JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 33, No. 2, March–April 1996

Decomposition Algorithm for Performance Optimization of a Launch Vehicle

M. Rahn* and U. M. Schöttle†
Stuttgart University, 70550 Stuttgart, Germany

A decomposition algorithm is described that performs both the flight-path and system design optimization tasks for a space transportation system by a two-level optimization scheme, partitioning the entire mission into a sequence of flight segments. These segments are optimized separately on a subproblem level employing nonlinear programming methods, whereas a main iteration controller subsequently combines the subarc solutions, determining subproblem targets and master problem controls that optimize the general objective. In addition, trajectory optimization is performed on a parallel processing workstation cluster to reduce the overall computational time. Optimization examples are presented first to verify the coupled trajectory and design optimization procedure applied to the mission of an airbreathing Sänger-type space transportation system and second to demonstrate the enhanced performance of the parallel algorithm.

Nomenclature

a	= acceleration, m/s^2
F, f	= objective functions
$\boldsymbol{g}$	= restriction vector
h	= flight altitude, m
M	= Mach number
m	= mass, kg
$\boldsymbol{p}, \boldsymbol{q}, \boldsymbol{l}, \boldsymbol{\pi}$	= optimization parameter vectors
$\boldsymbol{s}$	= search direction vector
t	= time, s
$\boldsymbol{u}$	= control vector
v	= velocity, m/s
$\boldsymbol{x}$	= state vector
α	= angle of attack, deg
γ	= flight path angle, deg
δ	= latitude, °
λ	= longitude, °
μ	= mass fraction
τ	= time interval, s
χ	= flight heading, deg
ψ	= airbreathing-engine thrust ratio

Subscripts

COM	= communication
elp	= elapsed computer time
l, u	= lower, upper bound
st	= staging
$0, E$	= initial, end value

Introduction

THE predictions of the performance capabilities and competitiveness of the many launch-vehicle concepts presently being investigated[1–5] rely on extensive numerical modeling, simulation, and optimization. The performance optimization of future space transportation systems (STSs) implies the two tasks of system design and trajectory optimization, which for fully reusable launch systems has to include both the ascent flight legs of different vehicle stages and the re-entry or flyback mission segments.

In principle, this problem can be solved by collecting all elements of the trajectory control vector and system design variables in one vector of optimization parameters to be manipulated by an appropriate nonlinear programming (NLP) optimization algorithm. This approach has recently been applied successfully to the ascent mission of a rocket-powered single-stage-to-orbit vehicle in a multidisciplinary design environment.[6,7] Our experience, however, has shown poor convergence properties even for the less complex mission example of an expendable multistage rocket launcher, when major system design parameters such as, e.g., the mass split of stages or engine sizing were included to optimize trajectory control and vehicle parameters simultaneously.

To improve on this disappointing situation, a multistep sequential optimization procedure has since been developed, which combines optimization cycles and separate vehicle-design-related analyses that cannot be included in the optimization steps.[8,9] Though this scheme was able to solve the optimization problem of a two-stage, winged rocket launch vehicle designed for vertical takeoff, severe convergence problems were encountered when it was applied to the more complex mission of an airbreathing Sänger-type STS. These difficulties were attributed in part to different performance sensitivities of the various flight phases, controls, and major system design parameters, and to scaling problems.[8] The observations suggest that optimization of different ascent, re-entry, and flyback branches of an airbreathing transportation system should be tried in a sequence of subproblems to separate stage 1 flight and design sensitivities from those of the orbiter stage. Therefore a decomposition approach has been taken in the present study to solve the overall optimization problem of a Sänger-type launch system. This attempt has been encouraged by ascent optimization results for expendable rocket launchers obtained with decomposition methods.[10–12]

Decomposition of a mission means partitioning the trajectory into subarcs such that each mission segment can be optimized independently. These subproblems constitute the first level of optimization. A second-level controller is then used to optimize the entire mission. Hence, a two-level optimization procedure results, with the master-level algorithm optimally coordinating the solution of the subproblems. A significant disadvantage of this technique is associated with the high computational cost, because each of its main iterations requires the re-solution of all of the subproblems. This drawback, however, may be compensated for by the attractive feature that subproblems can be solved independently on a parallel-processing computer, reducing the computational time for problem solution.

