_k)$ then $\{z\}^*$ is rejected, the trust region is shrunk by 50%, $\Delta = \Delta/2$, and the algorithm proceeds from step (c). If the new point is accepted, $\{z\}_{k+1} = \{z^*\}$, the trust region size may still be modified. If $\{z^*\}$ is on the boundary of the trust region, then the region size is increased, $\Delta = 2\Delta$. Otherwise, the region is shrunk, $\Delta = \Delta/2$. This update algorithm was used successfully in a general second order optimization algorithm^[34].

The second approach examines the relative accuracy of the model in predicting the objective function value. This approach evaluates the new point using the actual analysis and determines how well the model predicted the unknown function. The trust region is then adjusted based on a ratio of the predicted to actual value. A general convergence proof exists for this approach but no claim is made about speed

of solution^[61]. Both this, and the previous method, adjust the size of the trust region symmetrically.

The third method is the trust region update scheme used in the three design problems described later in this text. The update algorithm follows the same rules as described for the first method except that the region is adjusted asymmetrically. If the new solution $\{z^*\}$ is accepted, and lies on the boundary of the trust region, then the region is expanded in only that dimension. Conversely, if the point is accepted but lies within the trust region, the dimension of the trust region is reduced in only that degree of freedom. All three algorithms performed well on a series of simple analytic problems.

5.2.4 Form of Objective Function

There is no guarantee that a $\{z\}$ exists for which $J_{rs}\{z\} = 0$ for all subproblems. The implication of this observation is that response surfaces can not be used with a system-level constrained optimization algorithm that requires intermediate feasible designs, such as SQP. Several alternatives are possible, but a simple external penalty function can be used to create an unconstrained system level problem:

$$F = -f + \sum_1^N w_i J_i \quad \text{Eqn. 5-2}$$

This function has the disadvantage of requiring the adjustment of weighting parameters relative to the merit function being minimized. However, because the function is quick to evaluate; it is possible to execute system level optimizations with different weighting parameters and discern the effect on the resulting optimal design point based on the composite objective function. This is the procedure used in the system level optimization problem in the remaining examples of this text.

5.2.5 Theoretical Performance

As was done in Section 3.1 we can evaluate the impact of the use of response surfaces in the collaborative architecture in simple terms. Figure 5.3 shows the proportion of different variables in the problem with response surfaces. As mentioned in Section 5.1, the response surface may be formed by executing the design optimizations in parallel. Doing so means that the number of consecutive function evaluations to generate a response surface of the i^{th} subproblem is:

$$[(n_{g_i} + n_{\text{aux}_i} + n_{l_i})^2 + (n_{g_i} + n_{\text{aux}_i} + n_{l_i})] \quad \text{Eqn. 5-3}$$

The total number of consecutive function evaluations needed to find a stationary point is given by:

$$N_{\text{cycle}}[(n_{g_i} + n_{\text{aux}_i} + n_{l_i})^2 + (n_{g_i} + n_{\text{aux}_i} + n_{l_i})] \quad \text{Eqn. 5-4}$$

Comparing this equation with Eqn. 3-5 we see that the number of function evaluations with the use of response surfaces will be fewer when:

$$N_{\text{cycle}} < (n_g + n_{\text{aux}}) \quad \text{Eqn. 5-5}$$

It is important to keep in mind that this theoretical comparison of the collaborative approach, with and without response surface estimation, does not account for the additional function evaluations required by the basic formulation to converge due to implementation sensitivity and ill-conditioning of the Jacobian.

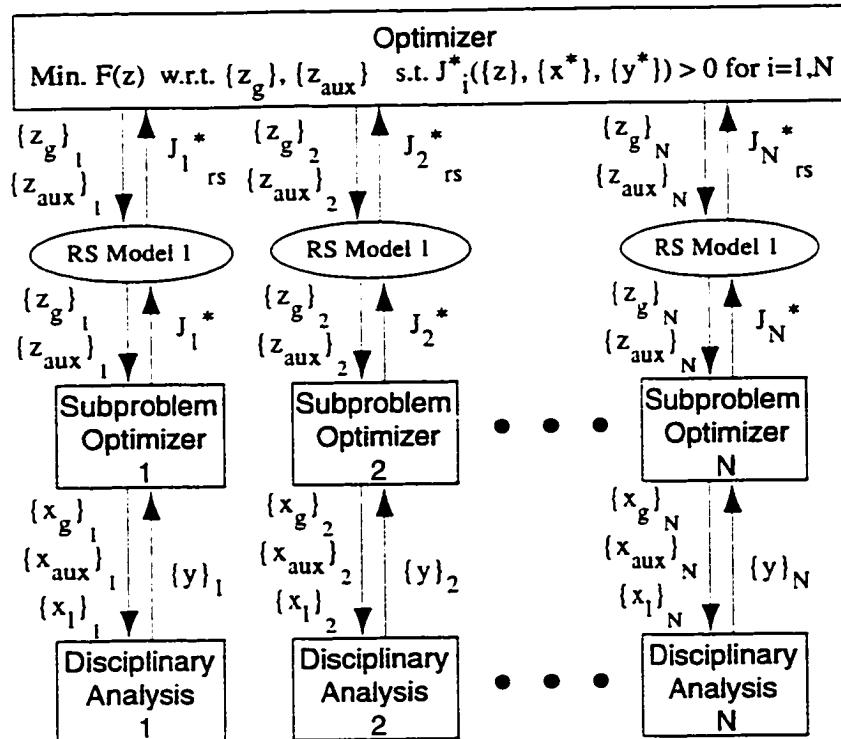


Figure 5.3 Collaborative optimization with response surface estimation

5.2.6 Convergence Tolerance and RS Models

Since response surfaces are a model of the subspace variation over a large range, variations in the value of J^* , relative to the actual variation, are expected. Thus, small variations in the value of the subproblem optimal design points used to generate the fit should not severely corrupt the model.

In [102], the response surface fitting approaches of the previous section were examined by modeling the performance disciplinary design space for an aircraft design sample problem and comparing the results. The response surface model was

generated by evaluating the subproblem at three different target design points $\{z\}$ and fitting the resulting local design $\{x^*\}$ and $\{y^*\}$.

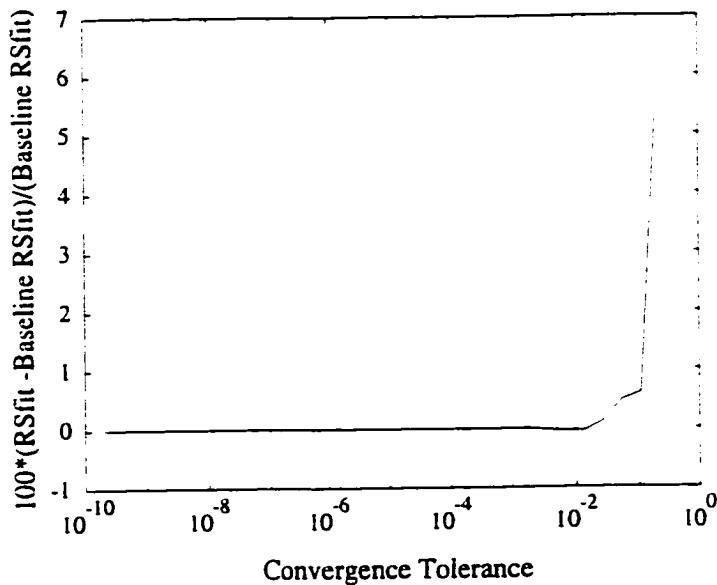


Figure 5.4 Variation in response surface shape as a function of subproblem optimization convergence tolerance

In Figure 5.4 the variation in the response surface model of this subproblem is shown as the convergence tolerance is varied from 10^{-10} to 10^{-1} . The 10^{-10} result serves as the baseline and the models were compared with each other at 750 points. As is shown in Figure 5.5, the resulting response surface model varies little over this range of tolerances while the number of function evaluations, required to obtain a converged result, decreases as the convergence tolerance is loosened. Similar experience with post-optimal gradients shows that the required convergence tolerance for good gradients is about one order of magnitude tighter than for a stable RS model.

Thus, use of RS models have the added benefit of allowing subproblem optimizations to be performed with looser requirements on subproblem convergence.

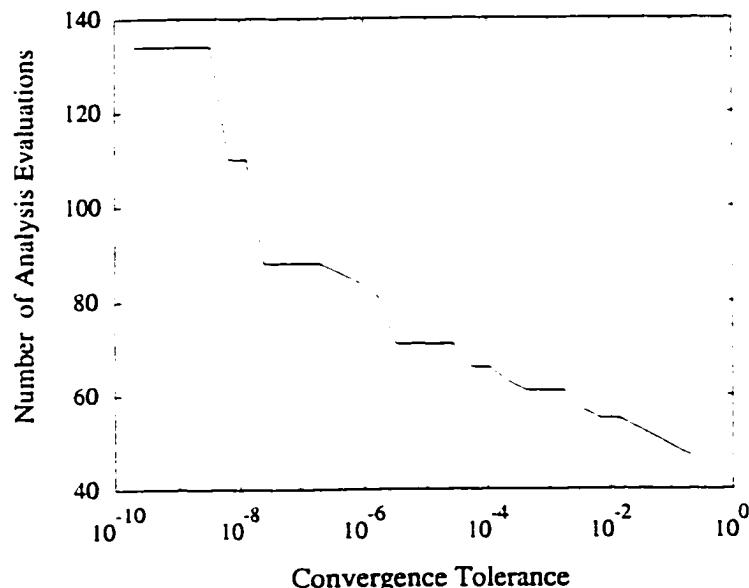


Figure 5.5 Number of analysis executions as a function of subproblem optimization convergence tolerance

5.3 Ship Design Using Response Surfaces in Collaborative Design

The methods of the previous section are applied to the design of an oceangoing oil tanker. This problem demonstrates the ability of this procedure to determine the correct optimal result and increased efficiency resulting from parallel development of response surfaces.

5.3.1 Nomenclature

HP = Actual effective horsepower

HP_i = Installed horsepower

W_p = Weight of propulsive system

C_p = Cost of propulsive system

f_c = Fuel consumption

- W_d = Displacement weight
- σ_l = Maximum direct stress
- σ_s = Maximum shear stress
- W_c = Cargo weight
- W_h = Hull weight
- V = cruise speed
- L = Keel length
- H = Height
- C_s = Ship cost
- ROI = Return on investment
- R = Cruise range

5.3.2 Tanker Design Problem

The four-discipline preliminary ship design problem was developed by the Lockheed corporation and used in an evaluation of their simulation based design architecture^[15]. The analysis tools used to model the ship performance are relatively simple analytic and statistical expressions used to approximate the results of more complex and exacting analysis tools. Figure 5.6 is an illustration showing four of the design variables that define the ship. In addition to these variables the installed horse power of the engine and the amount of fuel are included as free design parameters.

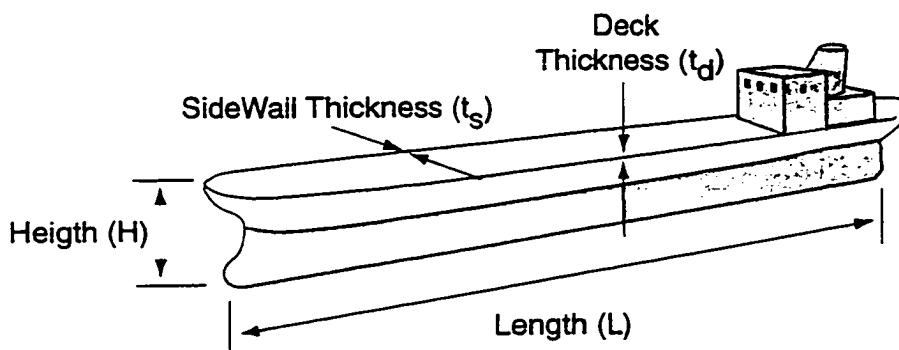


Figure 5.6 Conceptual ship design problem

Figure 5.7 shows the relationship of the various analysis modules used in the evaluation of the ship design. Computed quantities include the displacement weight, stresses in the hull under storm loading conditions, and range. The overall design problem is summarized by:

$$\begin{aligned}
 & \text{minimize:} && -\text{ROI} \\
 & \text{with respect to:} && L, H, W_f, t_d, t_s, \text{HP} \\
 & \text{subject to:} && W < 2(10^8) \text{ lbs} \\
 & && \sigma_c < 30 \text{ kpsi} \\
 & && \sigma_d < 30 \text{ kpsi} \\
 & && R > 10,000 \text{ nmi}
 \end{aligned} \tag{Eqn. 5-1}$$

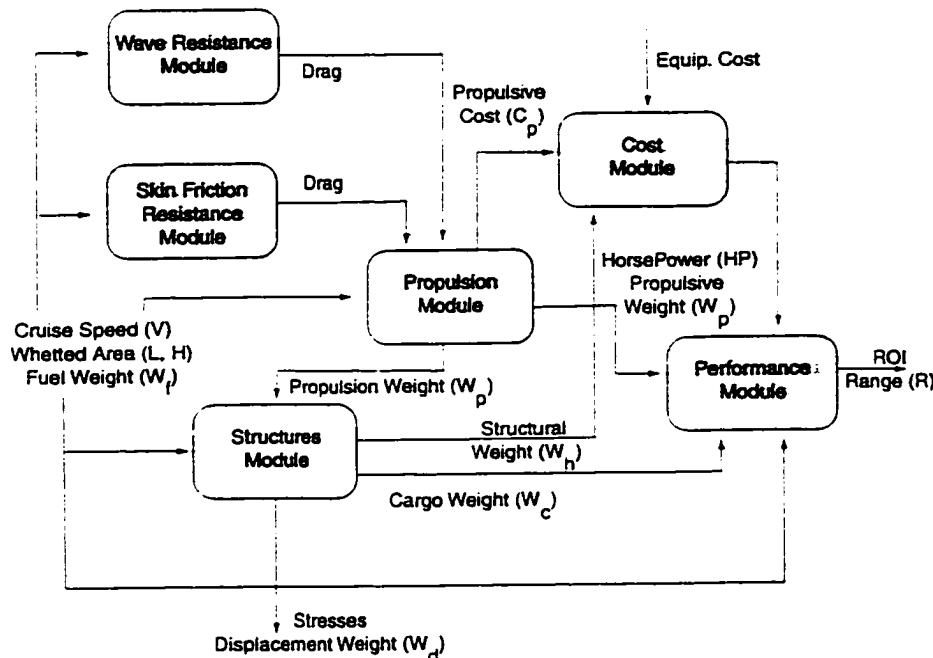


Figure 5.7 Conceptual ship design analysis flow

5.3.3 Basic Collaborative Implementation

The design problem summarized in Figure 5.7 is decomposed into four disciplinary groupings. Each of these groups has complete autonomy over variables and con-

straints that are strictly local. As a result, the structures design group alone manipulates the hull thickness and satisfies the stress constraints. Other variables, that were internal to the unified analysis, such as the propulsion weight or fuel consumption, become system level targets. This problem is expressed by the system level design problem and subproblem optimizations shown in Figure 5.8. There are 10 interdisciplinary target variables and four system level compatibility constraints. This is a poor problem to demonstrate efficiency advantages of the basic CO formulation because the problem has many more system level variables than local variables, while Section 3.2.3 described how problems well suited for CO have the opposite character. In addition, the analysis codes are very simple and fast running; enabling a solution to be found in a very short amount of time using single level optimization. Nevertheless, this problem serves as a ready example of the effectiveness of CO with response surfaces in determining the correct design point.

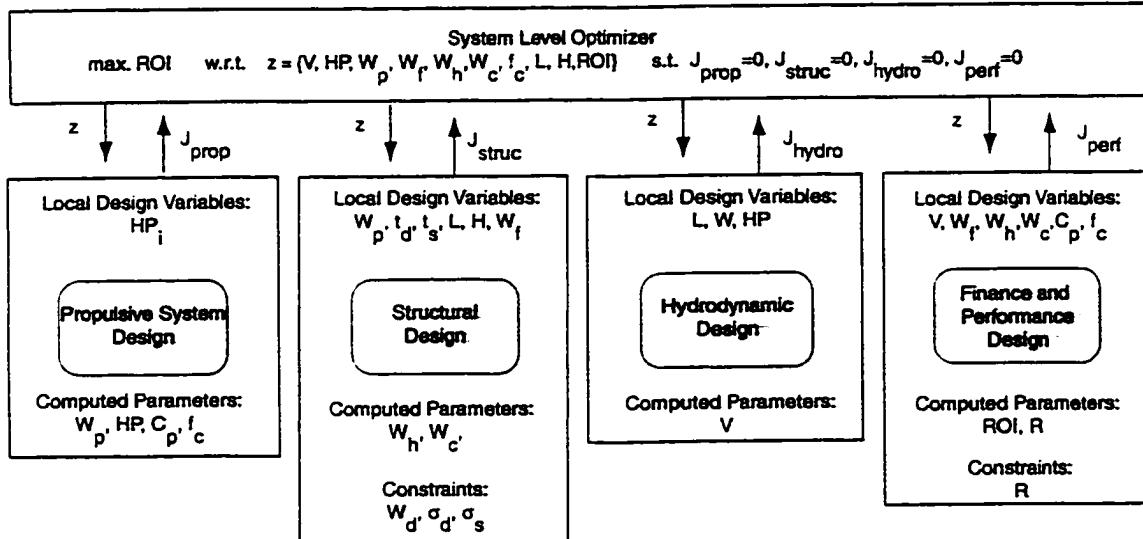


Figure 5.8 Standard collaborative implementation of ship design problem

5.3.4 Results Using Basic Collaborative Optimization

The tanker design problem was solved collaboratively as described in the previous section. The overall objective function converged on an optimal return on

investment of 0.277, the same result as that obtained using single level optimization of the unified analysis and obtained by others using different methods^[70]. Figure 5.9 and Figure 5.10 illustrate the convergence behavior of the basic collaborative formulation. As discussed in Section 3.4.7, the basic collaborative formulation is characterized by overshooting the optimal result and returning to it from the incompatible, or system level infeasible, side. Also characteristic is the large number of design cycles used by the system level optimizer to reduce the discrepancy between the local subproblem optimal designs to within the feasibility tolerance at the system level.

