N87-11722

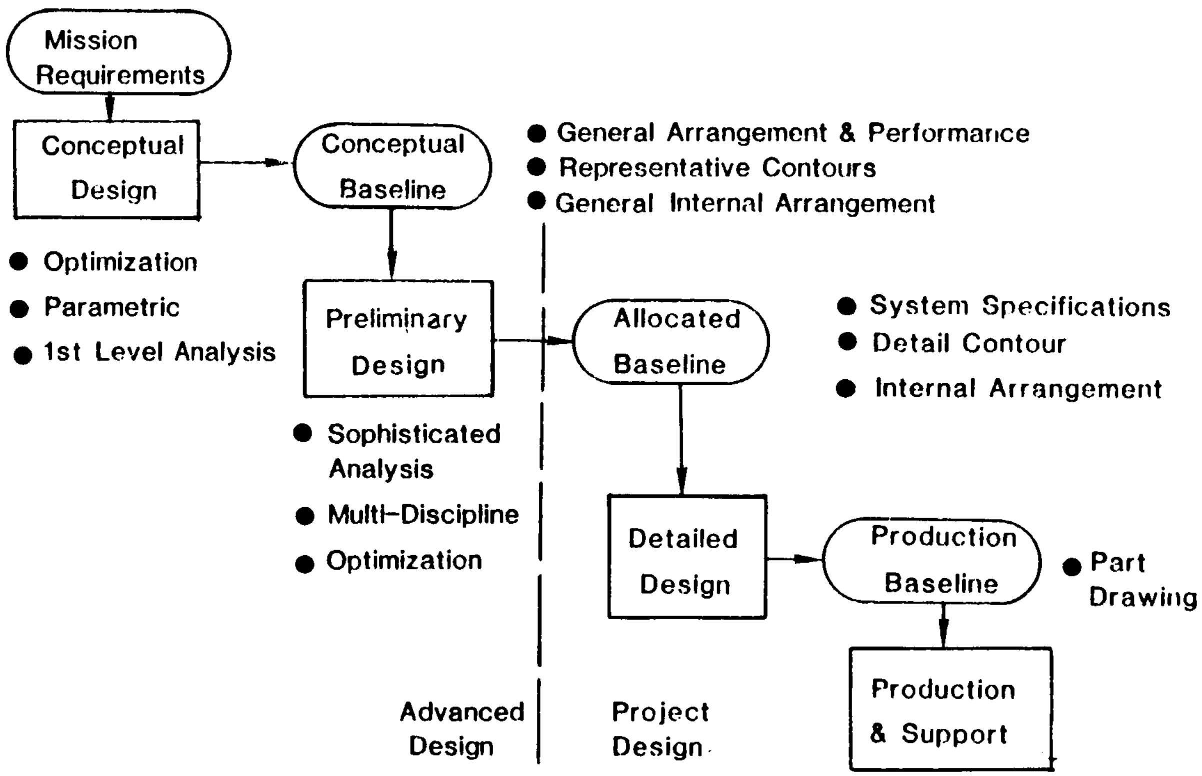
APPLICATION OF THE GENERALIZED REDUCED GRADIENT METHOD TO CONCEPTUAL AIRCRAFT DESIGN

Gary A. Gabriele
Lockheed-Georgia Company

AIRCRAFT DESIGN PHASES

The complete aircraft design process can be broken into three phases of increasing depth: conceptual design, preliminary design, and detail design. Conceptual design consists primarily of developing general arrangements and selecting the configuration that optimally satisfies all mission requirements. The result of the conceptual phase is a conceptual baseline configuration that serves as the starting point for the preliminary design phase.

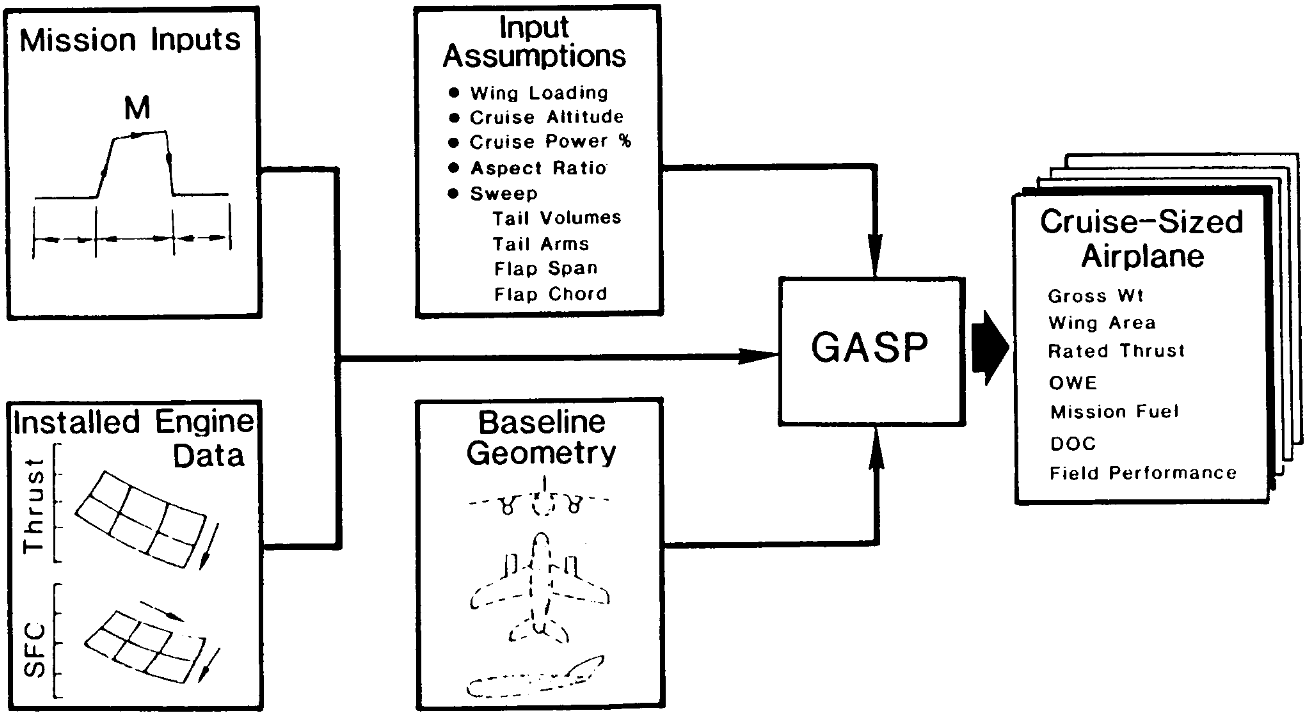
The conceptual design of an aircraft involves a complex trade-off of many independent variables that must be investigated before deciding upon the basic configuration. Some of these variables are discrete (number of engines), some represent different configurations (canard vs conventional tail) and some may represent incorporation of new technologies (aluminum vs composite materials). A particular combination of these choices represents a concept; however there are additional variables that further define each concept. These include such independent variables as engine size, wing size, and mission performance parameters, which must be selected before a particular configuration can be evaluated. Generally, these additional variables are chosen to optimize each concept before selecting a final configuration.