Parallel computations in trajectory optimization have been proposed by many researchers as a way to accelerate the solution.[13,14] Hence, a second effort is aimed at exploring the potential for improvement by performing parallel computations of functions, constraints, and their derivatives required for optimization.

Received Sept. 15, 1995; revision received Oct. 19, 1995; accepted for publication Nov. 28, 1995.

*Research Engineer, Institut für Raumfahrtsysteme.
†Research Engineer, Institut für Raumfahrtsysteme; Head, System and Mission Analysis Department. Senior Member AIAA.

In this paper, a two-level optimization scheme is described, and results are presented to demonstrate the capability of the methodology to optimize the performance of an airbreathing two-stage launch vehicle. The standard NLP formulation of the flight performance optimization problem is described, and is extended by the decomposition method. Optimal solutions are obtained during the numerical validation for a complex Sänger-type STS mission, where other optimization techniques previously applied have failed. Finally, results of trajectory optimizations utilizing a parallel-processing workstation cluster are presented.

Problem Description

The flight performance optimization task may be formulated as a NLP problem: Minimize

$$F(\boldsymbol{p}) \tag{1}$$

subject to

$$g_j(\boldsymbol{p}) = 0, \qquad j = 1, \ldots, m_e \tag{2}$$

$$g_j(\boldsymbol{p}) \geq 0, \qquad j = m_e + 1, \ldots, m \tag{3}$$

and

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}[\boldsymbol{x}(t), \boldsymbol{u}(t)], \qquad \boldsymbol{x}(t = 0) = \boldsymbol{x}_0 \tag{4}$$

with

$$\boldsymbol{u}(t) = \boldsymbol{T}(\boldsymbol{p}, t), \qquad t \in [t_0, t_E] \tag{5}$$

$$p_i^l \leq p_i \leq p_i^u \tag{6}$$

For flight performance assessment of STS two direct methods have been conceived in the past for approximating the solution of the optimal-control problem involved in a finite-dimensional space, which approximate either both the state variable $\boldsymbol{x}$ and the control function $\boldsymbol{u}$ or the controls alone by a finite set of parameters. The former (direct transcription) formulation has received much attention and is claimed to be less sensitive and more robust in attaining optimum solutions of trajectory problems with flight-path constraints.[12,15,16] It requires, however, a comparatively large number of parameters to be determined optimally with an appropriate NLP algorithm to represent $\boldsymbol{x}$ and $\boldsymbol{u}$. The alternative second approach, in which only the controls are parametrized, has seen widespread application in trajectory optimization and has been successfully employed in the evaluation of a large variety of space transportation systems.[4,5,17]

In this paper we shall limit our attention to the latter technique, which explicitly approximates only the control function. This problem formulation converts the original continuous optimal-control problem into one of finite-dimensional parameter optimization and is chosen because efficient algorithms exist to solve the standard mathematical programming problem (1–3). It requires the control vector function $\boldsymbol{u}(t)$ to be represented by a transformation function $\boldsymbol{u}(t) = \boldsymbol{T}(\boldsymbol{p}, t)$, $t \in [t_0, t_E]$, of a parameter vector $\boldsymbol{p}$. For this purpose, the whole trajectory interval $[t_0, t_E]$ is split into K subintervals $[t_0, t_1), [t_1, t_2), \ldots, [t_{K-1}, t_K = t_E]$, and a time grid with points t_K $(k = 0, \ldots, K)$ is chosen to define the set of control parameters

$$\boldsymbol{p}_i = \{u_i(t_0), u_i(t_1), \ldots, u_i(t_E)\} \tag{7}$$

representing one component $u_i(t)$ of the control vector $\boldsymbol{u}(t)$. Linear interpolation is used to define control variables at intermediate points. Alternatively, the parameters $\boldsymbol{p}$ can be regarded as coefficients of some mathematical model, which may be either a function of time or a function of current state to serve guidance purposes.[8,18] The optimization parameter vector $\boldsymbol{p}$ may be extended by inclusion of system design variables. Most state-of-the-art optimization algorithms generate sets of improved parameters $\boldsymbol{p}$ approaching the solution along an iterative sequence of search directions that are based on gradient information. The solution thus obtained is optimum only in the sense of the parametrization model chosen, i.e., near-optimum. Also, the result is local in nature, since gradient information is used.