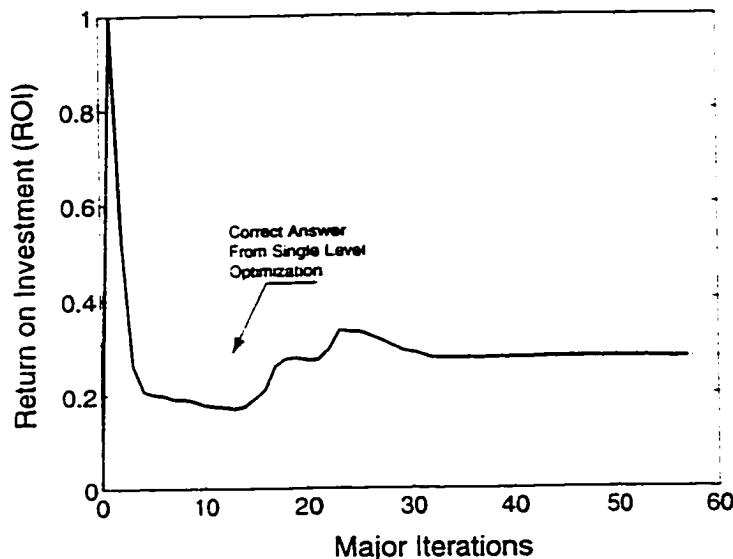


Figure 5.9 System level objective function, ROI, as a function of major iteration

Note that by the thirty second design cycle the correct objective function has been found but that an additional ten cycles are required to find a result that is within $10e^{-5}$

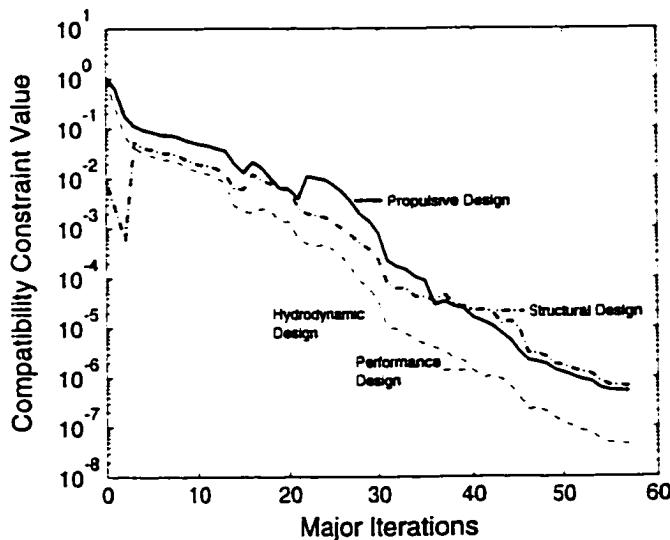


Figure 5.10 Compatibility constraint value as a function of major iteration

5.3.5 Ship Design with CO and Response Surfaces

Response surfaces are used as was described in Section 5.2.3 to represent the subproblem optimal objective functions of the four ship subproblem design optimizations. As shown in Figure 5.11, each response surface, once generated, provides a value of J^* as a function of the system level design vector z . As explained in Section 5.2.3, once an optimal value of F was found for a given response surface the corresponding z was evaluated using the full subproblem optimization and the results compared to the predicted value.

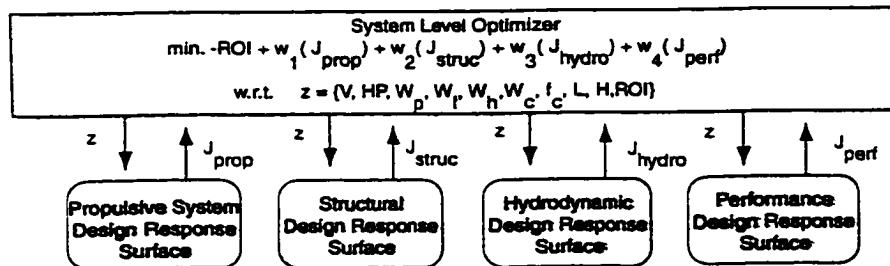


Figure 5.11 Use of response surfaces to represent ship optimal subproblem designs

Following the algorithm in Section 5.2.3 the response surfaces were updated and the trust region size modified. The procedure was repeated until the subproblem discrepancy function was reduced to within the given tolerance of 10^{-5} .

Below are the plots of system level objective ROI, Figure 5.12, and compatibility constraint value, Figure 5.13, for the solution of this problem. The correct objective function value is achieved in just three optimization cycles and the compatibility constraints also achieve a satisfactory tolerance. The stability of the answer is demonstrated by the following two design cycles that stay in the region of the correct solution while the trust region continues to shrink.

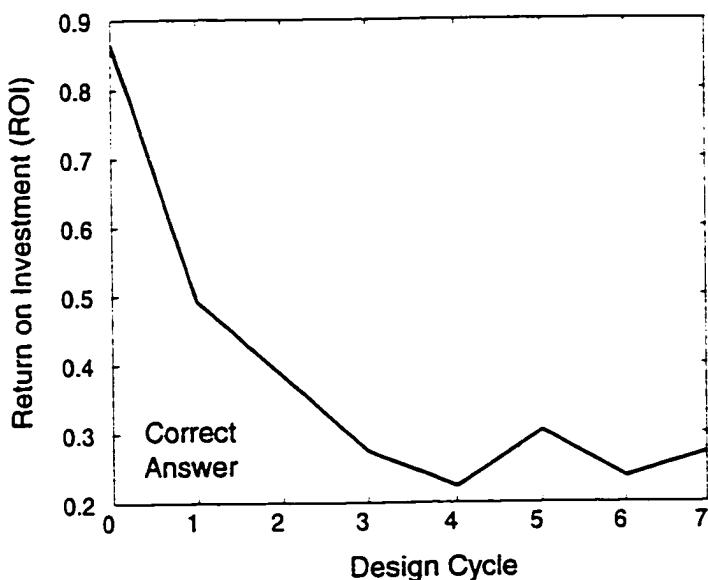


Figure 5.12 Target ROI vs. optimization cycle for tanker design using CO and response surfaces with trust region updates.

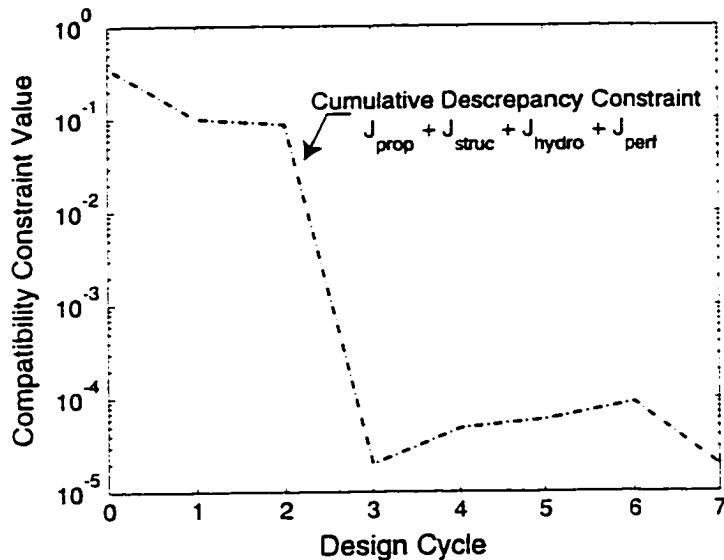


Figure 5.13 Compatibility constraint value versus RS update cycle for trust region using RS approximation

The cumulative compatibility constraint versus design cycle is shown in Figure 5.13. Comparing this plot with that of the target ROI value one sees that the compatible result is obtained within three cycles and a stable answer is achieved.

Figure 5.14 shows the values of some of the design variables used in the collaborative with response surface design of the ship. Included in this plot is the starting collaborative design point, the final design point, and the ‘correct’ point determined using single level optimization and an integrated analysis. Note that two of the design variables, the thicknesses, are local to the structures design problem but are system level design variables in the single level implementation.

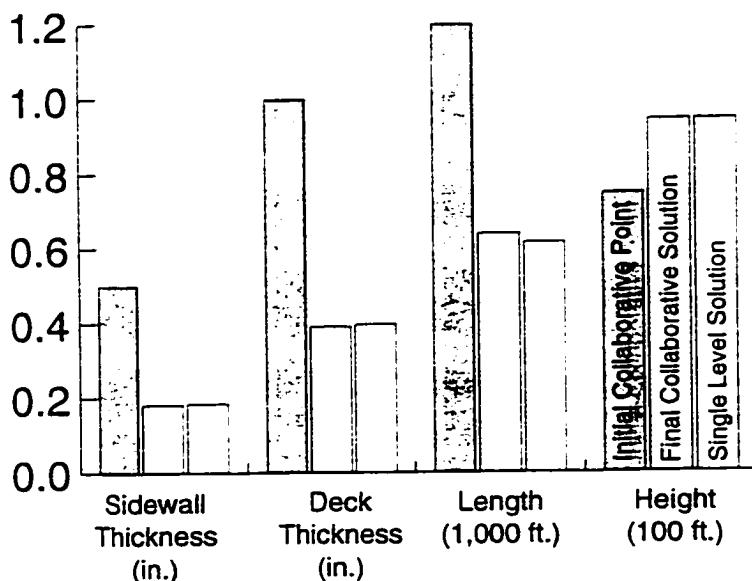


Figure 5.14 Initial and final design point for tanker design with collaborative optimization using response surfaces matches result from single level

If parallel processors were used to generate the response surfaces of each of the subproblem optimal results, the solution to the ship design problem using response surfaces would be found 10 times faster than with the basic formulation. That is, each major iteration of Figure 5.9 requires at least one complete subproblem optimization. Each design cycle, shown in Figure 5.11, requires one generation of a response surface.

5.4 Summary

This chapter has suggested that the system level optimizer use response surface models of the subproblem design to develop a new target design point. The relatively narrow interdisciplinary coupling that characterizes problems well suited for collaborative optimization make modeling the optimal subproblem objective function more efficient than modeling the subproblem analysis. However, results from even a simple problem show that it is unlikely that a particular quadratic RS model will be sufficient in obtaining a globally optimal solution, thus motivating a manage-

ment method for performing a series of response surface fits around progressively better points. Borrowing ideas from trust regions, a response surface updating algorithm was applied to a four subproblem ship design. This problem was successfully solved; finding the correct optimal solution in substantially fewer design cycles than the basic collaborative formulation.

Response surfaces provide several benefits in CO. The greatest benefit is that they facilitate parallelism in the design solution. A single, second order response surface can be solved in the same real time as a single design point; assuming only that there are enough processors to run the required number of points in parallel. Second, the response surfaces are extremely cheap to analyze and any artifacts of the system level optimization algorithm that may require excessive evaluations of the function will not require additional subproblem designs.

As a practical matter the number of processors is not unlimited and serious design problems at the sublevel may be constrained by other information management issues than the number of processors. Thus motivation exists to reduce the number of points required to generate a response surface. As discussed in Section 5.2.1 a general function requires more than n^2 function evaluations to form a quadratic model. However, the next chapter will demonstrate that the special form of the collaborative subproblem optimization, and implicit information in each design solution, may be exploited to reduce the number of required subproblem optimizations to as few as $(n+1)/2$.

Chapter 6

Response Surface Improvements

6.1 Modeling Optimal Subproblem Designs

This section develops the details of determining a quadratic fit to the subproblem optimal design results. Two approaches for approximating the subproblem results are presented and evaluated on a sample problem. The first directly models the subproblem optimal objective function, J^* , as a function of the target variable vector, $\{z\}$. Post-optimality gradient information is shown to reduce the number of required points to generate this fit by order of n . The second method models the optimal subproblem design vectors, $\{x^*\}$ and $\{y^*\}$, as a function of $\{z\}$, with a similarly small number of required optimizations. Several advantages of this second approach over the former are discussed. Finally, information inherent in each subproblem solution is exploited to reduce the number of required subproblem optimizations, using either method, by another factor of two.

As an example, consider a simple analytic aircraft performance design problem expressed by Eqn. 6-1 and Eqn. 6-2. This “performance” discipline analysis consists of Breguet’s range equation for propeller aircraft, Eqn. 6-2, which is a function of the initial to final weight ratio (W) and the lift to drag ratio^[108]. The subproblem optimization adjusts the local design variable, W , to minimize the difference between W and R , and the system level targets of W_o and R_o .

$$\text{minimize : } J = (W - W_o)^2 + (R - R_o)^2 \quad \text{Eqn. 6-1}$$

with respect to : $\{x\} = \{W\}$

where,

$$R = 0.325 \left(\frac{\eta}{sfc} \right) \left(\frac{L}{D} \right) \ln(W) \quad \text{Eqn. 6-2}$$

Figure 6.1 illustrates the variation in J^* as a function of target values of W_o and R_o , for fixed values of $\eta = 0.8$, $sfc = 0.8$, $L/D = 18$. The subproblem objective function is reduced to zero ($J^*=0$) only when:

$$R_o = 0.325 \left(\frac{\eta}{sfc} \right) \left(\frac{L}{D} \right) \ln(W_o) \quad \text{Eqn. 6-3}$$

The locus of points that satisfy this equation is shown in Figure 6.1. For all other values R_o and W_o , the minimum discrepancy function J^* is nonzero. Note that the contour values on this plot are nonlinear: each contour line indicates a J^* value double its predecessor. It is apparent that in this case the surface representing J^* is quite shallow in the vicinity of $J^*=0$.

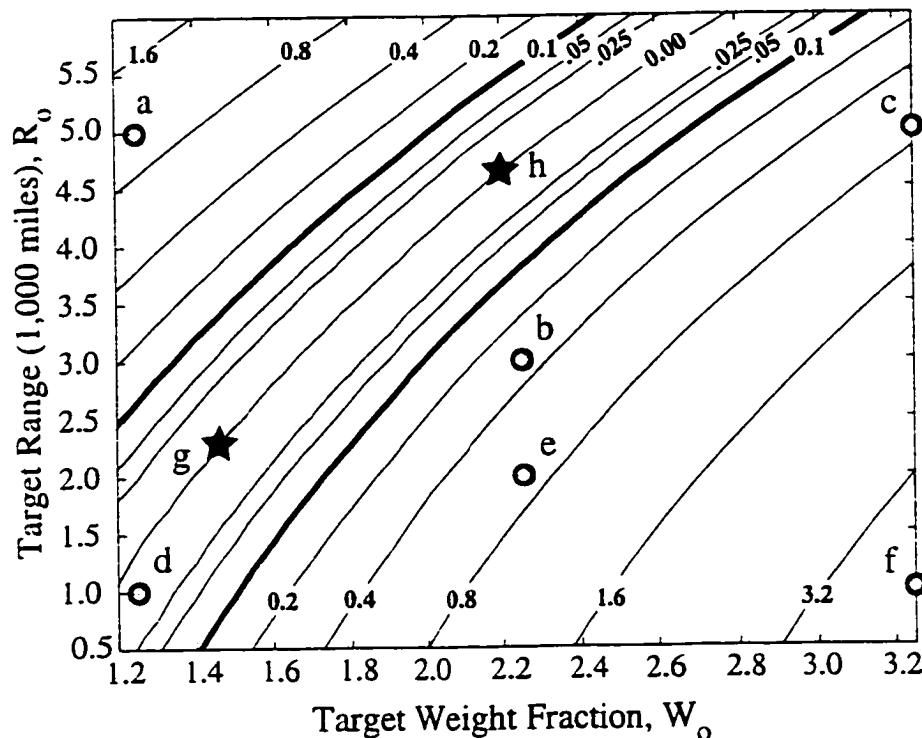


Figure 6.1 Value of subproblem optimal objective function J^* ,
Eqn. 6-1, vs. W_o and R_o

6.2 Estimating the Subproblem Objective, J^*

The general form of a quadratic response surface fit in n variables is given by Eqn. 6-1.

$$J^*(z) = C_0 + \sum_i^n C_i z_i + \sum_{1 \leq i < j \leq n} C_{ij} z_i z_j \quad \text{Eqn. 6-1}$$

The number of unknown coefficients in Eqn. 6-1 is given by: $m = (n+2)(n+1)/2$. To determine the coefficients of Eqn. 6-1 requires m linearly independent equations. These may be obtained by executing the subproblem optimization for m linearly-independent target design vectors: $\{z\}_1, \{z\}_2, \dots, \{z\}_m$.