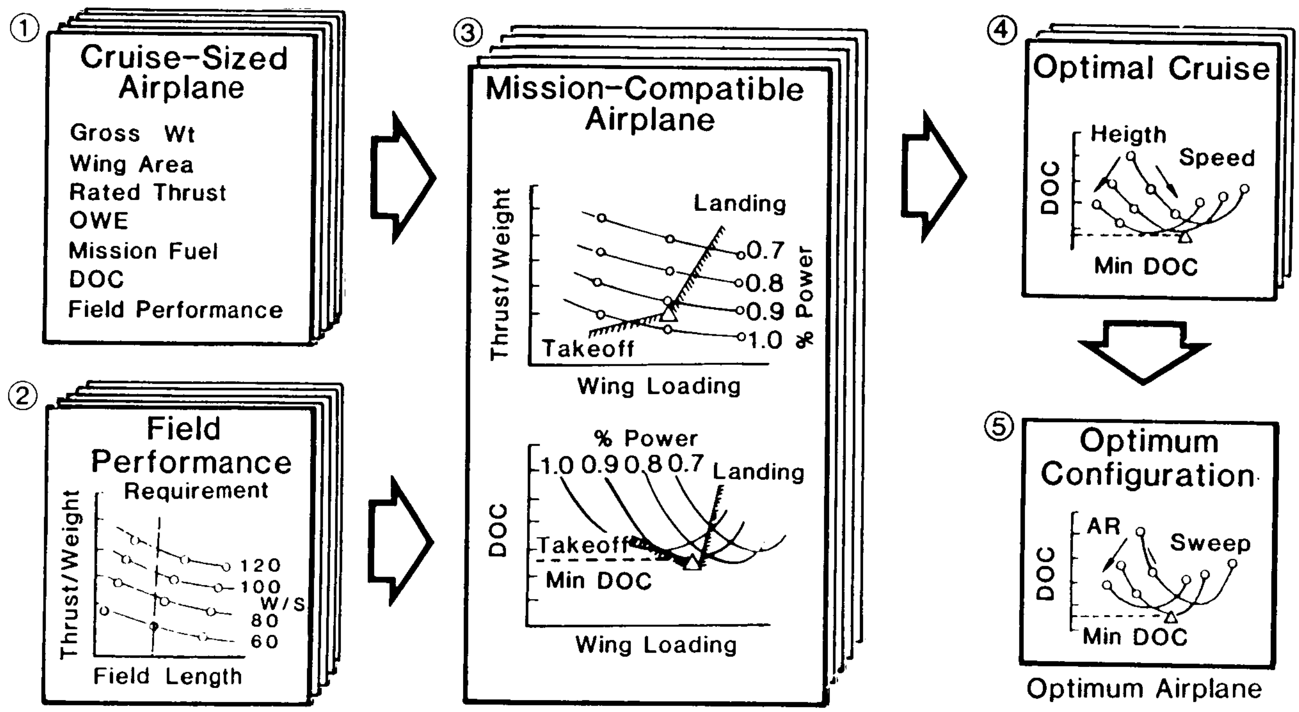
OPTIMAL VEHICLE SELECTION BY PARAMETRIC DESIGN

The principal analysis tool used during the conceptual design phase is the sizing program. At Lockheed-Georgia, the sizing program is known as GASP (Generalized Aircraft Sizing Program). GASP is a large program containing analysis modules covering the many different disciplines involved in defining the aircraft, such as aerodynamics, structures, stability and control, mission performance, and cost. These analysis modules provide first-level estimates the aircraft properties that are derived from handbook, experimental, and historical sources.

To make a run of the sizing program, the engineer develops a data set defining the fuselage geometry, the mission profile, a candidate propulsion system, the general arrangement of the components, the extent of new technologies to be incorporated into the design, and the values for the independent design variables. The sizing program provides a complete weight breakdown of the airplane, aerodynamic properties, mission and airport performance, center of gravity ranges, and cost data.