Decomposition Technique

For convenience, let us recall at the outset the typical mission profile of a reusable two-stage space transportation system. This mission comprises an ascent flight segment of the launch vehicle configuration until stage separation, a subsequent hypersonic turn and flyback segment of the booster stage, and the ascent flight leg, orbital mission phase, and re-entry flight of the orbiter vehicle. These primary mission segments may be subdivided into further flight legs as illustrated schematically in the upper part of Fig. 1, which illustrates the general approach of the decomposition.

The underlying idea is to introduce two optimization levels by splitting up the problem into an ordered sequence of self-contained subsegments to be optimized separately on a subproblem level (level 2).[10] A superior optimization cycle (level 1) combines these subsegments while determining optimal initial and boundary conditions $\boldsymbol{l}$, respectively, as well as optimal design parameters $\boldsymbol{\pi}$. The trajectory optimization is affected by the decomposition model and the steering model:

1) The decomposition model defines the different flight segments separated by an input or optimization-specified sequence of events and linking conditions $\boldsymbol{l}_i$; see Fig. 1.

2) The steering model controls the applied aerodynamic and thrust forces for flight-path shaping within each subarc. For parameter model definition, each flight segment is split into time intervals similarly to the method described in Eq. (7). The control models $\boldsymbol{u}(t) = \boldsymbol{T}(\boldsymbol{p}, t)$ employed in this study are mostly of the piecewise linear model type, though different approximations are appropriate in the different flight regimes. Also, different control models may be adopted for the various control variables within a flight segment.[8]

Mathematical Formulation

More formally, the decomposition approach can be stated as follows:

1) Master optimization cycle (level 1). Determine $\boldsymbol{q} = (\boldsymbol{\pi} \cup \boldsymbol{l})$ so as to minimize

$$F(\boldsymbol{p}, \boldsymbol{q}) = F[f_1(\boldsymbol{p}_1, \boldsymbol{q}), \ldots, f_s(\boldsymbol{p}_s, \boldsymbol{q})] \tag{8}$$

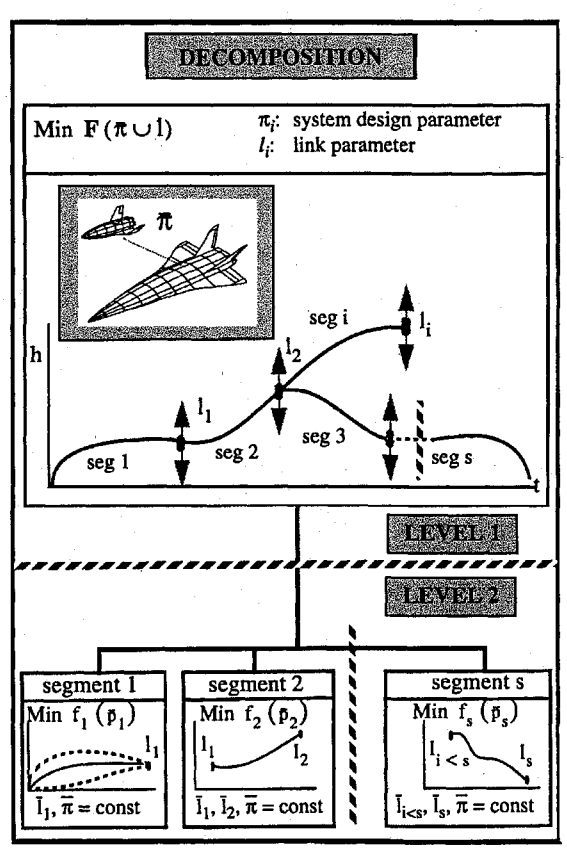

Fig. 1 Methodology of the decomposition approach.

where $\boldsymbol{q} = [\pi_1, \pi_2, \ldots, \pi_u; l_1, l_2, \ldots, l_v]^T$ represents the master parameter vector and $\boldsymbol{p} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_s]^T$ the subproblem controls.