The sample problem has two system level design variables: R_o and W_o . Thus six target vectors are required to generate a fit. The subproblem optimization of Eqn. 6-1 and Eqn. 6-2 was evaluated for the target points labeled a through f in Figure 6.1:

$$\{W_o R_o\} = \begin{matrix} \{1.25, 5.0\} & \{2.25, 3.0\} & \{3.25, 5.0\} \\ \{1.25, 1.0\} & \{2.25, 2.0\} & \{3.25, 1.0\} \end{matrix}$$

which formed the left hand side of the system of equations shown in Eqn. 6-3. The right hand side of this system is made up of the corresponding six optimal subproblem objective function values $\{J^*\}$.

$$\begin{aligned} [Z]\{C\} &= \{J\} \\ [\{1\} \{z_i\} \{z_i z_j\}] \{C\} &= \{J\} \end{aligned} \quad \text{Eqn. 6-2}$$

$$\left[\begin{matrix} \{1\} \{W_o\} \{R_o\} \{W_o R_o\} \left\{ W_o^2 \right\} \left\{ R_o^2 \right\} \end{matrix} \right] \{C\} = \{J\} \quad \text{Eqn. 6-3}$$

When this system is solved for $\{C\}$ and substituted into Eqn. 6-1, the following functional fit is generated:

$$\begin{aligned} J_{rs}^* &= 0.4061 - 1.3628 W_o \\ &+ 0.5544 R_o - 0.5544 R_o W_o \\ &+ 0.8787 W_o^2 + 0.0668 R_o^2 \end{aligned} \quad \text{Eqn. 6-4}$$

This fit is illustrated in the contour plot of Figure 6.2. The average sum squared error, $\frac{1}{p} \sum (J_{rs}^* - J^*)^2$, is 0.1005 as measured at 750 points evenly distributed over the range shown in Figure 6.1.

The fit is a reasonable approximation of the actual function. The region where it predicts J^* to be less than 0.1 (bold lines in all contour plots) roughly matches the similar region of the actual function shown in Figure 6.1. An important characteristic of this fit, however, is that it predicts negative values; as can be seen in the

upper right of Figure 6.2. The actual function can never become negative because of the form of Eqn. 6-1.

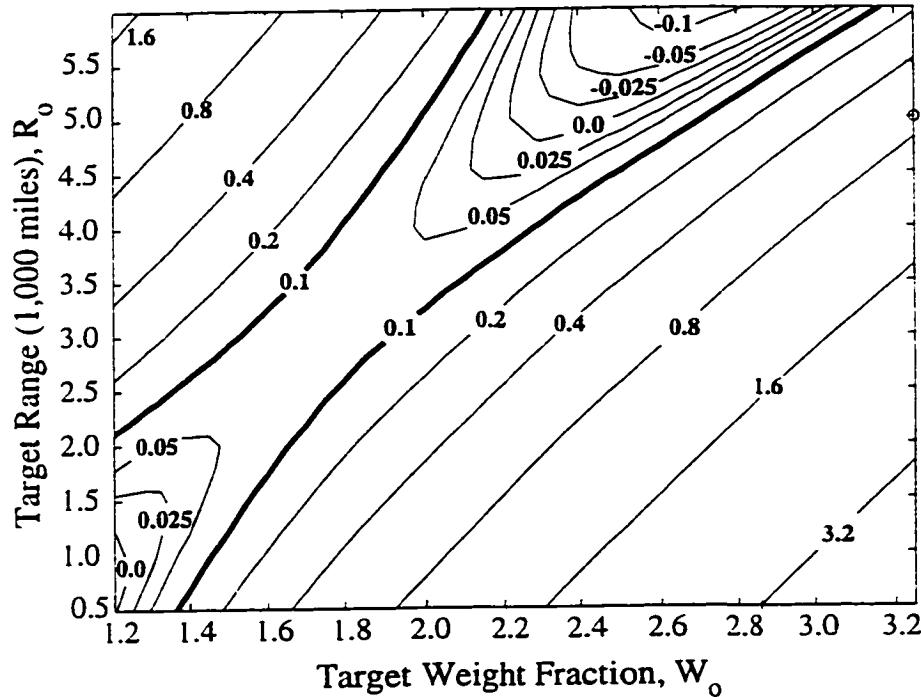


Figure 6.2 RS fit of J^* based on points a, b, c, d, e, and f from Figure 6.1

6.3 Fit Using Post-Optimality Gradients

The solution to a CO subproblem optimization determines not only the value of J^* , but also the derivative of J^* with respect to each target variable^{[54],[105]}. This additional information may be exploited to reduce the computational expense required to generate a response surface.

Each subproblem optimization result yields $n+1$ equations—the value of the subproblem objective function J^* and n derivatives. However, each pair of points contains one redundant equation for a quadratic model. In any particular dimension, x_i , we have two function values J^* and two derivatives dJ^*/dx_i , while a quadratic model is defined with just three pieces of information. The total number of

redundant equations, for m points, is $m(m - 1)/2$. The number of points that must be evaluated to solve for a quadratic fit using gradient information is the total number of equations minus the number of redundant equations:

$$m(n + 1) - m(m - 1)/2 = (n + 1)(n + 2)/2 \quad \text{Eqn. 6-1}$$

or

$$m = n + 1 \quad \text{Eqn. 6-2}$$

This is order n fewer points required to generate a quadratic model than was required by fitting functional values alone—eliminating the quadratic growth with n in the number of required points to generate a response surface.

Adding the gradient information to Eqn. 6-2 one obtains:

$$\begin{bmatrix} \{1\} \{W_o\} \{R_o\} \{W_o R_o\} \{W_o^2\} \{R_o^2\} \\ \{0\} \{1\} \{0\} \{R_o\} \{2W_o\} \{0\} \\ \{0\} \{0\} \{1\} \{W_o\} \{0\} \{2R_o\} \end{bmatrix} \{C\} = \begin{bmatrix} \{J^*\} \\ \{\frac{\partial J}{\partial W_o}\} \\ \{\frac{\partial J}{\partial R_o}\} \end{bmatrix} \quad \text{Eqn. 6-3}$$

Eqn. 6-3 has more rows than columns but is rank sufficient if m points are evaluated. The subproblem results were evaluated at points a, c, and e shown in Figure 6.1. The resulting fit is given by:

$$\begin{aligned} J_{rs}^* &= 0.12 + 0.32W_o + 0.32R_o \\ &\quad + 0.32R_o W_o + 0.32W_o^2 + 0.32R_o^2 \end{aligned} \quad \text{Eqn. 6-4}$$

which has a sum squared error of 0.8247. The contour plot for this fit is shown in Figure 6.3 and, as suggested by the relatively large sum-squared error value, the fit is poor.

This poor fit occurs for two reasons. The fit coefficients were defined by the least squares solution of Eqn. 6-3 for all the given information, including the redundant

rows. That is, for the three points and six derivatives. This however, generates a fit that matches the slopes as closely as the function values, and since there are twice as many derivatives as J^* values, the quality of the fit as a predictor of the J^* values decreases. This phenomena in regard to fitting data and derivatives has been previously recognized and a scheme for accommodating gradient information was developed for RS fitting using neural nets^[106]. If however, we use the fact that we have more equations than necessary, we can throw out some of the gradient information. The fit shown in Figure 6.4 is the result of throwing out three derivatives but keeping three others and the three functional values.

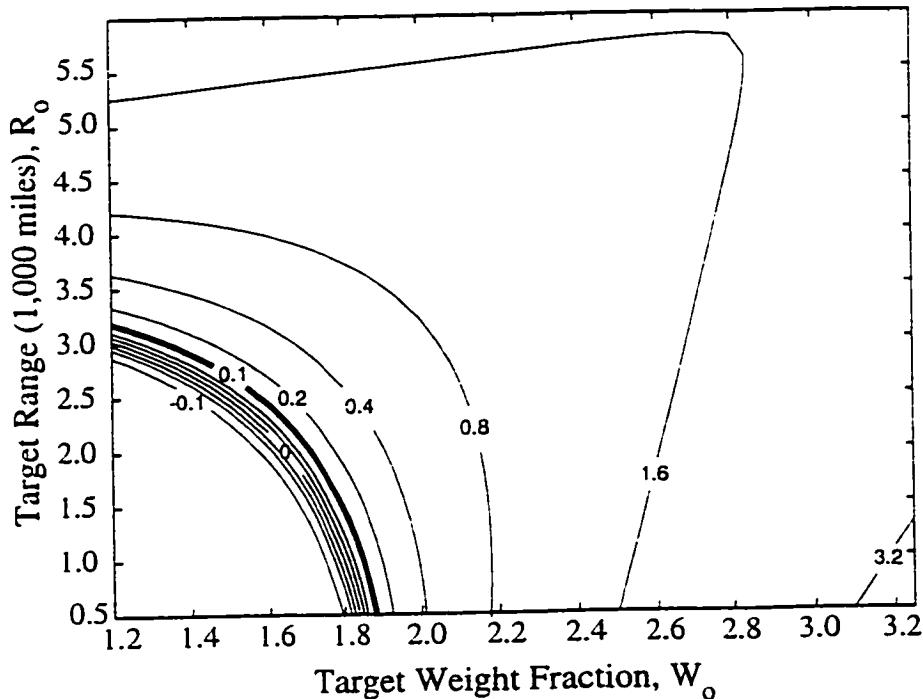


Figure 6.3 RS fit of J^* based on evaluations and gradients at points a, c, and e from Figure 6.1

The RS fit does a better job of approximating the actual surface; with a sum squared error of 0.2803. This fit exhibits the same qualities as the fit based on functional values alone. Figure 6.2, including negative values and a saddle point.

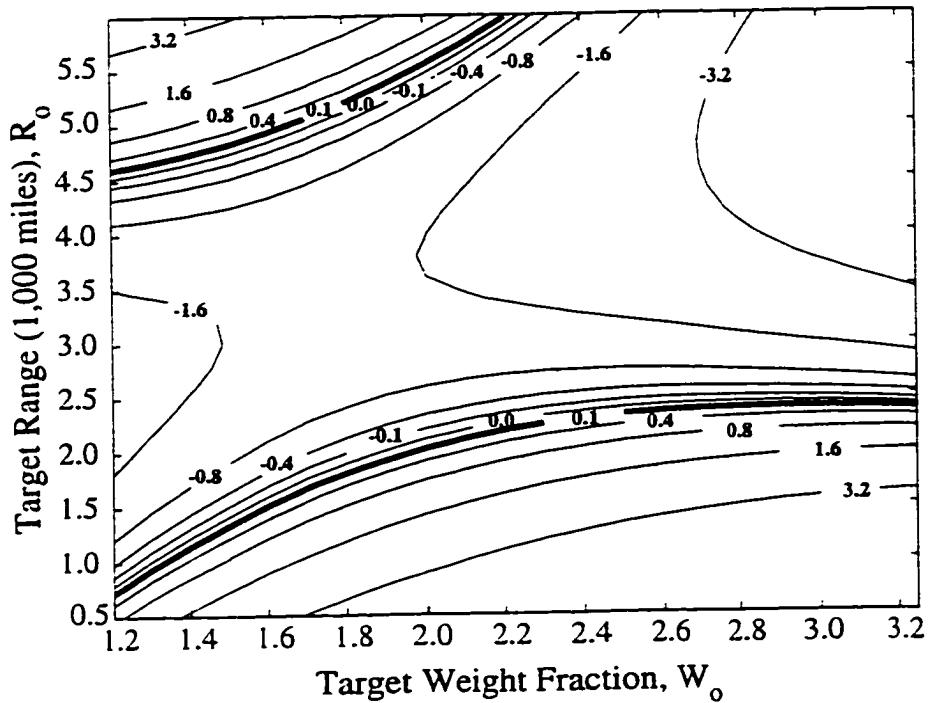


Figure 6.4 RS fit of J^* based on evaluations and gradients, but no redundant information, at points a, c, and e from Figure 6.1

6.4 Modeling Subproblem Optimal Interdisciplinary Design Vector

In the previous sections a response surface model was generated, mapping J^* as an explicit quadratic function of the system level targets vectors $\{z\}$.

$$J^* = f(\{z\}) \quad \text{Eqn. 6-1}$$

An alternate approach is to create a response surface model that predicts the subproblem design parameters $\{x^*\}$ and $\{y^*\}$ as functions of $\{z\}$:

$$\begin{aligned}\{x^*\} &= f(\{z\}) \\ \{y^*\} &= g(\{z\})\end{aligned}\quad \text{Eqn. 6-2}$$

and then substituting $\{x^*\}$ and $\{y^*\}$ into Eqn. 2-2 to determine J^* . Choosing a linear functional relationship for the model, Eqn. 6-2 becomes:

$$\begin{aligned}x_j^* &= B_{j0} + \sum_{i=1}^n B_{ji} z_i \\ y_j^* &= C_{j0} + \sum_{i=1}^n C_{ji} z_i\end{aligned}\quad \text{Eqn. 6-3}$$

For the sample performance problem Eqn. 6-3 is:

$$\begin{aligned}W^* &= c_0 + c_1 W_o + c_2 R_o \\ R^* &= b_0 + b_1 W_o + b_2 R_o \\ J^* &= (W^* - W_o)^2 + (R^* - R_o)^2\end{aligned}\quad \text{Eqn. 6-4}$$

The resulting model of J^* is still of second order but now has slightly different properties from the previous explicit fit of J^* . First, like the actual variation in J^* , Eqn. 6-4 is always positive. Second, no explicit gradient information is required to generate this fit. Third, since there are now only $n+1$ coefficients to solve for, and one independent equation is obtained for each subproblem optimization, the number of subproblem optimizations required to generate the response surface is equal to $m = n + 1$.

The number of points required to generate this fit is the same as that required to generate an explicit quadratic fit of J^* with post-optimal gradient information and a factor of n fewer points than required to fit J^* using function values alone.

The problem of Eqn. 6-1 was solved at only the 3 points a, c, and e shown in Figure 6.1. The resultant topology predicted by the response surface fit is shown below:

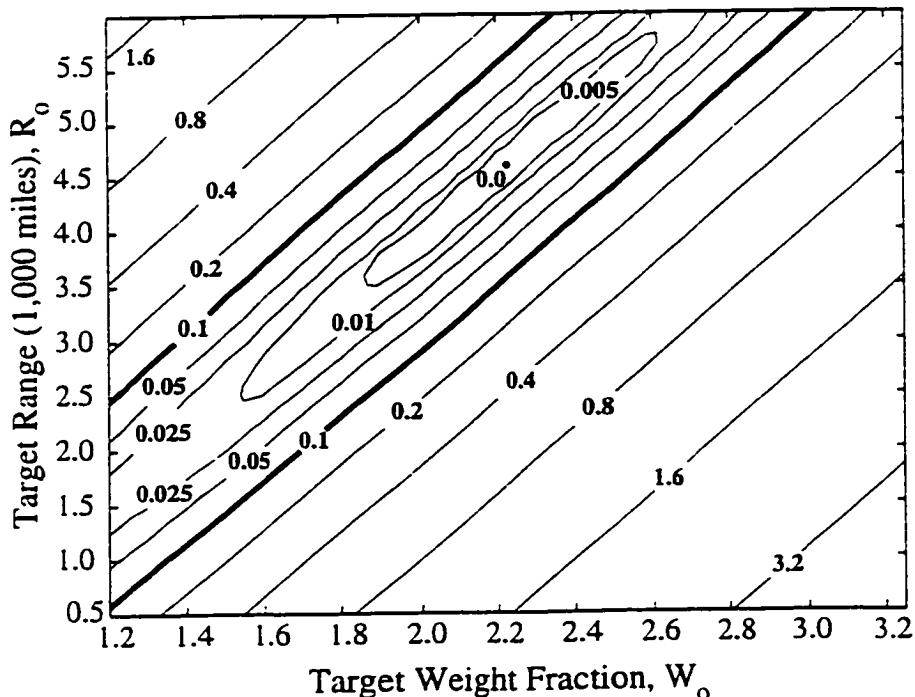


Figure 6.5 RS fit of J^* based fitting the optimal subproblem design vectors $\{x^*\}$ and $\{y^*\}$ evaluated at points a, c and e from Figure 6.1

This fit is a much better representation of the space, with a sum squared error of 0.1417. As before, the contour lines show nonlinear increments, each line representing double the previous value of J^* . Thus, although J^* vanishes at exactly one point ($R_o = 4.5644$, $W_o = 2.2127$), a valley of near zero values (<0.1) closely matches the actual $J^*<0.1$ region illustrated in Figure 6.1.