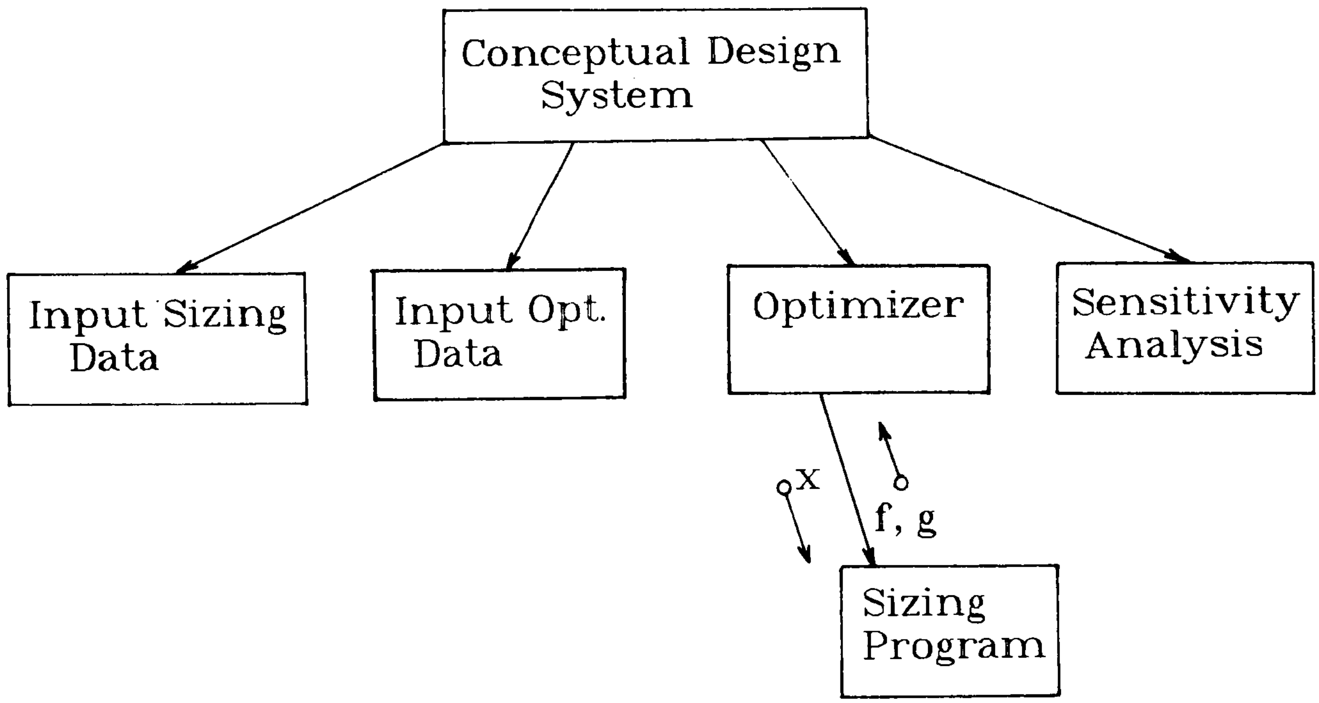
To optimize a design, the engineer must choose a selection criteria such as minimum weight and determine design constraints that define feasible designs. He is then faced with the classical design optimization problem: find the optimal values of the independent design variables that minimize the selection criteria and satisfy the design constraints. Without some automated optimization method, this process is generally performed by plotting the results of the sizing program obtained by parametically varying the independent variables throughout their ranges. For a problem involving 4 design variables, this may result in as many as 256 runs of the sizing program. This can be a very time-consuming process when many different designs must be investigated.



AUTOMATED CONCEPTUAL DESIGN SYSTEM

This paper describes our experiences in combining a numerical optimization algorithm with the aircraft sizing program to obtain an automated conceptual design system. The structure of the system is shown below indicating that the optimizer functions as a black box interacting with the sizing program, which provides the required function values. Such a structure allows substitution of any appropriate optimization algorithm with very little impact on the sizing program, or changes to the sizing program with very little effect on the optimizer.

In the past decade, advances in optimization methods have produced several algorithms that have proven to be both reliable and robust in a number of engineering applications. One of these is the Generalized Reduced Gradient (GRG) method. The GRG method is an extension of the reduced gradient method for linear constraints to the nonlinear case from which highly robust and efficient implementations (refs. 1, 2, 3) have been produced. It is this method that we have chosen for the optimizer.



THE NONLINEAR PROGRAMMING PROBLEM

The general nonlinear programming problem (NLP) can be stated as shown below. The function f(x) is a scalar function representing the criteria to be optimized and x is a vector of design variables. The h(x) functions represent equality constraints that require specific combinations of the design variables, and the g(x) functions represent inequality constraints that define feasible regions in the design space. All functions are assumed to be nonlinear.

Minimize $f(x)$; $x = [x_1, x_2, \ldots, x_N]^T$

Subject To

$$g_j(x) \geq 0 \qquad j = 1, 2, \ldots, J$$

$$h_k(x) = 0 \qquad k = 1, 2, \ldots, K$$

Where

$f(x)$ = Objective Function

x = A Column Vector Of Design Variables

$g_j(x)$ = Inequality Constraints

$h_k(x)$ = Equality Constraints

x^0 = Starting Point

x^k = Candidate Point

GENERALIZED REDUCED GRADIENT METHOD

The GRG method restates the NLP in the form shown below, where the vectors x^L and x^O represent the lower and upper bounds on the design variables x. The inequality constraints are included as equality constraints through the addition of slack variables. The parameter M represents the total number of constraints. The constraints include only the functional constraints; variable bounds are accounted for separately to allow for a more efficient handling of this special class of constraints.

The basic strategy of the GRG method is derived from trying to use each equality constraint $h_m(x)$ to eliminate a design variable from the problem. However, for most engineering problems, the constraints are too complex to allow this substitution. The GRG method accomplishes this by employing the Implicit Function Theorem.

Minimize $f(x),\ x = [x_1, x_2, ..., x_N]^T$

Subject to

$$h_m(x) = 0 \qquad m = 1, 2, ..., M$$

$$x^L < x < x^U$$

Strategy:

Solve each $h_m(x)$ explicity for a Variable and Substitute into $f(x)$.

Problem: Not always Possible for Complex Engineering Functions or Simulations.

Solution: Do It Implicity.

DERIVATION OF THE REDUCED GRADIENT

Consider the following strategy, whose foundations can be found in the simplex method of linear programming. Divide the design vector x into two classes, nonbasic or independent (z) variables and basic or dependent (y) variables, as shown in the figure, where Q = N - M. The search for the optimum will occur by searching in the design space of the nonbasic variables and the basic variables will be used to satisfy the constraints. A gradient vector for this new problem can be obtained by introducing the division of the design variables into the objective and constraint functions and following the steps shown in equations (1) to (3).