2) Subsegment optimization (level 2). Determine $\boldsymbol{p}_i$ of segment i so as to minimize

$$f_i(\boldsymbol{p}_i, \boldsymbol{q}) \tag{9}$$

where $\boldsymbol{p}_i = [p_{1,i}, p_{2,i}, \ldots, p_{n,i}]^T$ are the flight control parameters of subproblem i and $\boldsymbol{q} = (\pi \cup l)$ the fixed design and linking parameters of the superior optimization level 1, subject to

$$g_{ji}(\boldsymbol{p}_i; \boldsymbol{q}) = 0, \qquad j_i = 1, \ldots, m_{e,i} \tag{10}$$

$$g_{ji}(\boldsymbol{p}_i; \boldsymbol{q}) \geq 0, \qquad j_i = m_{e,i} + 1, \ldots, m_i \tag{11}$$

$$\boldsymbol{p}_i^l \leq \boldsymbol{p}_i \leq \boldsymbol{p}_i^u \tag{12}$$

and

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{p}_i, \boldsymbol{q}) \tag{13}$$

with

$$\boldsymbol{x}(t_i = 0) = \boldsymbol{x}_{0,i} = \mathfrak{S}[\boldsymbol{x}(t_{E,k}, \boldsymbol{p}_k, \boldsymbol{q})] \qquad (k < i) \tag{14}$$

In this formulation all possible mission constraints are distributed among the subsegments that affects them. By that means, the main iteration cycle becomes an unconstrained NLP problem limited only by the parameter upper and lower bounds $\boldsymbol{q}^u$ and $\boldsymbol{q}^l$, respectively. In general, the initial-state condition $\boldsymbol{x}_{0,i}$ of subsegment i is equal to the final-state condition $\boldsymbol{x}(t_{E,i-1})$ of a preceding segment. State discontinuities such as mass staging are included in the vector function $\mathfrak{S}$ of Eq. (14) and will result in discontinuities of $\dot{\boldsymbol{x}}$ according to Eq. (13).

Scaling

Convergence difficulties encountered during flight optimization can often be attributed to improper scaling, which can give rise to ill-conditioned problems. No general rule exists to select the relative weighting factors, and hence scaling is adapted to improved numerical experience with progress in optimization.[19] However, regarding the decision variables $\boldsymbol{p}$, which may vary widely during optimization with adverse effect on convergence properties, a scaling method described below has proved effective for our problem. The scaling employed represents an affine transformation and converts all physical parameter magnitudes into the dimensionless interval $[-1, +1]$ using the rule

$$\boldsymbol{p}_{sc} = D \cdot \boldsymbol{p} + E \tag{15}$$

where D and E are diagonal matrices with elements d_{ii} and e_{ii}:

$$d_{ii} = \frac{2}{p_i^u - p_i^l} \qquad \text{and} \qquad e_{ii} = \frac{p_i^l + p_i^u}{p_i^l - p_i^u} \tag{16}$$

where the lower and upper bounds p^l and p^u are to be specified by the user for each component of the parameter vector $\boldsymbol{p}$. Though this scaling proved satisfactory for the problem considered, further research regarding scaling sensitivities has to be performed.

Numerical Processing

The general procedure of the subproblem solution (level 2) is outlined in Fig. 2. It consists of simulation and optimization steps.