6.5 Fit Using Implicit Extra-Point Information

Previous sections have shown how the number of required subproblem optimizations to model J^* can be reduced by a factor of n —by using post-optimal gradients to fit J^* , or by fitting the subproblem design vectors $\{x^*\}$ and $\{y^*\}$. Here special

information in each subproblem optimization is exploited to further reduce the number of required subproblem optimizations by 50%.

A given subproblem optimization yields values of $\{x^*\}$, $\{y^*\}$, and J^* , for a given $\{z\}$. The key observation is that if a second subproblem optimization were performed, where the target values $\{z\}$ were set equal to the optimal solution from the first subproblem optimization, $\{x^*\}$ and $\{y^*\}$, the resulting J^* would be equal to zero. This is because the formulation of the subproblem optimization requires that all local constraints be satisfied and that the computed state variables $\{y^*\}$ be compatible with the local design variables $\{x^*\}$. Because the result is known a priori, there is no need to actually perform another subproblem optimization; rather an extra subproblem optimization solution is obtained implicitly, with no additional analysis.

In modeling $\{x^*\}$ and $\{y^*\}$, the implicit point provides one additional equation per subproblem solution, reducing the number of required subproblem optimizations to generate a quadratic response surface to $m = (n + 1)/2$.

When modeling J^* directly, each subproblem optimization yields one additional solution J^* and its full gradient, for a total $n+1$ additional equations. The total number of subproblem optimizations required also becomes $m = (n + 1)/2$.

In the example performance problem, this means that the required number of subproblem optimizations is reduced from three to two. The subproblem was solved at only two points, a and e, shown in Figure 6.1. When these two points were evaluated by Eqn. 6-1 and Eqn. 6-2 the resulting optimal solutions were point g ($W_o = 1.46$, $R_o = 2.19$) and h ($W_o = 2.21$, $R_o = 4.64$), shown also in Figure 6.1. Using a, g, and h, the resulting fit is given by Eqn. 6-1 and is shown in the contour plot of Figure 6.6.

$$\begin{aligned} R^* &= -0.2602 + 0.9101R_o + 0.2959W_o \\ W^* &= 0.6969 + 0.2811R_o + 0.0914W_o \end{aligned} \quad \text{Eqn. 6-1}$$

This fit is visibly very similar to Figure 6.1; it has a sum squared error of 0.1320. Qualitatively it is similar to the fit of Figure 6.5 except that it has no unique minimum. A line runs down the valley where $J^*=0$. In this way it is more like the actual objective function than any of the other fits.

It should be noted that there is no guarantee that multiple executions of a CO subproblem will yield implicit points that are linearly independent. Generally then, the number of function evaluations required to generate a quadratic fit with the implicit point information is problem dependent and between $(n+1)$ and $(n+1)/2$.

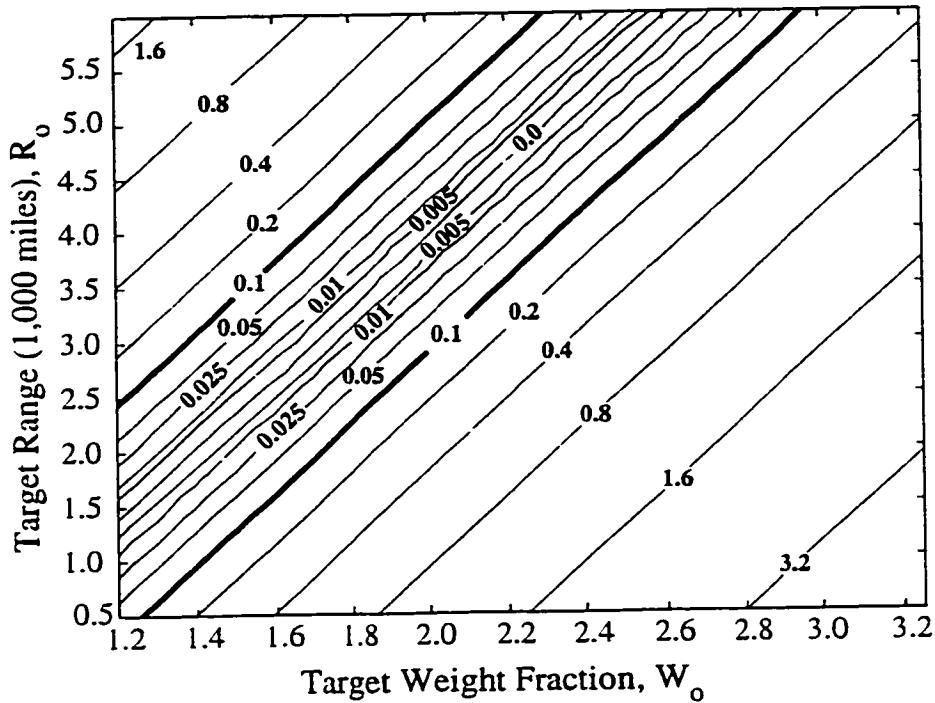


Figure 6.6 RS fit of J^* based fitting the optimal subproblem design vectors $\{x^*\}$ and $\{y^*\}$ evaluated only at points a and e from Figure 6.1

6.6 Summary and Comparison of Fitting Approaches

Two general approaches to fitting response surfaces to CO subproblem optimization results were examined; fitting J^* explicitly and fitting the subproblem optimal design represented by $\{x^*\}$ and $\{y^*\}$. As seen in Table 6.2, for the particular sub-

problem represented by Eqn. 6-1 and Eqn. 6-2, the best fit is obtained using explicit evaluation of J^* at 6 points. However, as seen in Table 6.1, the number of subproblem optimizations required for a design problem of even moderate dimensionality makes this approach impractical. The table also shows that using post-optimal gradients in fitting J^* , or modeling the subproblem optimal design $\{x^*\}$ and $\{y^*\}$, reduces the required number of subproblem optimizations by a factor of n.

Of these two approaches, modeling the optimal design $\{x^*\}$ and $\{y^*\}$ is substantially more accurate than explicitly fitting J^* using gradient information. As mentioned earlier, the resulting fit is also better in a qualitative sense. The fit is always positive—a general characteristic of CO subproblems regardless of the nature of the actual analysis. The more accurate model generated in fewer function evaluations, is made possible by recognizing that the form of the subproblem objective is known. Similar approaches to generating accurate search directions without derivative information, when more information about the form of the objective function is known than simply its value, has been successfully used in shape optimization approaches using genetic algorithms.

Dimension	Functional Evaluations Only	Fit using J and ∇J or $\{x^*\}$ and $\{y^*\}$	Extra Point Info.
1	3	2	1
2	6	3	2
3	10	4	2
4	15	5	3
5	21	6	3
10	66	11	6
20	231	21	11
30	496	31	16

TABLE 6.1 Number of points required for full quadratic fit

The use of the implicit extra point information is demonstrated to reduce the required number of subproblem optimizations by 50% (Table 6.1) while at the same time improving the quality of the fit by 7% (Table 6.2).

Approach	Error
Modeling J^* with functional evaluations at 6 points	0.1005
Modeling J^* using functional values and all post-optimality derivatives	0.8247
Modeling J^* using functional values and derivatives only to fill rank	0.2803
Modeling subproblem results $\{x^*\}$ and $\{y^*\}$ using three subproblem solutions	0.1417
Modeling subproblem results $\{x^*\}$ and $\{y^*\}$ using two subproblem solutions and one implicit solution	0.1320

TABLE 6.2 Comparison of RS fit error on "performance" subproblem

A general quadratic fit based on function values alone would require on order n^2 subproblem optimizations. The parallelism made possible by the response surface still requires $O(n^2)$ independent processors. As a practical matter, even if this number of processors exists, performing this many distinct designs may be prohibitively difficult in terms of implementation, interpretation and communication. The techniques described in this chapter enable the practical use of the parallelism the general use of response surfaces in collaborative optimization makes possible. In the following chapter results from the application of this methodology to design problems will be presented.

Chapter 7

Application to Example Design Problems

In this chapter, the response surface method introduced in Chapter 5 is applied, with the improvements of Chapter 6, to two example design problems. The first problem, designing a subsonic tailless unmanned autonomous vehicle (UAV) for maximum range, is used to demonstrate that the modified version of collaborative optimization converges to the same answer as the basic collaborative implementation, but in substantially fewer consecutive design cycles. The second design problem is to minimize the take-off gross weight of a 250 passenger, Mach 2.4, high speed civil transport (HSCT). The analyses used for the aerodynamic, structural, and propulsion analysis of the HSCT are substantially more sophisticated than those used for the tailless UAV design. This example problem demonstrates the first integration of industry standard analysis codes into the modified collaborative optimization architecture.

7.1 Nomenclature

- α = maneuver angle of attack
- b = span
- C_m = pitching moment coefficient
- C_L = airplane lift coefficient
- C_l = section lift coefficient
- δ_e = elevon deflection angle

η	= propeller efficiency
R	= computed aircraft range
R_t	= target aircraft range
sfc	= specific fuel consumption
t_i	= spar cap thicknesses
θ_i	= section jig twist
w_i	= penalty weight
W_r	= root trim ballast weight
W_t	= tip trim ballast weight

7.2 Tailless UAV

The problem is to design a 1000 lb. gross weight, propeller powered, tailless aircraft for maximum range. The planform, gross weight, and engine performance parameters are fixed, while jig twist, spanwise skin thicknesses, elevon deflection angle, maneuver angle of attack, and the mass of wing tip and wing root ballasts are free design variables. The optimizer uses these free design variables to meet trim constraints during cruise and a 4g maneuver. It also adjusts these variables to satisfy the structural stress and section lift constraints while maximizing the computed range.

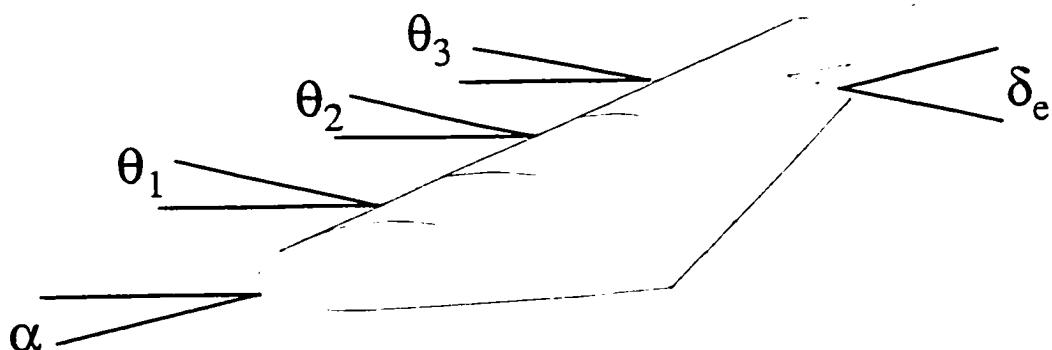


Figure 7.1 Design variables defining tailless UAV.

The design optimization problem is specified by:

maximize: Range
with respect to: $\{\theta\}, \{t\}, \alpha, \delta_e, W_r, W_t$

subject to:

$$\begin{aligned} C_{m_{cruise}} &= 0 && \text{Eqn. 7-1} \\ C_{m_{maneuver}} &= 0 \\ \{C_l\}_{maneuver} &< 1.45 \\ \{\sigma\}_{maneuver} &< 40 \text{ ksi} \\ n_{maneuver} &= 4 \end{aligned}$$

where the specified planform, take-off gross weight, and engine parameters for the design are given in Table 7.1.

Parameter	Fixed Value
Wing Area	200 sq.ft.
Aspect Ratio	15
Airfoil t/c	0.12
Taper, c_t/c_T	0.5
Sweep	15 deg.
Drag Area	1.5 sq. ft.
Takeoff Weight	1000 lbs.
sfc	0.8 lb/hr/hp
$b_{propeller}$	0.8
C_{d_p}	0.009
W_{fixed}	300 lbs.

TABLE 7.1 Fixed parameters in tailless UAV design problem

Figure 7.2 illustrates the structure of the analyses used to design the tailless UAV. The design variables are shown at the top of the figure and calculated output quantities, like stress and range, are shown on the right. Dots indicate that a computed output is used by another analysis as an input.

The same aerodynamic analysis is used to compute a lift distribution, drag, and overall lift and moment coefficients for two different flight conditions; a 4g maneuver

and cruise. The 4g maneuver is performed at the cruise speed, but at a lower altitude and higher dynamic pressure. The optimizer adjusts the elevon deflection angle and maneuver angle of attack to ensure that the generated lift is four times the take-off gross weight, and the aircraft is trimmed in both flight conditions ($C_m = 0$).

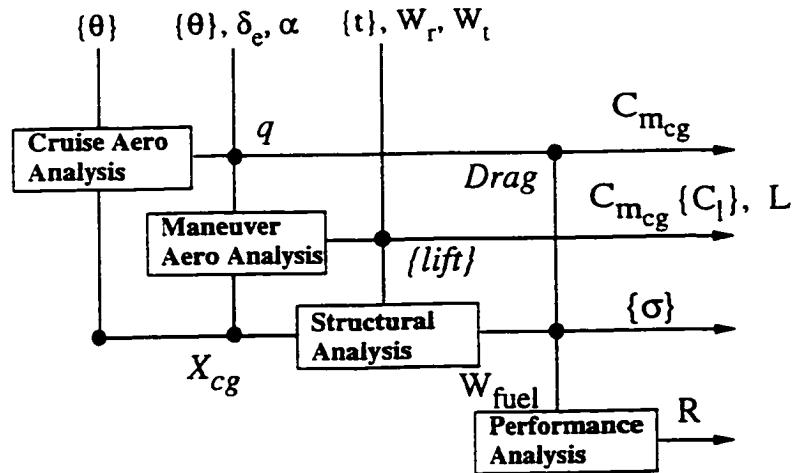


Figure 7.2 Structure of analyses in tailless UAV

Thickness design variable values are an input to the structural analysis which computes a spanwise stress distribution for the given maneuver lift distribution. The position of the aircraft center of gravity is computed accounting for the fixed weights specified in Table 7.1, possible root or tip ballast weights, and the structural weight of the wing spar. As seen in Figure 7.2 this information is used by the aerodynamic analysis for trim. Maximum fuel weight is also calculated in the structures discipline; this value, and the lift and drag computed by aerodynamics, is used by the performance analysis to compute the range.

7.2.1 The Analyses

The aerodynamic analysis uses a definition of wing incidence and the fixed planform information of Table 7.1 to compute the pitching moment coefficient, drag, and lift distribution. The wing is modeled using a vortex panel method with twenty

equally-sized panels. Because the wing planform shape is fixed, the aerodynamic influence coefficients are calculated just once. The linear system is solved for the moment, lift, and drag coefficients. Total drag is the sum of the induced drag, the parasite drag of the fuselage, and the wing parasite drag, proportional to planform area. The computed C_L and the known fixed weight determine the dynamic pressure.

For the maneuver case, the same linear system is solved with a new incidence distribution which is the combination of the maneuver angle of attack and, for the outboard 4 panels, the elevon deflection angle. The output includes the total lift, Cm_{cg} , and section lift distribution.

The wing is modeled structurally with a simple I-beam loaded in bending by the distributed lifting load computed in the aerodynamics analysis. The thickness of the I-beam spar caps is specified at twenty spanwise locations. The width of the beam is 30% of the local mean chord and 75% the local mean wing section thickness. The bending moment and section direct stress is calculated at the inboard end of each panel.

The wing bending weight is proportional to the material volume of the structural spar multiplied by 150% of the density of aluminum. The load independent wing weight is included in the fixed weight parameter specified in Table 7.1. The root ballast is positioned at 25% of the root chord; the fixed weight acts at 60% of the root chord, while the tip weight is located at the tip quarter chord. The total c.g. is computed using the location and size of the tip and root ballast masses, the position and size of the fixed weight and the wing c.g. location. A fuel distribution is assumed along the wing so as not to affect c.g. location. Range is computed using Breguet's equation and the fixed values of engine efficiency, sfc, and take-off gross weight from Table 7.1^[13].

7.2.2 Single-Level Optimization Solution

Since the original problem is a relatively simple example, the analysis can be integrated and the entire problem solved using single-level optimization resulting in an optimal tailless UAV design with a range of 5660 miles. The following table lists other optimal design variable values and computed results:

Parameter/ Variable	Correct Optimal Value
W_r	0.0 lbs.
W_t	25.3 lbs.
α	8.4 deg.
δ_e	12.2 deg.
L/D	21.6
Range	5660 n.mi.

TABLE 7.2 Optimal values of design variables and computed parameters

The lift and twist distributions are shown as the ‘correct’ answer in Figure 7.6 and Figure 7.7.