The reduced gradient defines the rate of change of the objective function with respect to the nonbasic variables with the basic variables adjusted to maintain feasibility. In the presence of linear constraints, equation (3) represents the changes necessary in the basic variables for a given change in the nonbasic variables. Additional adjustment is necessary in the nonlinear situation. Conceptually, the above derivation corresponds to a transformation of the GRG problem into one having the following form:

Minimize: $F(z) \quad z = (z_1, z_2, \ldots, z_q)^T$

Subject to: $z^l \le z \le z^u$

where the basic variables y have been eliminated from the original problem by using the constraints $h_m(z,y) = 0$ to solve for y in terms of z. The gradient of F(z) is represented by the reduced gradient, and the necessary equations for y in terms of z represented by equation (3).

Divide X into two classes, dependent and independent

$X = [y,z]^T$

$Y = [y_1, y_2, \ldots, y_M]^T$ Dependent Variables

$Z = [z_1, z_2, \ldots, z_Q]^T$ Independent Variables

Calculate the first variation of f(X) and H(x) using Z and Y

$$df(x) = \nabla_z f(x)^T dz + \nabla_Y f(x)^T dy \qquad (1)$$

$$dH(x) = \nabla_z H(x)\, dz + \nabla_Y H(x)\, dy = 0 \qquad (2)$$

Solve (2) for dy

$$dy = -[\nabla_Y H(x)]^{-1} \nabla_z H(x)\, dz \qquad (3)$$

Substitute (3) for dy in (1) to arrive at the REDUCED GRADIENT

Reduced Gradient $\nabla_R F(z)$

$$\nabla_R F(z)^T = \nabla_z f(x)^T - \nabla_Y f(x)^T \nabla_Y H(x)^{-1} \nabla_z H(x) \qquad (4)$$

The Reduced Gradient defines the gradient for the new *reduced* problem

Minimize $F(z),\ z = [z_1, z_2, \ldots, z_Q]^T$

Subject to $z^L \le z \le z^U$

the change in Y necessary to maintain feasibility is defined by equation (3) for linear constraints.

CONVERGENCE PROPERTIES

A necessary condition for the existence of a local minimum of an unconstrained nonlinear function is that the elements of the gradient vanish. Similarly, a local minimum of the reduced problem shown in the previous figure occurs when the elements of the reduced gradient satisfy the conditions shown below.

Points that satisfy these conditions satisfy the Kuhn-Tucker conditions for the existence of a constrained relative minimum of the original NLP problem (ref. 3). An additional benefit of this method is that the Lagrange multipliers are calculated in the course of calculating the reduced gradient vector.

Convergence Conditions

$$\nabla_R F(z)_i \begin{cases} < 0 & \text{if } z_i = z_i^{u} \\ > 0 & \text{if } z_i = z_i^{\,l} \\ = 0 & \text{otherwise} \end{cases}$$

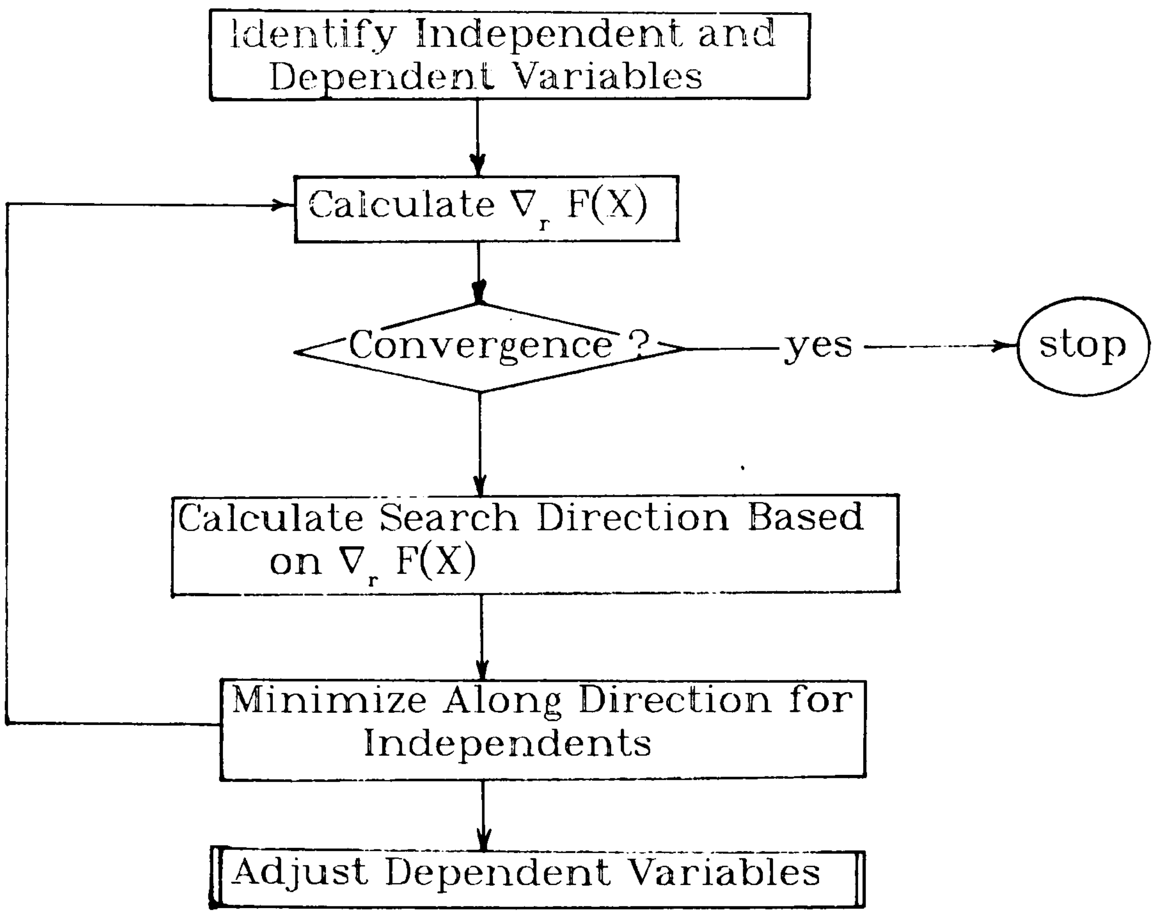
$$i = 1, 2, 3, \ldots, Q$$

When this conditon holds, the corresponding point X satisfies the Kuhn–Tucker conditions for the existence of a local constrained minimum of the original problem.