The simulation model defines the vehicle dynamics and the aerodynamic and propulsive properties of a reference vehicle. Also, geophysical and guidance-and-control models are provided. Given the initial conditions of the trajectory subarc considered and trajectory control parameters $\boldsymbol{p}_i$, the equations of motion (13) are numerically integrated to determine the associated flight path and the local performance index f_i. As mentioned before, the control functions are modeled by polygonals and are iteratively improved to obtain the optimum performance while satisfying both the in-flight path constraints and subarc terminal boundary conditions. The boundary conditions may be either specified or determined by the master optimization cycle (level 1). The gradients of the objective functions

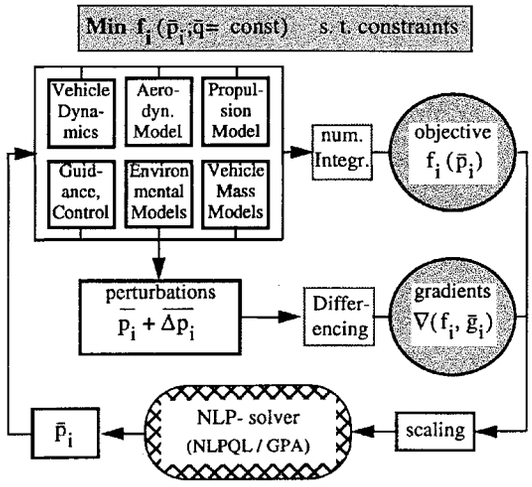

Fig. 2 Numerical procedure on optimization level 2.

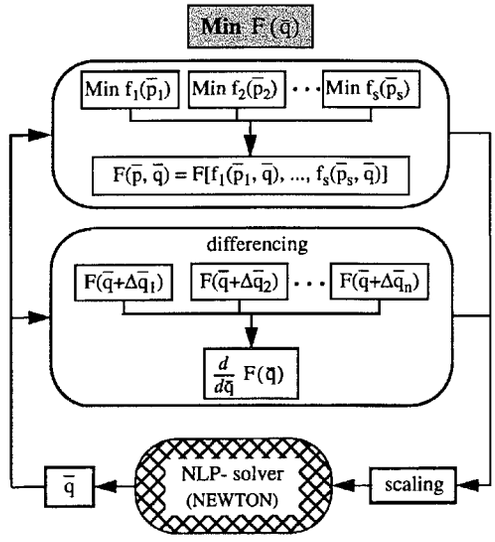

Fig. 3 System optimization level.

and the constraints are computed by numerical difference approximations of perturbed trajectories using standard forward or (optionally) central differencing schemes. During the level 2 optimization steps, fixed system properties $\boldsymbol{q}$ are assumed. As indicated in Fig. 2, subproblem solutions are accomplished by the sequential quadratic programming method NLPQL[20] or, optionally, by an accelerated gradient projection algorithm.[21]

The main optimization cycle (stage 1) is initiated as soon as all subproblems of stage 2 have been solved. It is handled in a manner similar to the subsegment optimization described above. The steps of the procedure are illustrated in Fig. 3. They require successful solution of all subproblems, and hence a failure at subproblem solution will result in premature termination of the overall algorithm.

After evaluation of the main objective F, which may be an arbitrary function of the subsegment objectives f_i, a sensitivity analysis is carried out, employing numerical differencing methods. A new global parameter vector $\boldsymbol{q}$ is determined subsequently by an unrestricted NLP solver, using a quasi-Newton algorithm.

Recall that, at this stage, the major system design parameters to be optimized simultaneously with the flight path are introduced and included in the optimization vector $\boldsymbol{q}$. From Fig. 3, it becomes evident that this problem formulation results in a high computational cost, since every main objective calculation requires the solution of all subproblems. The advantage is that it essentially decouples the trajectory control parameters from the system parameters and therefore helps to avoid convergence problems encountered during combined processing in a standard one-level optimization procedure.

Obviously, a compromise must be found to balance the two conflicting demands of fast subproblem and master-problem solution. Considering the subsegment level, it is desirable to partition the entire mission into many short subarcs to enhance the convergence of the subsegment optimizer. However, every additional subsegment increases the link vector $\boldsymbol{l}$, and therefore introduces additional main-cycle optimization parameters. An increase in main optimization control parameters requires a higher computational effort, and a lower convergence rate of the overall procedure is to be expected.