7.2.3 Collaborative Implementation

The single level analysis was split along the disciplinary boundary between aerodynamics and structures. The variables shown in italics in Figure 7.2 become system level targets and the variables at the top of the figure are strictly local subproblem design variables. The resulting aerodynamics subproblem problem is expressed by:

$$\begin{aligned}
 & \text{minimize: } J_1 \\
 & \text{with respect to: } \{\theta\}, \alpha, \delta_e, X_{cg} \\
 & \text{subject to:} \\
 & C_{m_{cruise}} = 0 && \text{Eqn. 7-2} \\
 & C_{m_{maneuver}} = 0 \\
 & \{C_l\}_{\text{maneuver}} < 1.45 \\
 & n_{\text{maneuver}} = 4
 \end{aligned}$$

The structures subproblem, which also includes the range calculation, is:

$$\begin{aligned}
 & \text{minimize: } J_2 \\
 & \text{with respect to: } \{t\}, W_p, W_t, a_1, a_3, \text{drag} \\
 & \text{subject to: } \{\sigma\}_{\text{maneuver}} < 40 \text{ksi}
 \end{aligned} \tag{Eqn. 7-3}$$

The system level optimization problem is:

$$\begin{aligned}
 & \text{minimize: } -R_o + w_1 J_1 + w_2 J_2 \\
 & \text{with respect to: } a_1, a_3, \text{drag}_o, X_{cg}, R_o \\
 & \text{subject to: } \\
 & \quad J_1 = 0 \\
 & \quad J_2 = 0
 \end{aligned} \tag{Eqn. 7-4}$$

where the system objective is a penalty function of the aircraft range and the two compatibility constraints (where w_1 and w_2 are the penalty weights). The lift is represented by a reduced basis model using two Fourier coefficients (a_1, a_3) as described in Section 4.2.4^[65]. The resulting system level optimization problem is in five variables and two constraints. The correct optimal solution is found in this manner but requires a large number of system level iterations, suggesting that RS methods may be of use.

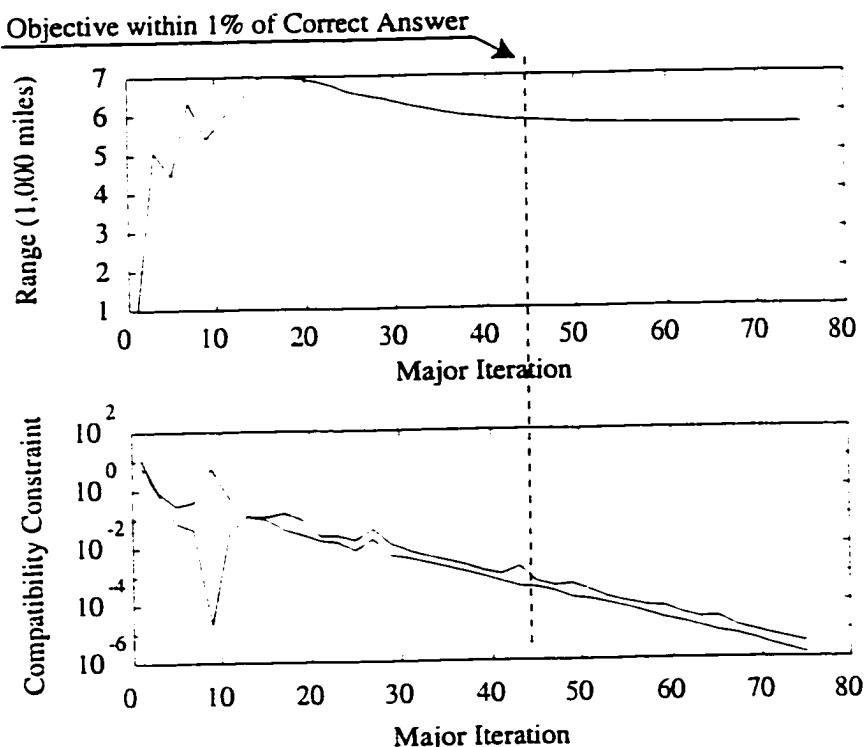


Figure 7.3 Convergence behavior of basic collaborative optimization implementation

Figure 7.6 shows the behavior of the conventional implementation of CO on the tailless UAV problem. Note that a large portion of the major iterations are used to define a compatible design in the region of the optimal range. Each major iteration is equivalent to the parallel development of a response surface model. Comparing this behavior to that shown in Figure 7.4 and Figure 7.5 illustrates the utility of response surface estimation in collaborative optimization.

7.2.4 Collaborative Results with Response Surfaces

Figure 7.4 shows the variation in range versus the RS update cycle for three different starting points. All three closely approach the correct optimal range (determined through the single level optimization in the previous section). Runs starting from points B & C were run with a tighter subproblem convergence tolerance. Each

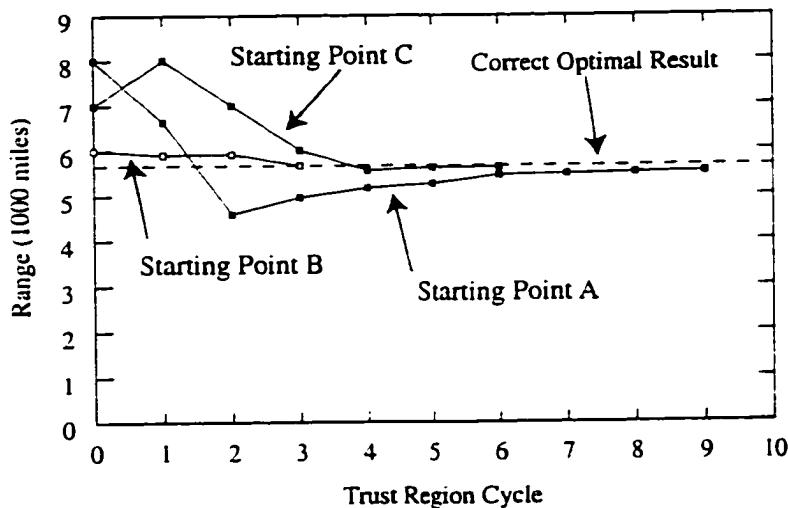


Figure 7.4 Range vs. trust region cycle

of these solutions were obtained in less than ten trust region cycles. The optimization starting from point B obtained the correct solution after only four cycles. Figure 7.5 shows the values of the optimal structures and aerodynamics subproblem objective function values, J^* , for this run. Viewed in combination with Figure 7.4, one sees that the reported range values correspond with compatible subproblems.

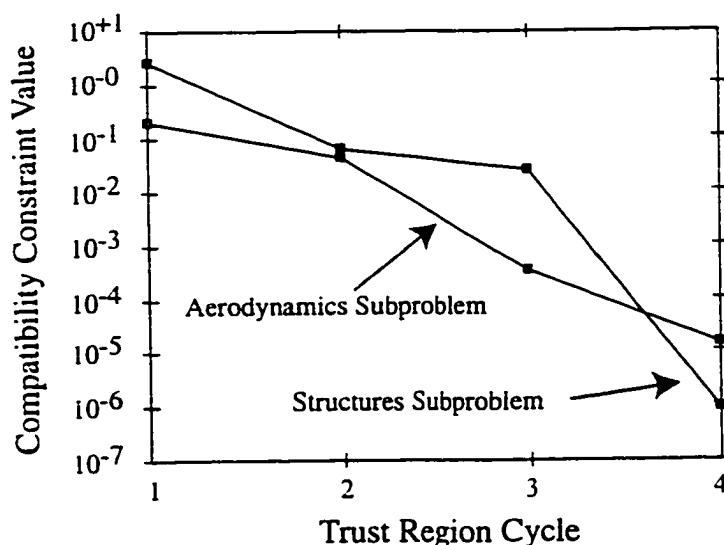


Figure 7.5 Optimal subproblem objective function values vs. trust region cycle

The convergence of the design to the optimal result is also demonstrated by comparing the optimal wing twist and lift distribution for the CO implementation with the single level solution, labeled 'correct' in these figures.

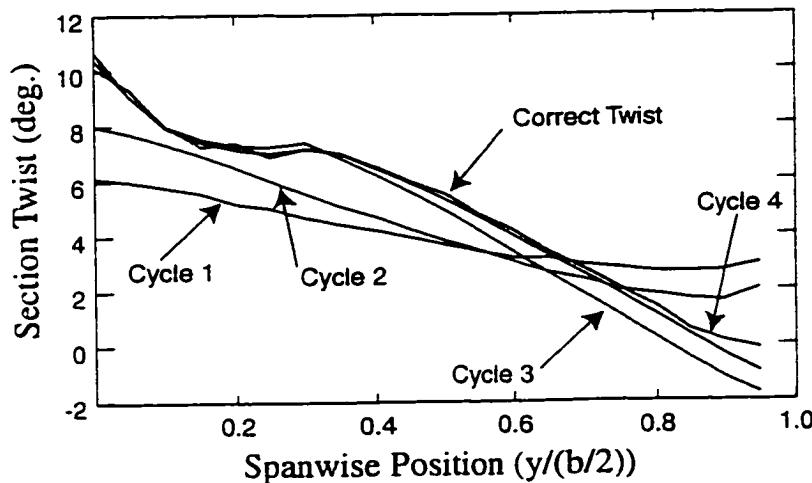


Figure 7.6 Spanwise twist vs. trust region cycle

Figure 7.6 shows the twist distribution determined by the aerodynamics subproblem optimization for the four cycles starting from point B. The final CO design is almost exactly the same twist distribution as the correct solution. Similarly, the lift shown in Figure 7.7 also converges to virtually the same spanwise distribution.

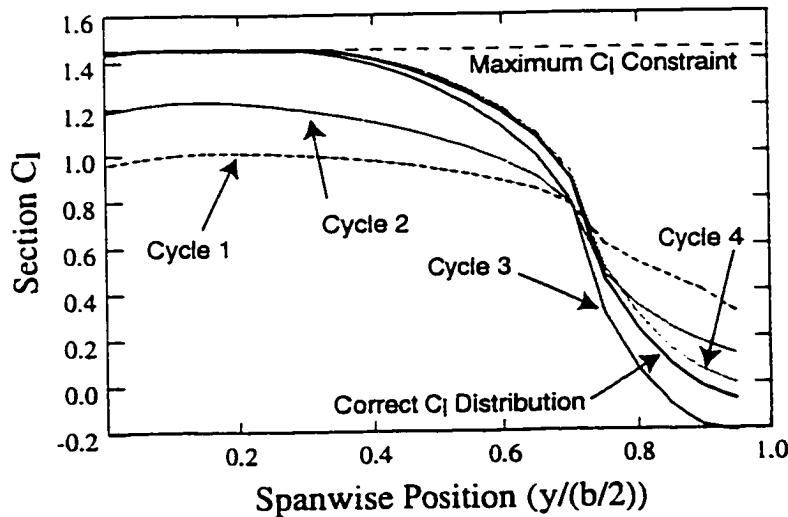


Figure 7.7 Section spanwise C_l

These accurate results were obtained in four trust region cycles. Each trust region cycle requires 6 subproblem optimizations, but these may be evaluated in parallel. Thus, the total number of serial subproblem optimizations required to generate this solution is between 4 (with coarse-grained parallelization) and 24 (for execution on a single processor).

7.3 HSCT Design

This section describes the multidisciplinary design of a high speed civil transport (HSCT) using collaborative optimization with response surface estimation and refinement. In this example, industry standard analysis tools were incorporated in the collaborative architecture for the first time.

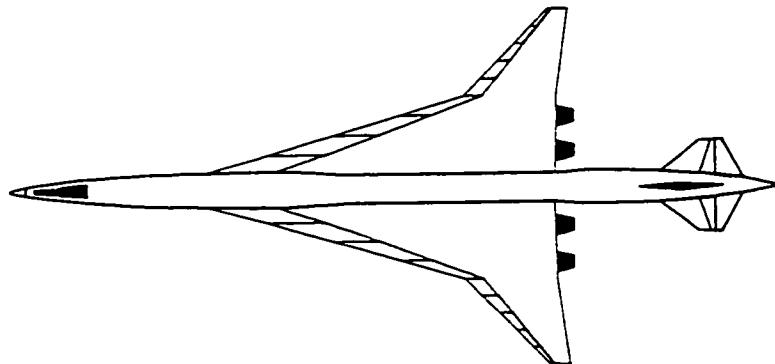


Figure 7.8 HSCT top view

The goal was to design a 250 passenger, Mach 2.4, 5,000 mile range, HSCT for minimum take-off gross weight, accounting for lift and stress constraints at cruise and a 2.5 g maneuver flight condition. The interdisciplinary design variables specified by the system level optimization were takeoff weight, zero fuel weight, sea level static thrust, initial cruise altitude, lift to drag ratio, the spanwise maneuver induced twist distribution, and the spanwise moment and torque distribution. The

planform variables were fixed at their values shown in Table 7.3. This design work was performed within the Aircraft Design Group at Stanford, using industry standard computational tools for the aerodynamic, structural, and performance disciplines.

Parameter	Value	Parameter	Value
TOGW	745000 lb	Mach	2.4
ZFW	400000 lb	h_{initc}	57000 ft.
SLST	60000 lb	Range	5000 m.
AR	2.43	Λ_{in}	75 deg.
S	8276 ft. ²	Λ_{out}	47 deg.
b	134 ft.	λ_{in}	0.458
C_{root}	149 ft.	λ_{out}	0.169

TABLE 7.3 HSCT baseline design

7.3.1 HSCT Aerodynamic Design Analysis

The HSCT aerodynamic disciplinary design requires computation of L/D, lift, and spanwise torque and moment distributions at two flight conditions. The wing planform is fixed as described in Table 7.3 but, as shown in Figure 7.10, the discipline is free to define the spanwise twist distribution, angles of attack, and initial cruise altitude.

The collaborative aerodynamics subproblem design attempts to find a set of local design variables that reduce the discrepancy between computed results, local design variables, and system level targets, subject to the local constraint that the maneuver lift is 2.5 times the cruise lift (the maneuver condition).

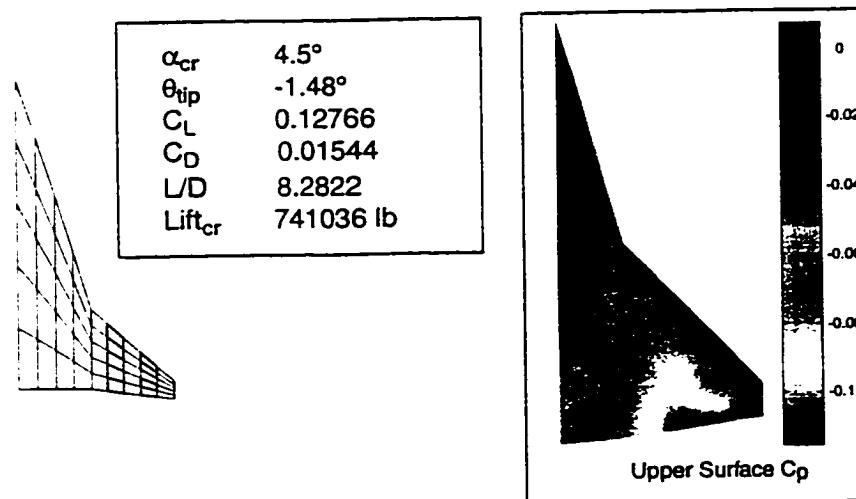


Figure 7.9 A502 wing panel model and computed upper surface pressure distribution

The aerodynamics analysis uses a higher order surface panel code, A502^{[118][119]}, to define the aerodynamic loading as a function of the fixed planform and the design variables (spanwise twist, cruise altitude, takeoff weight, and angle of attack). A502 was developed by Boeing to solve the Prandtl-Glauert equations for inviscid, irrotational flow and may accommodate non-linear singularity distributions along the panel mean surface. The HSCT wing is defined by five chordwise and nine spanwise panels with the twist defined at each spanwise panel location. The grid was intentionally kept relatively coarse to minimize the required computational time for a single analysis. A C_p is computed at the center of each panel for the supersonic cases considered. Figure 7.9 shows the panel geometry used to model the HSCT and a sample pressure distribution for the baseline aircraft.

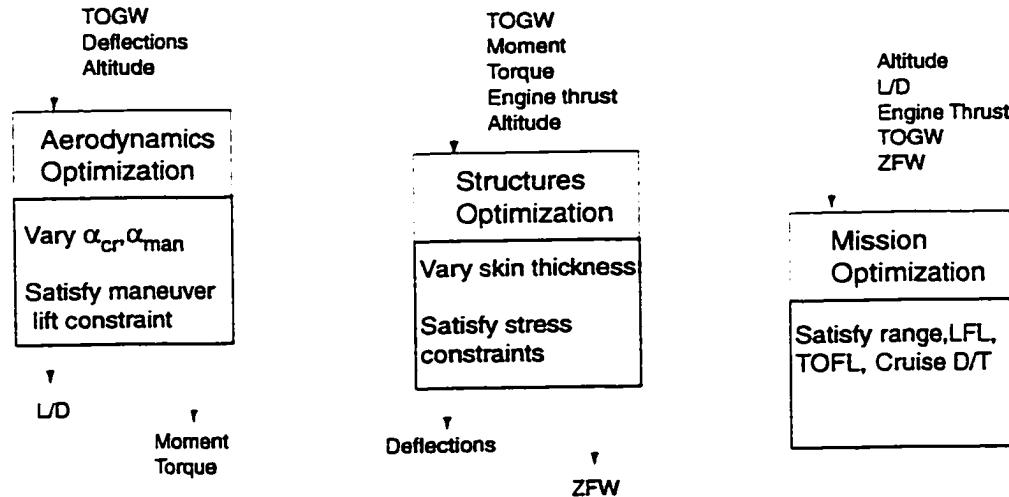


Figure 7.10 Computed results and design variables for each of the disciplinary design problems

7.3.2 HSCT Structural Design Analysis

The structural analysis used to model the HSCT wing is an industry standard FEM code^{[116][117]}. As depicted in Figure 7.11, the wing is represented with 324 triangular elements broken into 180 skin and 144 web panels. The root deflection is fixed as a boundary condition, and the deflections and stresses in the wing resulting from the loading are calculated at over one hundred node points. The resultant twist distribution is then computed. The finite element code, FESMDO, was developed at NASA-Langley and has been compared with NASTRAN results for several HSCT wing models with very similar results^[120].