GENERALIZED REDUCED GRADIENT ALGORITHM

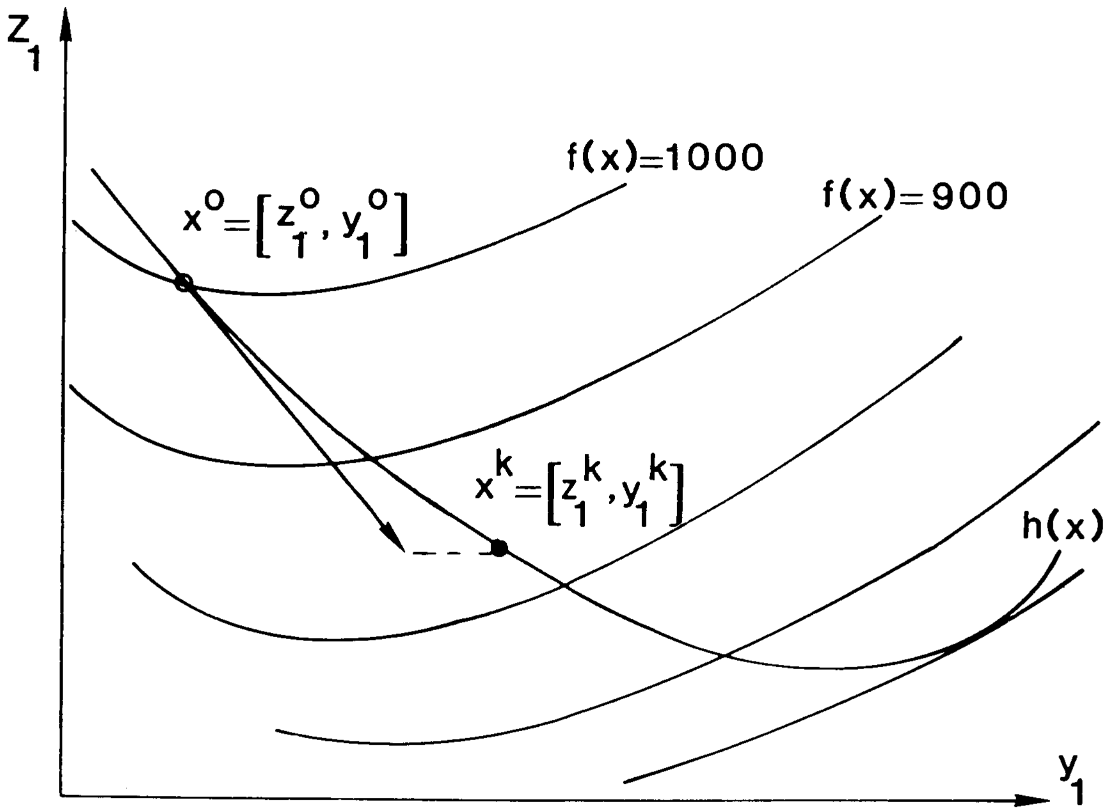
The basic steps of the GRG algorithm are given in this figure. The method looks very much like any gradient based method, with some exceptions. The search directions for the nonbasic variables are based on the reduced gradient vector and initial directions for the basic variables are then calculated from equation (3). In the calculation of the nonbasic direction any gradient-based search method, such as conjugate gradient or variable metric, may be used.

The line search phase is also similar, except additional logic is also required to adjust the basic variables and determine when a new constraint is encountered. The basic variable adjustment occurs in the presence of nonlinear constraints. As we move along the search direction defined for the nonbasic variables and calculated from equation (3) for the basics, we can expect, for nonlinear constraints, that the trial points will violate the constraints. To maintain feasibility, an adjustment of the basic variables at each trial point is undertaken to get back to the constraint surface before evaluating the objective function. During this adjustment the independent variables are held constant. The line search is terminated by one of the following conditions: a relative local minimum was located along that search direction, a new constraint was encountered which limited the search, or adjustment of the basic variables to maintain feasibility was not possible at some trial points.



DEPENDENT VARIABLE ADJUSTMENT

This figure depicts the adjustment of the dependent variable y_1 during the line search phase of the GRG algorithm. Here we have taken a step along the search direction from x^o. Holding the independent variable z_1 constant, we now adjust y_1 to get back to the constraint h(x).



METHODS FOR DEPENDENT VARIABLE ADJUSTMENT

A modified Newton method is usually employed to adjust the basic variables during the line search. The iteration sequence is given below, where A_0 is the initial inverse of $\nabla_y h(x)$ used at the start of the GRG iteration to calculate the reduced gradient and t is an iteration of Newton's method.

The modified Newton method has been used in all current implementations of the GRG algorithm. This is due primarily to the substantial savings in computation time obtained by avoiding successive reformulations of the Jacobian inverse. However, the major drawback of the method is that it does not possess the convergence rate of the classical method obtained by evaluating the Jacobian and its inverse at every Newton iteration. Poor convergence of the Newton method can lead to insufficient progress being made during a line search, which may hinder convergence of the algorithm to the optimal solution.

Two factors that have a major influence on the convergence are the approximations to the basic variables during the line search and the inaccuracies of using the inverse A_0. Suggestions for improving the former have appeared in Lasdon (ref. 2), and Gabriele and Ragsdell (ref. 3), and both offer improvements in convergence.

Techniques for improving the inverse A_0 have appeared in the literature for solving nonlinear systems of equations. Broyden's method (ref. 4) is one of these methods and is summarized below. This method is used in our implementation of the GRG algorithm.

MODIFIED NEWTON METHOD

$$y^{t+1} = y^t - A_0 H(z^k, y^t)$$

z^k = fixed values of independents

A_0 = initial inverse of $\nabla_y H(x)$ used in calculating $\nabla_r f(x)$

BROYDEN'S METHOD

The inverse Jacobian matrix A is updated at each iteration by

$$A_{i+1} = A_i - (A_{i+1} v^T - p_i s_i) p_i^T A_i / (p_i^T A_i v_i)$$

$$v_i = H(z^k, y^{t+1}) - H(z^k, y^t)$$

$$p_i = -A_i H(z^k, y^t)$$

$$y^{t+1} = y^t + s_i p_i$$

CONCEPTUAL AIRCRAFT DESIGN PROBLEM 1

We will now discuss two example problems that demonstrate the effectiveness of this method for conceptual aircraft design. The first problem is typical of the type of problem that is generally solved very early in a conceptual study. The number of design parameters and constraints is small (ref. 5) but is large enough to preclude the use of graphical techniques.