As previously mentioned, the basic idea of this decomposition method is equivalent to the approach of Petersen et al.,[10] Beltracchi,[11] and Nguyen.[12] It differs in the manner in which the entire mission is partitioned and in which the information on the master independent parameter gradient is calculated. In Ref. 10, the problem is broken into a series of full-rank subproblem targets, determining only subproblem controls to accomplish their restrictions and optimize one master control problem. Beltracchi and Nguyen divide the mission into full optimizable subsegments, but calculate the gradient information for the master optimization cycle with methods of postoptimality or sensitivity analysis,[22,23] which is possible in the case that the partial dependences of the several subsegment cost functions (f_i) from the overall objective (F) are explicitly available. In our case, however, the overall objective F can be an arbitrary user-chosen expression and may differ from the subsegment objectives f_i. Thus, the results of the postoptimality analysis (mission sensitivities $\mathrm{d}f/\mathrm{d}p$) are not sufficient to determine the gradients for the master optimization level 1, which requires the calculation of $\mathrm{d}F/\mathrm{d}p = (\mathrm{d}F/\mathrm{d}f)(\mathrm{d}f/\mathrm{d}p)$. Since the system derivatives $\mathrm{d}F/\mathrm{d}f$ are not available numerically because of their dependence on tabular datasets (e.g., thrust, aerodynamics, vehicle mass model), the main parameter derivatives $\mathrm{d}F/\mathrm{d}p$ are calculated numerically.

Numerical Validation

The coupled trajectory and system optimization procedure outlined above has been validated for the mission of a fully reusable two-stage-to-orbit Sänger-type launch vehicle. The Sänger concept consists of a winged, airbreathing booster vehicle and a rocket-propelled orbiter stage. Its mission is a challenging optimization example because complex interactions between mission requirements and constraints, flight-path selection, engine performance and weight, vehicle design, and flight loads, among other parameters, must be allowed for.

Reference Mission and Vehicle

The mission and vehicle models employed in this study are based on data sets described in detail in Refs. 24 and 25 and hence will only be summarized briefly. The reference mission assumes horizontal takeoff in Europe (Istres, France), an initial flight heading south with a supersonic cruise flight segment (Mach 4.5 at 27.5-km altitude), a subsequent accelerating flight turning east, and a pullup maneuver to staging conditions at Mach 6.8. After stage separation the orbiter vehicle continues its ascent mission to a circular target orbit at 463-km altitude and 28.5-deg inclination, achieved via a 90 × 463 km transfer orbit and an impulsive apogee maneuver. The flyback mission segment of the booster vehicle comprises an unpowered hypersonic turn after staging followed by a gliding phase until cruise flight conditions are achieved, at which time the airbreathing engines are reignited for a powered flyback to the landing site. The trajectory has maximum acceleration limits of 3 g and a dynamic pressure constraint of $q_{\max} = 50$ kPa. In addition, normal load constraints equivalent to 1.5 g are enforced on pullup maneuvers, and 3.0 g during the hypersonic flyback turn of the first stage.

The first stage of the Sänger STS consists of a hypersonic carrier aircraft powered by five turboramjet engines, which burn liquid hydrogen and provide a total thrust of 2.6 MN at takeoff. The second stage, attached to the booster stage until staging, will carry three astronauts and a designated payload of 3 Mg into the target orbit. Its main propulsion consists of an advanced high-pressure LH_2/LOX rocket engine sized to generate thrust for initial orbiter acceleration of 1.25 g. The total launch mass $m_0 = 435$ Mg of the Sänger reference system is split up into $m_{01} = 320$ Mg for the booster stage and $m_{02} = 115$ Mg for the orbiter mass fraction.

Problem Formulation for the Sänger STS

The objective of the performance optimization is to assess the maximum payload capability delivered to the target orbit by a

Determine the optimal ascent trajectory of an airbreathing launch vehicle of Sänger type that delivers a maximum payload to the 463 km target orbit. The staging condition and mass distribution of the two vehicles are unknown and to be determined.