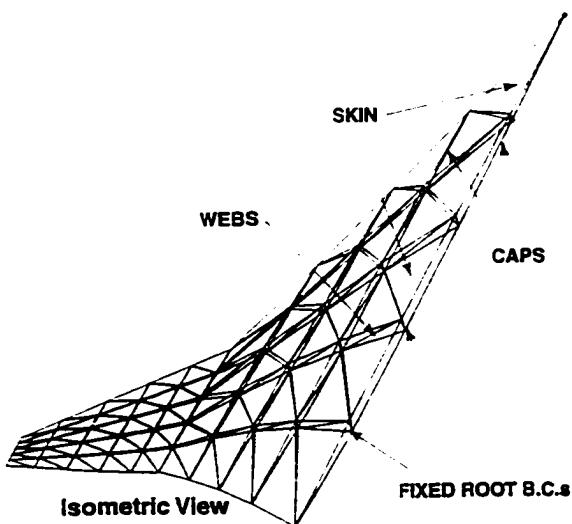


Figure 7.11 FESMDO FEM structural representation of HSCT wing

Figure 7.10 shows the local design variables manipulated by the structures subproblem to obtain the computed results. As a collaborative design subproblem the discipline tries to find the design variable vector that minimizes the difference between the computed results, the input vector, and the target design variable values, while also satisfying the allowable stress constraints. The computed zero fuel weight is based on the structural weight of the wing bending material, propulsive weight corresponding to the engine thrust requirements, and fuselage weight based on statistical data.

The mission analysis is based on algorithms from PASS^[32]. It computes takeoff and landing field length, drag-to-thrust ratio at different flight stages, and range as a function of initial cruise altitude, sea-level static thrust, takeoff gross weight, zero fuel weight, and cruise lift to drag ratio. The range is constrained to be at least 5000 miles, the takeoff field length to 12,000 ft. and landing field length to 7,000 ft. Engine performance requirements are defined by constraining the available thrust to exceed the computed drag. Elements of PASS are used in structures to determine

non-structural weights and in aerodynamics to compute skin friction and volume wave drag of components not included in the A502 model.

7.3.3 Collaborative Implementation with Response Surfaces

Figure 7.12 illustrates how response surface estimation was used in the HSCT collaborative design process. Response surfaces were used to model the subproblem optimization results, just as was done for the UAV design of the previous example except for the aerodynamics subproblem, where response surfaces were used in lieu of the analysis. This use of response surfaces was motivated by two observations. First, the aerodynamics design problem was defined in relatively few strictly local variables versus interdisciplinary variables. As discussed in Section 5.2.1, response surface representation of optimal designs, instead of analyses, is most appropriate when the number of strictly local variables is much greater than the number of interdisciplinary variables. The second motivation is that experience with A502 has shown it to cause optimizers difficulty due to noisy gradients. Similar observations were made in [76] and response surfaces were shown to be a useful remedy.

To reduce the size of the system level design variable vector, high bandwidth coupling variables such as the wing pressure distribution were represented using a reduced basis approach. This involved representing the pressure distribution as computed moments and torques and, in turn, representing these quantities with a reduced set of parameters.

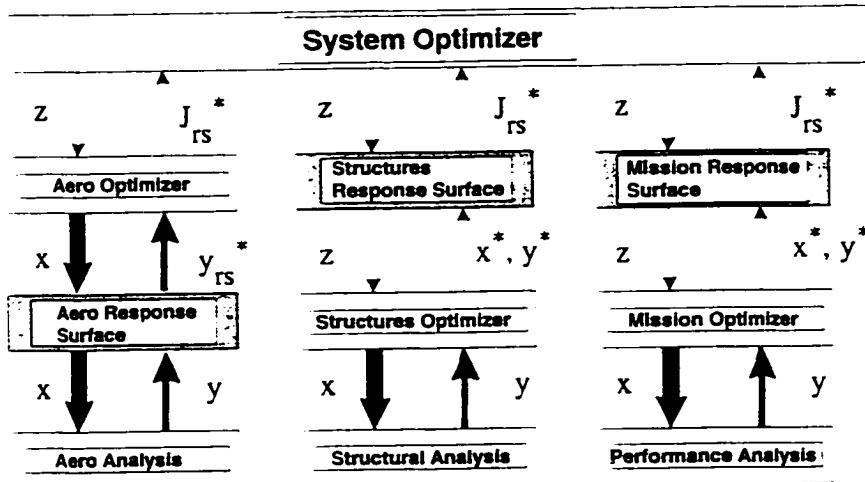


Figure 7.12 Use of response surfaces in HSCT CO design problem

The same response surface generation method and update scheme used on the UAV design problem was used on the HSCT, with one modification. As before, the system level optimizer iterated with the response surface models of the subproblem objective function in lieu of the actual subproblem design. In the algorithm described in Chapter 5 only a single optimization is performed within the current trust region boundary (Δ). The modification to this procedure is to solve the response surface based optimization with several different values of the trust region boundary (e.g. Δ , 2Δ , 3Δ , etc.). Doing this results in as many optimal solutions $\{z^*\}$, based on the response surface, as different trust region values are specified. The resulting $\{z^*\}$ vectors are then evaluated using the actual subproblem optimization and the best actual value of the overall merit function is taken as the new design point around which to generate a new response surface. The motivation for this procedure is to see if the current response surface may be acceptable over a broader domain than expected, and to take advantage of computational resources that were used in the parallel generation of the response surface model, but were previously unused in the evaluation of the candidate design point.

This procedure is similar to that proposed in [123] and can be implemented in such a way as to require no additional real-time computational cost. It has already been described how parallel processing may be used to generate a response surface in the real-time required for a single subproblem optimization. Similarly, many potential points (one for each different value of the trust region boundary) may be evaluated in parallel. In this way, in the real-time used to evaluate a single new candidate point, many potential candidate points may be evaluated. This is another useful application of the coarse-grained parallelism facilitated by this method.

7.3.4 HSCT Design Results

The initial baseline aircraft described in Table 7.3 is not a compatible design. For example, the baseline TOGW was 745,000 lbs while the best local design weights were 739,000 lbs from aerodynamics, 744,000 lbs from structures, and 720,000 lbs from performance. Figure 7.13 summarizes the variation in the subproblem compatibility constraints with update cycle.

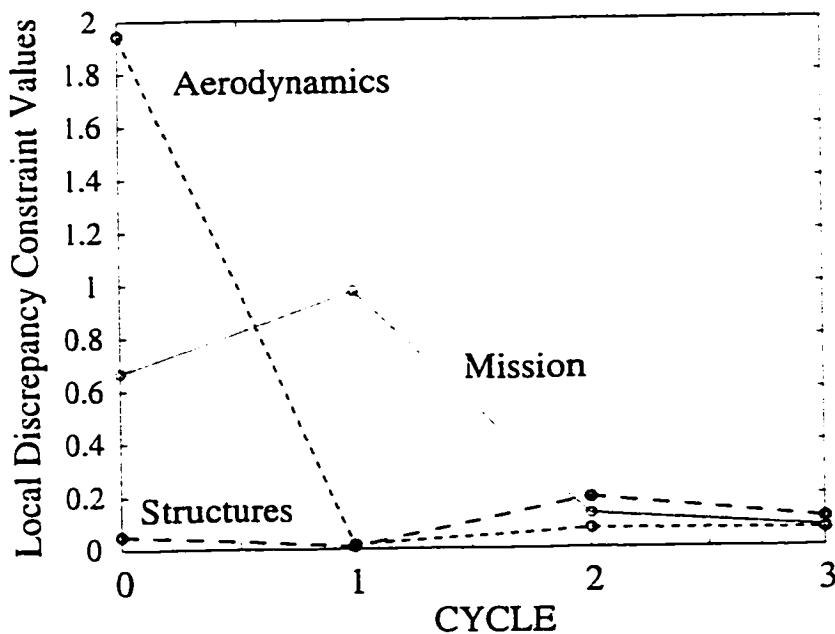


Figure 7.13 Compatibility constraint value with design cycle

The initial target design aircraft was too light, flew too high, and lacked sufficient thrust to satisfy the local design constraints. The cycle 3 aircraft was found by increasing sea-level static thrust, take-off weight, and decreasing initial cruising altitude. The convergence of the three disciplinary designs to a more compatible design is shown in Figure 7.14 and Figure 7.15 for two different design parameters.

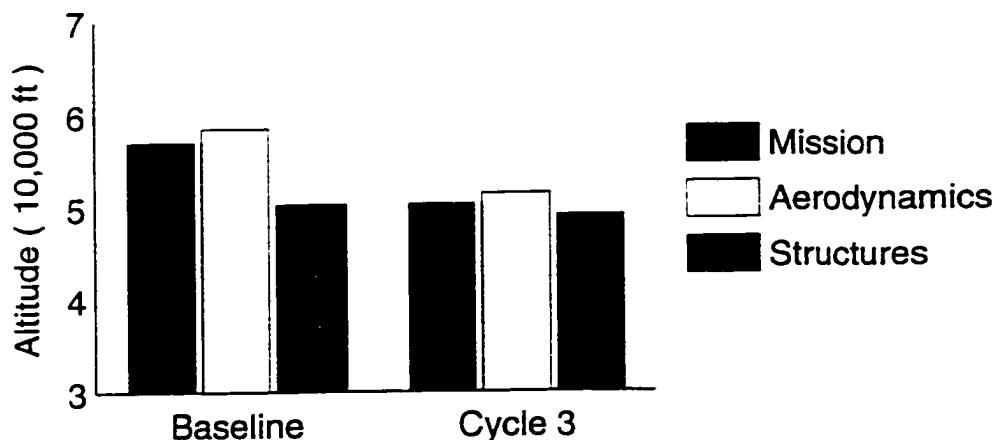


Figure 7.14 Initial cruise altitude from the three subproblems

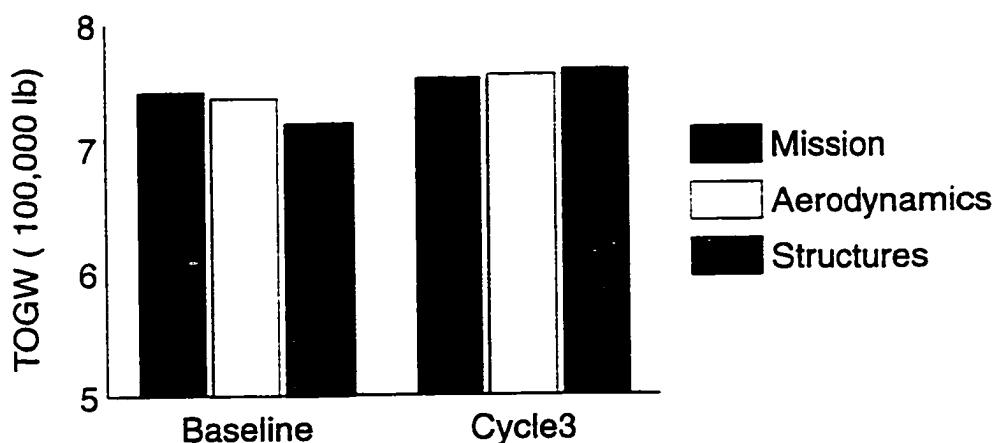


Figure 7.15 Takeoff gross weight (TOGW) from the three subproblems

The previous figures show changes in the target variable values and corresponding local values. However, locally feasible designs are generated in the subproblems through adjusting strictly local variables as well as local copies of global variables. Figure 7.16 shows the initial pressure distribution and the cycle 3 distribution. The cycle 3 pressure distribution results from the modified twist chosen by the aerodynamics subproblem to meet its local constraints while minimizing its discrepancy function.

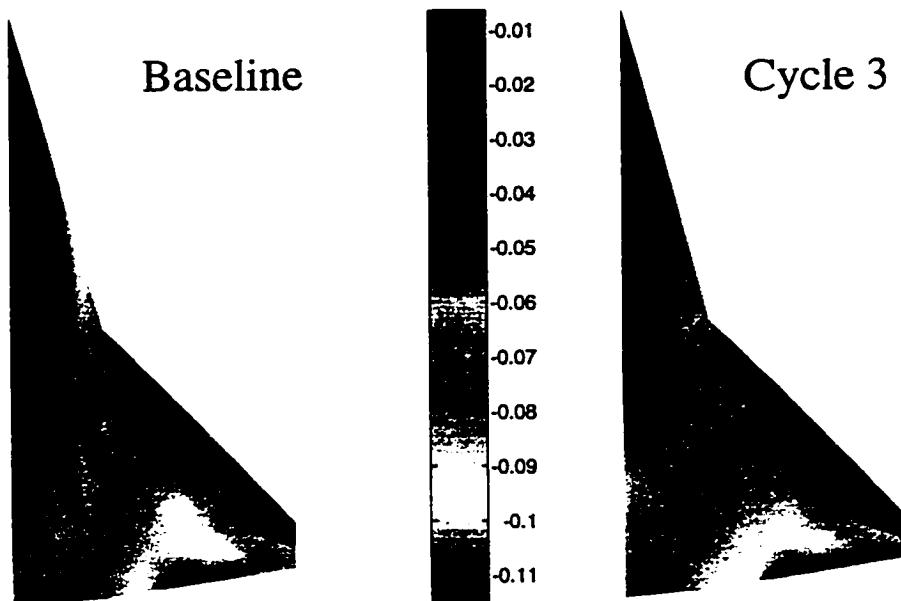


Figure 7.16 Initial and final upper surface pressure distributions

As before, each update cycle represents the generation of a new response surface. New candidate design points were found, on the basis of the response surfaces, that progressively reduced the discrepancy between local values of interdisciplinary variables. A design that was compatible across the three disciplines to within 1% was found in 4 design cycles. This solution is not considered a converged result since the trust region had not shrunk to a sufficiently small size and discrepancies between design variables still exist.

The design point represented by cycle 3 is a considerable improvement over the baseline design. The baseline aircraft had a landing field length of 7303 ft., a cruise drag to thrust ratio of 1.48, and a range of 4,786 miles. All of these values violated performance constraints. The cycle 3 design satisfied the constraints placed on these values of a maximum 7,000 ft. landing field length, drag to thrust ratio of less than one, and range of at least 5,000 miles.

The cycle 3 aircraft is a locally feasible design but not compatible to the same tolerances applied to the UAV or ship design problems. The results of the three design cycles shown here represent the successful integration of industry standard codes within the collaborative optimization architecture. The actual design of a HSCT using this methodology is an ongoing dissertation research project at Stanford.

7.4 Summary

The best test of any multidisciplinary optimization method is application in a real world design setting. However, in the research environment MDO methods, like those discussed in Chapter 2, tend to be applied to relatively small (in terms of the number of design variables) problems using simple analyses. Though the HSCT problem defined here is still relatively small compared to an industrial preliminary design of this aircraft, this section represents the first integration of industry standard codes in a collaborative optimization architecture using response surface estimation and refinement.

The results from the UAV design demonstrate that the collaborative optimization method, with response surface estimates, does converge to the same answer as the basic and single-level approaches. The performance advantages of the modified version are evident from the reduced number of design cycles required to find the correct optimal solution. In addition, the solution to the UAV design was found using the improved response surface construction techniques of Chapter 6, demonstrating their successful modeling of the subproblem optimal design space.

Chapter 8

Conclusions and Development Directions

This thesis describes the first incorporation of response surfaces into the collaborative optimization framework. The modified method has a variety of advantages demonstrated in its application to several example design problems. Compared with the basic implementation, the modified formulation is easier to implement and exhibits improved performance. Improvements in response surface generation techniques may have broader applications whenever additional information concerning the form of the objective function being modeled is known. The particular implementation of response surfaces in this thesis represents their first use to model optimal design results rather than analysis. These approximate models are a promising addition to the basic collaborative optimization architecture.

8.1 Summary and Conclusions

8.1.1 Review of Basic Formulation

The basic formulation of collaborative optimization was applied to four multidisciplinary problems. The theoretical applicability of collaborative optimization was addressed in general terms noting that the method is best suited for real world design problems where integration of analyses is prohibitively expensive or unmanageable. The sensitivity of the basic method to the size of the system level design variable vector was highlighted and techniques for representing special cases of high-

bandwidth coupling with a reduced variable set were proposed. The beneficial effect of warm starting in consecutive subproblem optimizations, the need to tightly converge subproblem optimizations for accurate post-optimal gradients, the benefit and successful use of post-optimal sensitivities, and the convergence behavior of the method were demonstrated on sample design problems. These characteristics motivated the use of approximate methods within the basic collaborative framework.