In this problem we are required to minimize the takeoff gross weight (TOGW) of a transport aircraft that will be required to fly a simple climb-cruise mission. The design variables are cruise altitude (H), wing loading (W/S), wing aspect ratio (AR), engine cruise power setting (PS), and wing sweep (SWEEP). The constraints and variable bounds are shown in the figure.

This problem posesses some interesting scaling problems that must be addressed before we can be sure a numerical optimization technique can be effectively applied. The variable scales range from multiples of 1000 for altitude to less than 1 for power setting. The constraints range from values less than 1 for lift coefficient to thousands of feet for takeoff distance. The engineer must be sensitive to these differences when establishing convergence criteria and constraint tolerances. The algorithm should be able to provide some help and should be as insensitive to scale as possible. This is more true of the GRG algorithm than some other available algorithms such as the penalty function based methods.

Minimize: TOGW(H, W/S, AR, PS, SWEEP)

Subject to:

Cruise $C_L \leqq C_L$ limit

Fuel Volume Ratio $\geqq$ 1.05

Take off Distance $\leqq$ 10500 ft.

Rate of Climb $\geqq$ 300 fpm.

Approach Speed $\leqq$ 150 knots

$31000 \leqq H \leqq 40000$ ft

$90 \leqq W/S \leqq 190$

$6 \leqq AR \leqq 14$

$.7 \leqq PS \leqq 1.$

$10 \leqq \text{Sweep} \leqq 35$ deg.

PROBLEM 1 RESULTS

The results shown in the figure were obtained using a modified version of the OPT program (ref. 5). All variables were scaled between 0. and 10. followed by a scaling of the objective and constraint partials using the approach developed by Root and Ragsdell (ref. 6). Each constraint was scaled by the engineer to avoid trying to obtain unreasonable values when the constraints were active.

The final solution has three functional constraints active and one variable bound active, leaving one degree of freedom. The problem terminated with the norm of the reduced gradient below the tolerance.

The functions evaluation refers to the number of times the sizing program was called. This is an important quantity because the time spent performing a function evaluation using the sizing program far outweighs the time spent by the optimizer generating trial points. This number compares favorably with that required to perform the analysis graphically. This solution required about 2-3 hours of elapsed time.

To solve this problem using a graphical technique such as carpet plotting would require approximately 4 calls to the sizing program for each design variable, or 1024 aircraft sizings. Even if we were to solve this problem using only 4 design variables, we would require about 256 calls to the sizing program. In addition to this, we would have to add the time required to plot and solve for the optimum. For a problem of this size we can expect an experienced engineer to take 1 to 2 days.

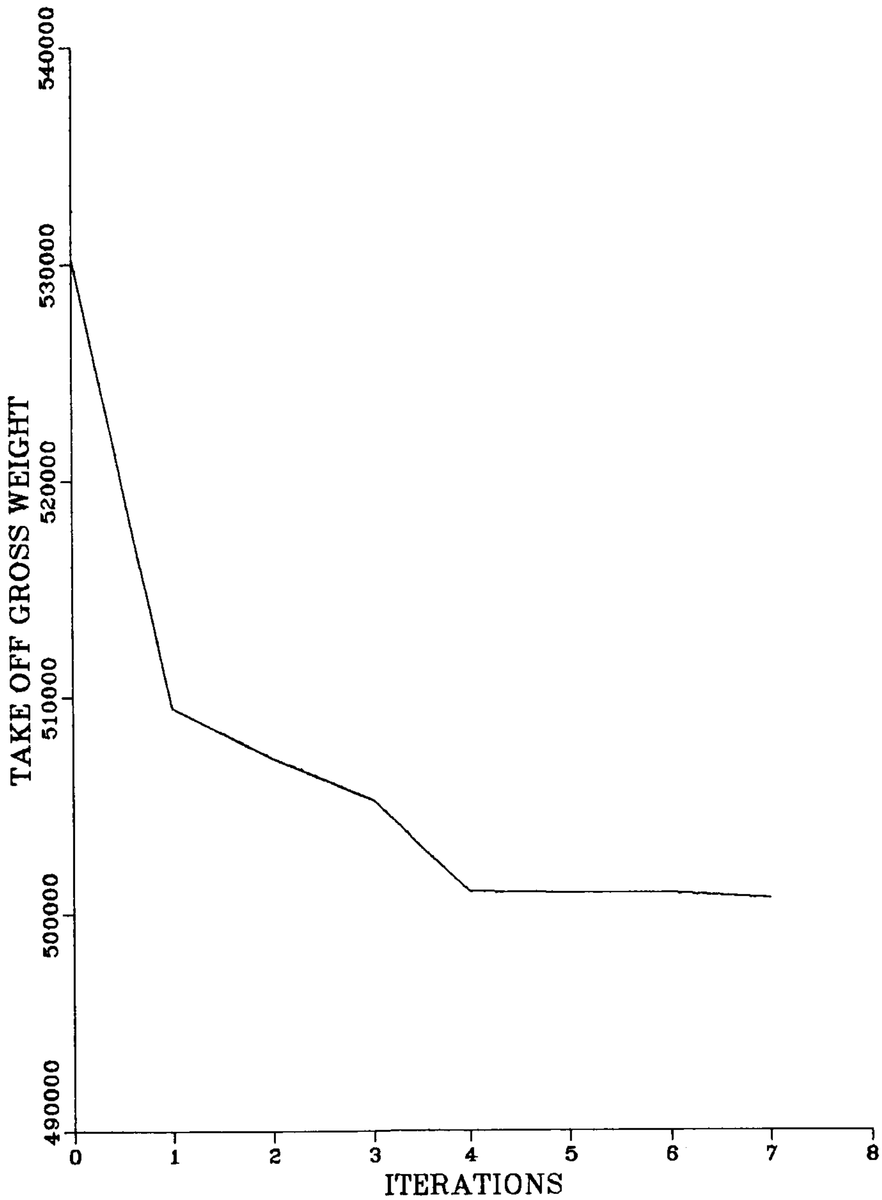
	Start Pt	Final
H	33000	31000
W/S	120	154.1
AR	9	7.6
PS	.9	.91
SWEEP	20	25.5
TOGW	530,278	500,727