Master Problem:

Maximize: m_{pl} Upper-Stage Payload mass

with the independent variables:

M_{st}	staging Mach number	λ_{st}	longitude at staging
n_{st}	load factor at pull-up	τ_{st}	time interval for pull-up

Subproblem 1

Minimize: $m_{pr,1}$ booster stage ascent propellant

subject to

$M_e = M_{st}$	staging Mach no. (master contr.)
$\lambda_e = \lambda_{st}$	staging longitude (master contr.)
$\delta_e = \delta_{st}$	latitude at staging (=28.5 deg)
$\chi_e = \chi_{st}$	heading at staging (=90.0 deg)

independent variables: *)

$p_{1,1}$:	flight heading after take-off
$p_{2,1}$:	supersonic cruise flight length
$p_{3,1}$-$p_{6,1}$:	bank angle control
$p_{7,1}$:	parameter determines the length of the turn flight

initial state variables

$\overline{x}_{0,1}$ = const.

Subproblem 2

Minimize: $m_{pr,2}$ booster stage flyback propellant

subject to

$n_{max} < n_{mx}$	max. flight acceleration (3.0 g)
$q_{max} < q_{mx}$	max. dynamic pressure (50 kPa)
$\chi_e = \chi_{ap}$	end heading towards landing site

independent variables: †)

$p_{1,2}$-$p_{7,2}$:	angle of attack control
$p_{8,2}$-$p_{13,2}$:	bank angle control
$p_{14,2}$:	parameter determines the length of the turn flight

initial state variables

$\overline{x}_{0,2} = \overline{x}_{E,1}$; except mass: $m_{0,2} = m_{E,1} - m_{0,3}$

Subproblem 3

Minimize: $m_{pr,3}$ orbiter ascent propellant

subject to

$a_{max} < a_{mx}$	max. long. flight acceleration (3.0g)
$v_e = v_{per}$	perigee velocity (v_{per}=7.9595 km/s)
$h_e = h_{per}$	perigee altitude (h_{per} = 90.0 km)
$\gamma_e = \gamma_{per}$	perigee path angle (γ_{per} = 0.0 deg)

independent variables:

$p_{1,3}$-$p_{4,3}$:	angle of attack control

initial state variables

$\overline{x}_{0,3} = \overline{x}_{E,1}$; except mass: $m_{0,3} = f(m_{pr,1}, m_{pr,2})$

* Angle of attack is determined by control laws for subsonic quasi-stationary climb (v=const.), cruise flight condition and constant dynamic pressure 50 kPa

† Terminal cruise flight conditions are accomplished by an angle of attack and throttle setting controller, respectively, activated as soon as the cruise conditions are reached.

Fig. 4 Decomposition formulation for the two-stage-to-orbit Sänger mission.

vehicle configuration of constant mass $m_0 = 435$ Mg at takeoff. The problem formulation includes the determination of the optimal mass fraction of the two stages and the optimal staging conditions, respectively. Vehicle mass predictions are performed by a procedure employing vehicle design scaling according to varying propellant and payload requirements and a vehicle mass estimation using statistical relationships.[9]

The mission characteristics lead to the following natural decomposition introducing the three subsegments: phase 1, ascent of the launch configuration until staging; phase 2, flyback of the booster stage; and phase 3, ascent flight of the orbiter. The re-entry mission of the orbiter vehicle has been investigated in separate analysis and will not be considered in the remainder. The problem statement of the present analyses including the master and subproblem formulation is given in Fig. 4, presenting the optimization parameters and restrictions in detail.

In this example, the overall mission is specified by initial, intermediate, and terminal boundary conditions and is controlled at the main iteration level 1 by four control parameters to be optimized for maximum payload. These parameters represent the Mach number M_{st} and the longitude λ_{st} at staging, which is performed at a latitude of $\delta_{st} = 28.5°$ and a heading of $\chi_{st} = 90.0°$ for inclinations $i = 28.5$ deg. The two remaining parameters are the load factor n_{st} (essentially the aerodynamic lift-to-weight fraction) and time duration τ_{st} of the pullup maneuver prior to staging. The Mach number M_{st} determines the propellant fractions consumed during stages 1 and 2 vehicle operations and thus determines the size and the weight of the vehicles. Also, the longitude λ_{st} affects the geographical flight path of the carrier vehicle and hence the propellants required during ascent and flyback of the first stage. The parameters n_{st} and τ_{st} will optimize the initial flight conditions of the orbiter stage for improved payload fractions. The subproblems defined in Fig. 4 involve angle-of-attack, bank-angle, and engine throttle controls for minimum propellant requirements of the trajectory subarcs considered. A description of the guidance model for the three flight segments is given in Ref. 8.