8.1.2 Introduction of Response Surfaces

Response surfaces are incorporated for the first time in collaborative optimization in this work, and are applied in a novel way. The response surfaces are used to model the subproblem optimal design rather than the analysis. Response surfaces improve collaborative optimization by making the method easier to implement and by improving its performance. Performance is primarily enhanced by the use of coarse-grained parallelism in the generation of the response surfaces. Given the availability of sufficient computational resources, response surface models of the optimal design space may be generated in the time required for a single subproblem optimization.

Because the points used to define the response surface are generated concurrently and are far apart, warm starting is less crucial to efficient performance. Additionally, since the response surface predicts the optimal subproblem objective (J^*) over a relatively broad domain, small errors in the value of J^* are less critical to overall response surface quality. Thus, the method requires only loose convergence of the subproblem optimizations, resulting in computational savings. In general, the use of response surfaces in collaborative optimization results in a method that is less sensitive to implementation details than the basic approach.

8.1.3 Response Surface Improvements

As a practical matter, the subproblem design may not be an automated process and computational resources may restrict the number of parallel designs that can be performed. Two techniques were investigated that reduce the number of designs

required to generate a quadratic response surface from $O(n^2)$ to $O(n)$. The first of these, which uses post-optimal derivatives, has several disadvantages: it requires careful scaling of the gradient and function values, tight convergence of subproblem optimization, and it creates response surfaces that have values unattainable by the actual function being modeled (it can predict J^* values less than zero, whereas the actual discrepancy function can only have non-negative values). The second method, which separately models individual terms of the objective function with a lower order fit, generates response surfaces in $O(n)$ subproblem optimizations that are closer to the actual function being modeled.

A fundamental observation concerning the form of the subproblem objective allows a further reduction in the computational cost of forming the response surface in some cases. Each subproblem optimization generates two values: the discrepancy function J^* and a design point at which we implicitly know that $J^*=0$. If all these implicit points are linearly independent then the number of subproblem optimizations required to form a quadratic response surface is reduced to $O(n/2)$.

In general, the response surface fits are valid over only a limited range of the design space. The solution found from a single set of response surfaces is reevaluated using the actual subproblem optimization and a new set of response surfaces is created. The domain over which these response surfaces are formed is adjusted according to rules from trust region methodology. The size of the trust region is adjusted asymmetrically depending on whether the solution of the previous fit was on the trust region boundary and on how much the merit function actually improved.

8.1.4 Applications

The resulting algorithm was applied to three design problems. The tanker design problem was solved using three methods: single level optimization as a benchmark, collaborative optimization, and collaborative optimization with response surface estimation methods. All three approaches found the same answer and the response surface approach demonstrated substantial real-time savings versus the basic implementation. The tailless UAV problem was implemented using the improved response

surface generation techniques. The problem was also solved by the same three methods. Again, all three methods found the same optimal answer and again collaborative optimization with response surface estimates showed substantial performance improvements over the basic implementation.

More sophisticated analysis codes were used in the structural and aerodynamic analysis of the HSCT. This problem was implemented using only the modified approach as a demonstration of the capability of the architecture to easily integrate industry standard analysis codes. A slightly modified update algorithm was utilized that allowed for extrapolation beyond the trust region boundary when actual design analysis showed improvement outside of that boundary. Several design cycles of the HSCT were performed and resulted in a substantially improved design.

8.2 Design in the Collaborative Environment

The use of response surfaces facilitates the practical use of the collaborative optimization architecture. In Chapter 1, the design process was described for the Boeing 777. The twenty people who gathered weekly in Renton, Washington, performed an analogous role to the system level optimizer in collaborative optimization. Through their interaction, interdisciplinary disputes were resolved and attempts were made to improve the performance of the overall design. As discussed in Chapter 1, the methodology behind these trade-offs is *ad hoc* and motivates a procedure that formalizes the process. Collaborative optimization provides a methodology for performing design optimization at this level in the multidisciplinary design process. The current procedure would be modified only in small, but important, ways. Each disciplinary design team would be tasked with the same kind of design objective, the development of a local design as similar as possible to the target design (e.g. in the case of the 777, that target design was specified in the weekly meetings). At the next design cycle each disciplinary group could report on their success or difficulty in matching the target design.

The basic formulation of collaborative optimization provides a method of analytically specifying how to change the target design to reduce incompatibility between subproblem local designs. This information is useful in making choices on how to change the target design.

The introduction of response surfaces qualitatively changes this process in the following way: rather than having information about a single design point (e.g. the current system level target), each disciplinary subproblem design generates a model of their optimal design space as a function of a relatively broad range of target values. In the meetings of the controlling authority the disciplinary design representatives can compare notes on the effect of macro changes to the target design. In short, the basic approach provides the controlling authority with a single point and a gradient pointing toward the direction of improvement. The response surface approach provides a second order model of the optimal design space over a specified domain.

This second approach provides richer information to the human experts that comprise the controlling authority. The basic approach is well tuned for a quantitative optimizer, not collective expertise. The basic method provides an assessment of quality of the design and a direction for the different disciplines to follow. The use of response surfaces, as models of the disciplines' optimal design space, allow the designers to add their experience and understanding to the disciplinary predictions in deciding how to modify the design and in understanding how the disciplines might be affected by the modifications.

8.3 Directions for Future Work

The incorporation of response surface approximations in the basic formulation of collaborative optimization is based upon quadratic models of the subproblem optimal design space and an algorithm for the refinement of those models. Improvements in either of these components of the method, the trust region update algorithm or the response surface construction, will enhance its performance.

8.3.1 Trust Region Updates

The update algorithm used in this study was found adequate for the test problems considered. However, trust region algorithm development is an active research area. This research has recently focussed primarily on the generation of robust update procedures for which convergence proofs may be developed. In collaborative optimization a primary goal is the minimization of the number of design cycles required to obtain interdisciplinary compatibility and a converged design. Unlike generic problems, in collaborative optimization the form of the subproblem objective function is known; this information may be useful in the design of an update algorithm tailored to collaborative optimization. Similar recognition of the special form of the subproblem objective allowed for the reduction in the number of required subproblem optimizations to generate a quadratic response surface.

8.3.2 Response Surface Construction

The form of the response surface model is also an area of potential improvement. The typical form of a subproblem optimal design space is characterized by areas of compatibility where $J^*=0$ and areas of incompatibility where J^* is nonzero. The quadratic model is ill-suited to model this flat area while still adequately modeling the nonlinear variation in the subproblem objective for incompatible target design variable values. A model well suited for this kind of design space, such as a neural net or n^{th} dimensional spline, might substantially increase the quality of the response surface model, increase its domain of applicability, and thus decrease the number of response surface updates and subproblem optimizations required for convergence. Neural nets have a variety of drawbacks as a modeling technique, including the need to design the net and requirements for substantial amounts of training data. However, they do have the feature of potentially retaining previous information in their approximation. This quality, of preserving previous subproblem design solutions in the response surface fit, is not currently included in the algorithm and may also improve performance.

The observation concerning the extra point information might be exploited to further enhance the modeling approach in collaborative optimization. The current measure of response surface quality has been directed toward the accuracy with which $J_{rs}^*(z)$ models the actual variation in $J^*(z)$. However, what the designer is most concerned with is the area of the design space where $J^*=0$. Only to a lesser extent is the designer interested in areas where a disciplinary designer is unable to meet a target. This observation suggests that the implicit extra-point information should be used to develop a response surface model of the design space for which $J^*=0$.

Finally, the best test of any design optimization architecture is industrial implementation. In fact, the primary features that make collaborative optimization attractive, such as maintaining disciplinary design autonomy and minimizing interdisciplinary communication, will only be evident in application to a real-world industry design project. Continued examination of the method's performance on academic problems is valuable, but an industrial application of collaborative optimization is clearly the next step.

References

1. Lindbergh, Charles. A. *The Spirit of St. Louis*, Scribners, 1927.
2. Sabbagh, Carl, *21st Century Jet*, Scribners, 1996. pg.84
3. *Vega Shop Manual - 1974* and *Vega Shop Manual Supplement - 1974*, General Motors Corporation, available from Llyod's Literature, Newbury, Ohio, 1974.
4. Serling, Robert, J., *The Electra Story: Aviation's Greatest Mystery*, Doubleday, 1963.
5. Mistree, F., Marinopoulos S., Jackson, D., Shupe, J., "The Design of Aircraft Using design Support Problem Technique", NASA Contractor Report 4134, April, 1988.
6. "Current State of the Art on Multidisciplinary Design Optimization", *AIAA White Paper*, AIAA Technical Committee on Multidisciplinary Design Optimization, September, 1991.
7. McCulley, C., Bloebaum, C. L., "Complex System Design Task Sequencing for Cost and Time Considerations", Sixth AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, WA, Sept. 1996.
8. Johnson, E., Brockman, J., "Reducing Overall Design Time through Efficient Allocation of Individual Task Times", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellevue, WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4159.

9. Altus, S. S., *Multidisciplinary Aircraft Design Optimization Using Single-Level Decomposition*, Ph.D. Thesis, Stanford. Univiersity, 1997.
10. Eason, E., Merton, A.. "Non-Hierarchic Multidisciplinary Design of an Automobile Suspension", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellevue. WA. Sept. 4-6, 1996. Also available as AIAA Paper 96-4082.
11. Devries, R.. Ford Motor Compnay, Personal Communication. July 28, 1998.
12. Sharifzadeh, S., Koehler, J., Owen, A., Shott, J., "Using Simulators to Model Transmitted Variability in IC Manufacturing". IEEE Transactions of Semiconductor Manufacturing, 2(3), Aug. 1989.
13. Lokanathan, A.N., Brockman, J.B.. and Renaud, M. "Concurrent Design of Manufacturable Intgegrated Circuits", AIAA-96-4094.
14. Lokanathan, A., Brockman, J., Renaud, M., "Concurrent Design of Manufacturable Integrated Circuits", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I. Bellevue. WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4094.
15. Kodiyalam, S., Sweson, L., Stehlin, B., "Multidisciplinary Design Optimization with Object-Oriented Product Modeling", Engineous Software, Inc., Raleigh, North Carolina, 1997.
16. Mistree, F., Smith, W.F., Bras, B., Allen, J.K., Muster, D., "Decision Based design: A Contemporary Paradigm for Ship Design". Transactions, *Society of Naval Architects and Marine Engineers*, vol. 98, pp.565-597, 1990.
17. "Embraer defining Structural Changes to Correct EMB-145 Aerodynamic Problems", Av. Week Nov.26 1990. pg.33
18. Gage, P.J., *New Approaches to Optimization in Aerospace Conceptual Design*, Ph.D. Thesis, Stanford University, 1995.
19. Gage,P.J., Kroo, I.M., Sobieski, I.P., " Variable Complexity Genetic Algorithm for Topological Design ", *AIAA Journal*, vol.33, no. 11, pp.2212-17, Nov. 1995.
20. Foster N., Dulinkravich,G., "Three Dimensional Aerodynamic Shape Optimization Using Genetic Search Algorithms:, *Journal of Spacecraft and Rockets*, Vol.34, No.11, Jan.1997.

21. Mason, W., and Arledge, T., "ACSYNT Aerodynamic Estimation - an Examination and Validation for Use in Conceptual Design", AIAA Paper 93-0973, 2nd Annual AIAA/USRA/AHS/ASEE/ISPA Aerospace Design Conference, Irvine, CA, February 16-19, 1993.
22. Wang, F., Wright, P., "A Multidisciplinary Concurrent Design Environment for Consumer Electronic Product Design", *Concurrent Engineering*, Volume 4, No. 4, December 1996.
23. Gupta, S., Nau, "Systematic Approach to Analysing the Manufacturability of Machined Parts", *Computer Aided Design*, Vol. 27, No.5, pg. 323-342, May, 1995.
24. Cutkosky, M., Tenenbaum, J., "Providing Computational Support for Concurrent Engineering", *International Journal of Systems Automation*, Vol 1, pg. 239-261, 1991.
25. Fruchter, Renate, "Conceptual, Collaborative Building Design Through Shared Graphics", *AI in Civil and Structural Engineering*, vol. 11, no. 3, pg. 33-41. 1996.
26. Altus, S.S, Kroo, I.M., Gage, P.J.. "A Genetic Algorithm for Scheduling and Decomposition of Multidisciplinary Design Problems", ASME Paper 95-141. Boston. MA, Sept. 1995.
27. Rogers, J.; McCulley, C.; and Bloebaum, C., "Integrating a Genetic Algorithm into a Knowledge-Based System for Ordering Complex Design Processes", Proceedings of the '96 Artificial Intelligence on Design Conference, pp. 119-133, June 1996, Stanford University, CA, also NASA TM - 110247.
28. Bischof, C., Carle, A., Corliss, G., Griewank, A., Hovland, P., "ADIFOR: Generating derivate codes from FORTRAN programs", *Scientific Programming*, vol.1, no.1, pp.11-29, 1992.
29. Grose, D., "Re-engineering the Aircraft Design Process", 5th AIAA/NASA/ USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, 1994. Also available at AIAA paper 94-4323.
30. Joshi, B., Morris, D., White, N., Unal, R., "Optimization of Operations Resources Via Discrete Simulation Modeling", AIAA 96-1181, 1996.

31. Kaufman, M., Balabanov, V., Burgee, S.; Grossman, B.; Mason, W. H.; and Watson, L. T. "VariableComplexity Response Surface Approximations for Wing Structural Weight in HSCT Design" AIAA Paper 96-0089, 34th Aerospace Sciences Meeting, Reno, NV. January, 1996.
32. Kroo, I.M., "An Interactive System for Aircraft Design and Optimization". AIAA 92-1190, Feb. 1992.
33. Sobieszcanski-Sobieski, J., Haftka, R., "Multidisciplinary Design Optimization: Survey of Recent Developments", AIAA-96-0711, 1996.
34. Gill, P., Murray, W., Saunders, M, Wright, M., *User's Guide to NPSOL*, Technical Report SOL 86-2, Dept. of Operations Research, Stanford University, California, 1986
35. Braun, R.D., Powell, R.W., Lepsch, R.A., Stanley, D.O., Kroo, I.M., "Multidisciplinary Optimization Strategies for Launch Vehicle Design", 1994.
36. Braun, R.D., and Kroo, I.M., "Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment", *Multidisciplinary Design Optimization: State of the Art*, N. Alexandrov and M. Y. Hussaini, editors, SIAM, 1995.
37. Kroo, I., Altus, S., Braun, R., Gage, P., Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," AIAA-94-4325, 1994.
38. Braun, R.D., Kroo, I., Gage, P., "Post-Optimality Analysis in Aerospace Vehicle Design", AIAA-93-3932, 1993.
39. Sobieszcanski-Sobieski, J.. " Sensitivity of Complex, Internally Coupled Systems", *AIAA Journal*, Vol. 28, No. 1, 1990.
40. Gage, P.; Kroo, I., " Development of the Quasi-Procedural Method for Use in Aircraft Configuration Optimization ", AIAA-92-4693, September 1992.
41. Haftka, R.T., Sobieszcanski-Sobieski, J., and Padua, S.L., *On the Options for Interdisciplinary Analysis and Design, Structural Optimization*, 4:65-74, 1992.
42. Barthelemy, J.F., " Engineering applications of heuristic multilevel optimization methods ", NASA CP-3031, Hampton, VA, Sept. 1988.
43. Cramer, E.J., Dennis, J.E., Frank, P.D., Lewis, R.M., Shubin, G.R.. *Problem Formulation for Multidisciplinary Optimization. SIAM Journal of Optimization*, vol 4, pg. 754-776, Nov. 1994

44. Kroo, I. *Decomposition and Collaborative Optimization for Large-Scale Aerospace Design. Multidisciplinary Design Optimization: State of the Art.* SIAM Publications. 1995.
45. Braun, R.D., Kroo, I.M. Development and Application of the Collaborative Optimization Architecture in a Multidisciplinary Design Environment. SIAM. Jan. 1996.
46. Kroo, I.M., Altus,S.S., Braun, R.D., Gage, P.J., Sobieski. I.P.. "Multidisciplinary Optimization Methods for Preliminary Aircraft Design", AIAA94-4325. Panama City FL, Sept. 1994.
47. Haftka, R.T., " Simultaneous analysis and design ", *AIAA Journal*. vol.23. pg.1099-1103, 1982.
48. Braun, R.D., Powell, R.A., Lepsch, D.O., Kroo, I.M. " Comparison of Two Multidisciplinary Optimization Strategies for Launch Vehicle Design ", *Journal of Spacecraft & Rockets*, vol. 32 pg. 404-410, March 1995.
49. Sobieszczanski-Sobieski. J.. " Optimization by decomposition: A step fro hierarchic to nonhierarchic systems ", CP-3031, Hampton, VA, Sept. 1988.
50. Renaud, J.E., Gabriele, G.A., " Approximation in non-hierarchic system optimization ". *AIAA Journal*, vol. 32, Jan. 1994.
51. Eason, E.D., Wright, J.E., " Implementation of non-hierarchic decomposition for multidisciplinary system optimizations ", AIAA 92-4822, Cleveland. OH. Spet. 1992.
52. Sellar, R.S., Batill, S.M., Renaud, J.E., " Response surface based concurrent subspace optimization for multidisciplinary system design ", AIAA 96-0714, Jan. 1996.
53. Shankar,J., Ribbens, C., Hftka, R. Watson, L., " Computational study of nonhierarchiochal decomposition alogorithm ", *Computational Optimization and Applications*, vol. 2, pg. 273-293, 1993.
54. Braun, R.D., "Collaborative Optimization: An Architecture for Large-Scale Distributed Design", Ph.D. Dissertation, Stanford University, May 1996.
55. Renaud, John; Rodriguez, Jose; "Trust Region Augmented LaGrangian Methods for Sequential Response Surface Approximation", ASME Design Engineering Technical Conferences, Sept. 1997.