Iterations	7
Functions Eval.	73

Active Constraints: x_1^L,
constraints 2,4,5

PROBLEM 1 ITERATION HISTORY

As can be seen from this figure, the progress to the optimum was fairly rapid. A review of the output produced by the optimizer would show that each 'kink' in the curve (iterations 1, 3 and 4) corresponds to one of the constraints being encountered. This points to the strength of the GRG method for engineering problems; as it locates a constraint or the intersection of two or more constraints, it can easily track or follow that constraint to an optimum. In aircraft design, our optimal design points generally lie on one or more constraints.



SENSITIVITY ANALYSIS

Another important feature of the GRG algorithm is the generation of the Lagrange multipliers. The Lagrange multipliers allow the engineer to check the sensitivity of the objective function to changes in the active constraints. For this problem, the lower limit on cruise altitude was set at 31000 feet, and the resultant optimum altitude was at this limit. Using the Lagrange multiplier printed for this constraint, the engineer can use the procedure shown below to estimate how much the optimum objective function value would change if he were to lower the limit to 30000 feet. We see that the estimated change from the sensitivity analysis is 499,251 lb, which compares favorably with the result (498,945 lb) obtained by re-optimizing the problem with the new lower limit. Making the Lagrange multipliers available to the engineer allows him to interpret the results of his optimization more effectively and have more confidence in the results produced by the optimizer.

Change in optimal value of f(x) can be estimated by:

$$\Delta f = \mu_j \, \Delta g_j$$

where μ_j = Lagrange multiplier
Δg_j = change in active constraint

For our problem, lower bound on x_1 is active with a corresponding multiplier value

$$\mu_1 = 1.47531$$

Change lower bound from 31000 to 30000, $\Delta g_1 = -1000$

New optimal f(x) = 499,251 lb. from above analysis

Re–optimization produces f(x) = 498,945 lb. (0.06 % difference)

CONCEPTUAL AIRCRAFT DESIGN PROBLEM 2

The second example problem demonstrates an expansion of the original problem to allow for optimizing and balancing of the aircraft in one step. In the first problem we presented, the balance and loadability of the aircraft were ignored. Usually the engineer fixes values for the variables that effect the balance of the aircraft at the start of the optimization, performs the optimization, then checks the balance of the aircraft. If the aircraft is not balanced, changes in the balance parameters are made and the problem is re-optimized. This continues until he produces a balanced, optimal design. For most conventional configurations, this occurs in about 2-3 cycles of this process. For unconventional configurations for which there is little experience, balancing may take considerably longer.

In this problem we have included the balance parameters, wing position, main gear position, and horizontal and vertical tail coefficients as design variables. We also have included eight additional constraints that will define the balance of the aircraft. This problem will allow us to balance the aircraft at the same time that we optimize the other system parameters. This eliminates the need to perform the above cycle of re-optimization and provides an effective method by which stability and control requirements and loadability requirements can be integrated within the sizing process. The disadvantage to this approach is that we have almost doubled the number of variables and possible active constraints that the optimizer must handle.

Minimize: TOGW (H, W/S, AR, PS, SWEEP, WING POSITION, MAIN GEAR POSITION, HORIZONTAL AND VERTICAL TAIL COEFFICIENTS)

Subject to:

Cruise $C_L \leqq C_L$ limit

Fuel Volume Ratio ≧ 1.05

Take off Distance ≦ 10500 ft.

Rate of Climb ≧ 300 fpm.

Approach Speed ≦ 150 knots

Forward and Aft C.G. limits required for S&C

Minimum Vertical Tail Size for Engine Out and Control

Minimum Nose Gear Load under Critical Loading Conditions (5)

PROBLEM 2 RESULTS

The results for this larger problem are shown below. (The design concept is different from the previous example, therefore comparison of weight is meaningless.) Again, this problem presents a challenge in variable and constraint scaling for the optimizer that was handled in the same manner as for problem 1.

As can be seen, the number of function evaluations is still low relative to the size of the problem. The majority (117) of the evaluations were spent calculating the numerical gradients.

In addition to the lower limit on altitude and the rate of climb specification, the active constraints for this problem were the three stability and control constraints (6, 7 and 8) on the tail sizes, and the minimum nose gear load under one of the 5 critical loading conditions (constraint 13). This last constraint contributes mostly to limiting the main gear location. This solution corresponds to within .5% of a result obtained using the old method described earlier.