56. Sobiesczanski-Sobieski, J., "Sensitivity of Complex, Internally Coupled Systems", *AIAA Journal*, Vol.28, No.1, pg.153-160, 1990.
57. Fiacco, A. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, Inc. 1983.
58. Sobieski-Sobiesczanski, J. Barthelemy, J., and Riley, K., "Sensitivity of Optimum Solutions to Problem Parameters", *AIAA Journal*, Vol. 20. No.9. Sept.1982.
59. Barthelemy, J., Sobieski-Sobiesczanski, J., "Extrapolation of Optimum Design Based on Sensitivity Derivatives", *AIAA Journal*, Vol. 21, No.5, May, 1983.
60. Beltracchi, T., Gabriele, G., "An Investigation of New Methods for Estimating Parameter Sensitivity Derivatives", Proceedings of ASME Design Automation Conference, ASME DE, Vol. 14, Sept. 1988.
61. Alexandrov, N., *Multilevel Algorithms for Nonlinear Equations and Equality Constrained Optimization*. Ph.D. Thesis, Rice University, 1993.
62. Balling, R., Wilkinson, C., "Execution of Multidisciplinary Design Optimization Approaches on Common Test Problems", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellvue, WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4033.
63. Braun, R., Moore, A., Kroo, I., "Use of the Collaborative Optimization Architecture for Launch Vehicle Design", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellevue, WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4018.
64. Braun, R., Gage, P., Kroo, I., Sobieski, I. "Implementation and Performance Issues in Collaborative Optimization", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellevue, WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4017.
65. Sobieski, I.P., Kroo,I.M., "Aircraft Design Using Collaborative Optimization", AIAA Aerospace Sciences Meeting, Reno, NV., AIAA-96-0715, Jan. 1996.
66. Holden, M., Kroo, I. "A Collocation Method for Aeroelastic Optimization", 6th AIAA Symposium on MDO, AIAA-96-3980-CP, September 1996.

67. Mason, W.H., et al., "Variable Complexity Response Surface Approximations for Wing Structural Weight in HSCT Design", AIAA-96-0089, Aerospace Sciences Meeting, Reno, NV. 1996.
68. Balabanov, Vladimir O., "Development of Approximations For HSCT Wing Bending Material Weight Using Response Surface Methodology", VPI&SU Ph.D. Thesis, Sept. 1997.
69. Cornell, John, A., *How to Apply Response Surface Methodology*, American Society of Quality Control, 1989.
70. Alexandrov, Natalia; Dennis, J.E. "A Trust Region Framework For Managing The Use Of Approximation Models In Optimization", NASA TM 112870, 1997.
71. Gill, Philip; Murray, W.; Wright, M.; *Practical Optimization*, Harcourt Brace Jovanovich, 1992, pg. 113-115.
72. Fletcher, R., "Practical Methods of Optimization", Vol.1, *Unconstrained Optimization*. Wiley and Sons, New York, 1980.
73. Sorenson, D., "Newton's method with a model trust region modification". ANL-80-106, Argonne National Lab., Argonne, Illinois, 1980.
74. Booker, A.J., Conn, A., Dennis, J., Frank, P., Serafini, D., Torczon, V., Trosset, M. "Multi-level design optimization", Technical Report ISSTECH-95-031. Boeing Information and Support Service, Seattle, WA.
75. Nelder, J., Mead, R., "A simplex method for function minimization", *Computer Journal*, vol. 7, pp. 303-313.
76. Giunta, A., Dudley, J., Narducci, R., Grossman, B., Haftka, R., Mason, W., and Watson, L., "Noisy Aerodynamic Response and Smooth Approximations in HSCT Design," AIAA Paper 94-4376, 5th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Panama City Beach, Fl., September 7-9, 1994.
77. Toropov, V.V., van Keulen, F., Markine, V. L. and de Boer, H., "Refinements in the Multi-Point Approximation Method to Reduce the Effects of Noisy Structural Response", AIAA 96-4087, pp. 941-951.
78. Barthelemy, J.F., Haftka, R., "Approximation concepts for optimum structural design-A Review", *Structural Optimization*, 5:130-144.

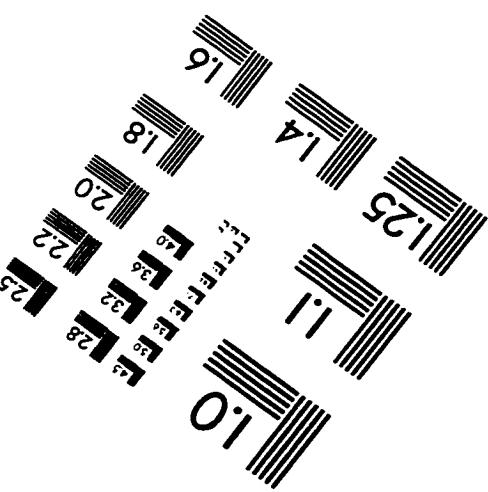
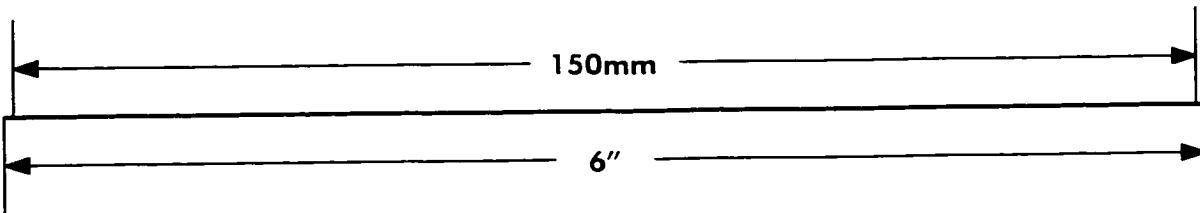
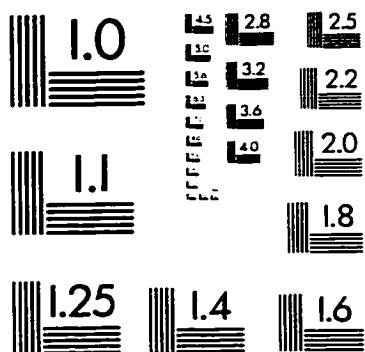
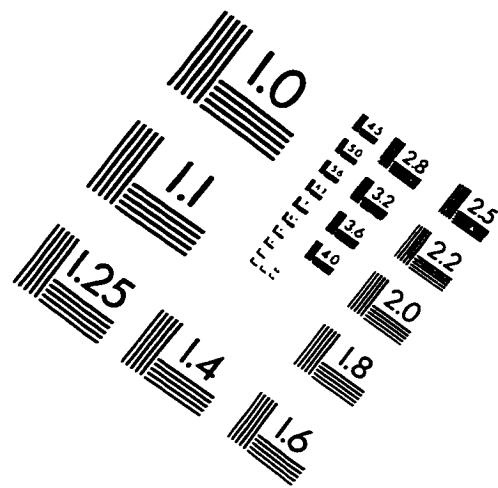
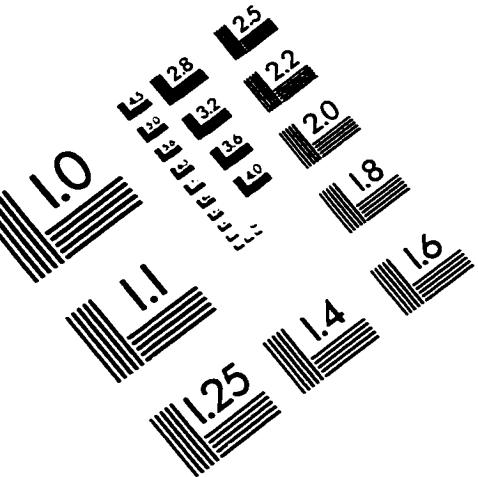
79. Conn, A., Scheinberg, K., Toint, P.L. "Recent progress in unconstrained nonlinear optimization without derivatives". *Mathematical Programming* 397-414, April, 1997.
80. Conn, A., Scheinberg, K., Toint, P.L. "Recent progress in unconstrained nonlinear optimization without derivatives", Powellfest, July, 1996.
81. Knill, D.L.; Giunta, A. A.; Baker, C.A.; Haftka, R. T.; Grossman, B.; Mason, W. H.; and Watson, L. T. "HSCT Configuration Design Using Response Surface Approximations of Supersonic Euler Aerodynamics", AIAA Paper 96-0905, 36th Aerospace Sciences Meeting, Reno, NV. January, 1996.
82. Koehler, J.R., *Design and Estimation Issues in Computer Experiments*, Dissertation, Department of Statistics, Stanford University, 1990.
83. Morris, M., Mitchell, Toby J., "Bayesian Design and Analysis of Computer Experiments: Use of Derivatives in Surface Prediction", *Technometrics*, vol. 35, No. 3, August, 1993.
84. Alexandrov, N, Kodiyalam, S., "Initial Results of an MDO Method Evaluation Study", proceedings of 7th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, St.Louis, MO, Sept.2-4, 1998. Also available as AIAA Paper 98-4884.
85. Sellar, R.. Batill, S., "Concurrent Subspace Optimization Using Gradient-Based Neural Network Approximations", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellevue, WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4019.
86. Sellar, R., Batill, S., Renaud, J. "Response Surface Based, Concurrent Subspace Optimization for Multidisciplinary System Design ", AIAA Paper 96-0714, AIAA Aerospace Sciences Meeting, January, 1996, Reno, NV.
87. Sellar, R., Stelmark, M., Batill, S., Renaud, J. "Response Surface Approximations for Discipline Coordination in Multidisciplinary Design Optimization", AIAA Paper 96-1383, AIAA Structures, Structural Dynamics, and Materials Conference, April, 1996, St. Lake City, UT.
88. Carpenter, W.C., Barthelemy, J. "A Comparison of Polynomial Approximations and Artificial Neural nets as Response Surfaces", *Structural Optimization*, Vol. 5, pp.166-74, 1993.

89. Unal, R., Lepsch, R., Engelund, W., Stanley, D. "Approximation Model Building and Multidisciplinary Design Optimization Using Response Surface Methods", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellevue, WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4044.
90. Trosset, M., Torczon, V., "Numerical Optimization Using Computer Experiments", NASA Contract #NAS1-19480
91. Dennis, J.E., and Torczon, V.: "Approximation Model Management for Optimization", AIAA-96-4099, pp. 1044-1046.
92. Torczon, V. "On the Convergence of Pattern Search Methods". TR93-10. Dept. of Computational and Applied Materials, Rice Univ., June 1993.
93. Torczon, V. "Pattern Search Methods for Nonlinear Optimization", TR95-22. Dept. of Computational and Applied Materials, Rice Univ., July 1995.
94. Carpenter, W., "Effect of Design Selection on Response Surface Performance", NASA Contractor Report 4520, June 1993.
95. Kroo, I.M., Altus, S., Braun, R.D., Gage, P.J., Sobieski, I.P., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design", AIAA/USAF 4th MDO Conference, Panama City Beach, FL., AIAA-94-4325, Aug. 1994.
96. Mason, W.H., et al., "Variable Complexity Response Surface Approximations for Wing Structural Weight in HSCT Design", AIAA-96-0089, Aerospace Sciences Meeting, Reno, NV, 1996.
97. Gill, Philip; Murray, W.; Wright, M.; *Practical Optimization*, Harcourt Brace Jovanovich, 1992, pg. 113-115.
98. Padula, S.L., Alexandrov, N. and Green, L.L.: MDO Test Suite at NASA Langley Research Center. AIAA-96-4028, pp. 410-420.
99. Wakayama, S., Page, M., Liebeck, R., "Multidisciplinary Optimization on an Advanced Composite Wing", proceedings of 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Part I, Bellevue, WA, Sept. 4-6, 1996. Also available as AIAA Paper 96-4003.
100. Alexandrov, N.: Robustness Properties of a Trust Region Framework for Managing Approximations in Engineering Optimization. AIAA -96-4102, pp. 1056-1059.

101. Rogers, J., McCulley, C., Bloebaum, C. L.. "Integrating a Genetic Algorithm into a Knowledge-based System for Ordering Complex Design Processes." accepted for Fourth International Conference on AI in Design, Standord, CA, June 1996.
102. Sobieski, I.P., Kroo, I.M., "Collaborative Optimization Using Response Surface Estimation", AIAA-98-0915, 38th Aerospace Sciences Meeting, Reno NV, 1998.
103. Renaud, John; Rodriguez, Jose; "Trust Region Augmented LaGrangian Methods for Sequential Response Surface Approximation", ASME Design Engineering Technical Conferences, Sept. 1997.
104. Balabanov, Vladimir O., "Development of Approximations For HSCT Wing Bending Material Weight Using Response Surface Methodology", VPI&SU Ph.D. Thesis, Sept. 1997.
105. Braun, R.D., Kroo, I., Gage, P., "Post-Optimality Analysis in Aerospace Vehicle Design", AIAA-93-3932, Aug. 1993.
106. Sellar, R.S., and Batill, S.M., "Concurrent Subspace Optimization Using Gradient-Enhanced Neural Network Approximations", AIAA-96-4019, Sept. 1996.
107. Cornell, John. A., *How to Apply Response Surface Methodology*, American Society of Quality Control, 1989.
108. Shevell, Richard, *Fundamentals of Flight*, Prentice Hall, New Jersey, 1989, pg. 280
109. Fletcher, R., "Practical Methods of Optimization", Vol.1. *Unconstrained Optimization*, Wiley and Sons, New York, 1980.
110. Sorenson, D., "Newton's method with a model trust region modification", ANL-80-106, Argonne National Lab., Argonne, Illinois, 1980.
111. John, R.C., Draper, N.R., "D-Optimality for Regression Designs: A Review", *Technometrics*, Vol. 17, No. 1, February, 1975, pg. 15-20.
112. Lucas, James, "Which Response Surface Design is Best", *Technometrics*, Vol. 18, No.4, November, 1976, pg. 411-6.
113. Shevell, Richard, *Fundamentals of Flight*, Prentice Hall, New Jersey, 1989, pg. 280

114. Hsu, Y.L.. *Zeroth Order Optimization Methods for Two Dimensional Shape Optimization*, Ph.D. Dissertation, Department of Mechanical Engineering, Stanford University, 1992.
115. Richards, R.A.. *Zeroth Order Optimization Utilizing a Learning Classifier System*. Ph.D. Dissertation. Department of Mechanical Engineering, Stanford University, 1995.
116. Nguyen, D.T.. "Final Report: Finite Element Software for Multidisciplinary Optimization", Old Dominion University, for NASA Langley Contract NASI-19858, 1995.
117. Storaasli, O.O., Nguyen, D.T., Agarwal, T.K., "A Parallel-Vector Algorithm for Rapid Structural Analysis on High-Performance Computers", Old Dominion University, Department of Civil Engineering.
118. Woodward, F.A., "An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow, Part I, Theory and Applications" NASA CR-2228, 1973.
119. Carmichael, R.L., Erickson, L.L., "PAN AIR - A HigherOrder Panel Method for Predicting Subsonic or Supersonic Linear Potential Flows about Arbitrary Configurations". AIAA Paper 81-1255, June 1981.
120. Manning, V., "Comparison of FESMDO Results with Nastran Baseline". Technical Memo, Stanford University, August, 1997.
121. Tang, Boxin, "Orthogonal Array-Based Latin hypercubes", *Journal of the American Statistical Association*, vol. 88, no. 424, December 1993.
122. Simpson, T., Peplinski, J., Kock, P., Allen, J., "On the Use of Statistics in Design and the Implications for Deterministic Computer Experiments", 1997 ASME DTM Conference.
123. Conn, A., Scheinberg, K., Toint, P.L. "An Algorithm Using Quadratic Interpolation for Unconstrained Derivative Free Optimization", Nonlinear Optimization and Applications, November, 1995.
124. Balabanov, V.; Kaufman, M.; Giunta, A. A.; Haftka, R. T.; Grossman, B.; Mason, W. H.; and Watson, L. T.: Developing Customized Wing Weight Function by Structural Optimization on Parallel Computers. Salt Lake City, Utah, AIAA Paper 96-1336, pp. 113-125, April 1996.

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved