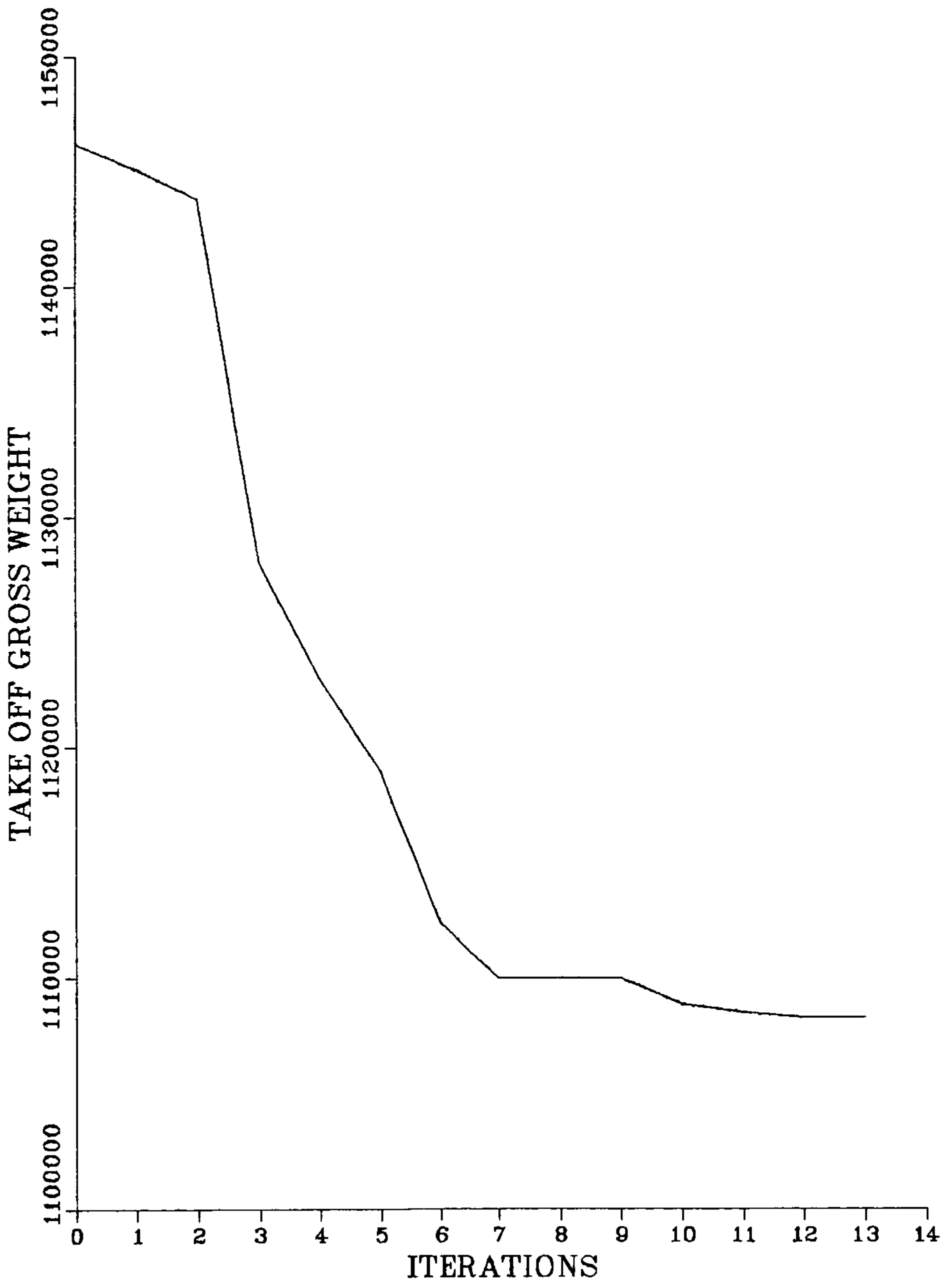
	Start Pt	Final Pt
H	32000	31000
W/S	140	154.1
AR	6.5	7.6
PS	.9	.91
SWEEP	20	25.5
WING POS.	.463	.479
M.G. POS.	.697	.652
V_H	.655	.524
V_V	.090	.079
TOGW	1,146,220	1,108,335

Iterations 13
Functions Eval. 189

Active Constraints: x_1^L,
constraints 4, 6, 7, 8, 13

PROBLEM 2 ITERATION HISTORY

The figure below illustrates that the method made good progress toward the optimum and was close after about seven iterations. The problem terminated again with the norm of the reduced gradient below the selected criterion. The elapsed time for this problem was between 3-4 hours. The advantage here is that the final optimal design is also an aircraft that is acceptable with respect to stability and control and loadability requirements. This provides a valuable design tool for those new concepts or configurations that prove difficult to balance.



CONCLUSIONS

We have seen from these two examples that numerical optimization provides substantial improvements in designer productivity over graphical techniques. This allows the designer to investigate many more designs and concepts at a very crucial time during the design process.

In our experience, the GRG algorithm provides a very reliable method for conceptual aircraft optimization. The automated conceptual design system is used on a daily basis at Lockheed in all conceptual design studies. The basic ability of the method to easily locate optimum points that lie on constraint boundaries appears to be well suited to this type of problem.

We have seen in the second design example that optimization can be used to help solve design problems in which we have limited design experience. In fact, we can now use optimization to formulate new design methods in areas in which it is difficult to understand the interaction among design parameters and new technologies or concept. This is particularly true in conceptual aircraft design, in which innovation is more or less the rule.

The automated conceptual design system is used by engineers who are not optimization experts. These engineers have been trained in how the optimizer works and how to evaluate the results. But they often still require help in the development of new formulations or in resolving whether the optimizer has truly reached a solution. For these situations our experience suggests that someone with a strong optimization background should be a member of any conceptual design study.

- Numerical optimization provides substantial improvements in designer efficiency over manual techniques.
- The GRG method is a reliable method for conceptual aircraft optimization.
- Numerical optimization can help solve difficult design problems where conventional wisdom is lacking.
- A team concept employing an optimization expert and an experienced designer is essential.

REFERENCES

1. Abadie, J., and J. Carpenter, "The Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," in Optimization (R. Fletcher, Ed.), Academic Press, New York, 1969.

2. Lasdon, L. S., A. D. Waren, A. Jain, and M. Ratner, "Design and Testing of a Generalized Reduced Gradient Code for Nonlinear Programming," ACM Trans. Math. Software, Vol. 4, No. 1, pp. 34-50, 1978.

3. Gabriele, G. A. and K. M. Ragsdell, "The Generalized Reduced Gradient Method: A Reliable Tool for Optimal Design," ASME J. Eng. Ind., Vol. 99, No. 2, pp. 384-400, May 1977.

4. Broyden, C. G., "A Class of Methods for Solving Nonlinear Simultaneous Equations," Math. Comp., Vol. 21, pp. 368-381, 1965.

5. Gabriele, G. A. and K. M. Ragsdell, "OPT: A Nonlinear Programming Code in Fortran IV - User's Manual," Purdue Research Foundation, West Lafayette, IN, Jan. 1976.

6. Root, R. R., and K. M. Ragsdell, "Computational Enhancements of the Method of Multipliers," ASME J. Mech. Des., Vol. 102, pp. 517-523, 1970.