Propellant consumption is an appropriate criterion function on the subproblem level, since the requirements represent the major design driver for both vehicle stages. However, for the booster stage there is a conflicting situation because the specific impulse (fuel consumption) and thrust-to-weight ratio of airbreathing propulsion candidates exhibit an inverse relationship such that when one engine performance parameter is high the other is low. In our approach this conflicting influence on vehicle weights and hence payload is taken into account in the master iteration cycle that maximizes payload mass.

The following results exemplify the overall convergence properties and omit those of the subproblem solutions, which pose only minor difficulties. The payload (performance index) improvements during the optimization process are depicted in Fig. 5a for two examples of initial control parameters. The corresponding optimization variables are presented in Figs. 5b and 5c. The two sets of initial parameters have been selected to investigate the procedure's capability to converge to the same solution.

The initial controls of example 1 apply to an ascent flight heading south after takeoff and a subsequent hypersonic turn to an eastern directory at stage separation, which occurs at Mach 6.8 and a geographical longitude $\lambda_{st} = 12.5°$E. The optimization process trades off the fuel requirements of the ascent and flyback segments of the carrier vehicle, resulting in a change of the early flight heading to southwest and a subsequent staging location at approximately $\lambda_{st} = 3.5°$E, as shown in Fig. 6.

It is not surprising that only a small change in the staging Mach number is observed, since the present analysis departs from a reference design point of the well-defined Sänger STS. Both examples considered yield the same staging condition M_{st}, λ_{st} and the same payload in the target orbit. However, the optimized parameters of the pullup control show differences (though small), indicating that lower load-factor controls n are compensated by longer pullup maneuver durations τ_{st} and vice versa.

Two remarks are in order regarding Fig. 5a. First, the payload shown is more than two times as high as the specified value of 3 Mg for the Sänger STS. This is because of the larger amount of fuel

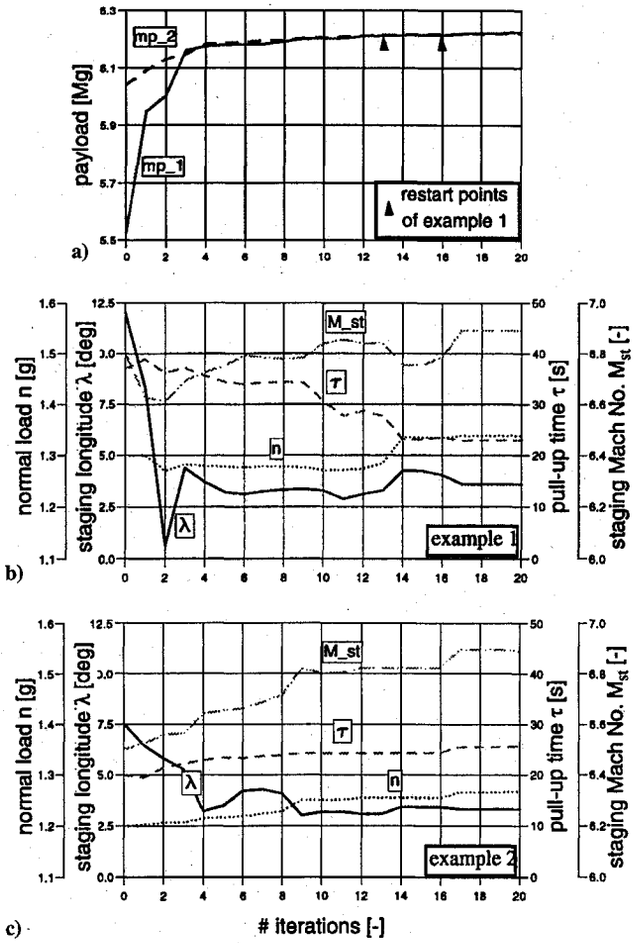

Fig. 5 a) Payload improvements during optimization and b) and c) control parameter development for two initial parameter settings.

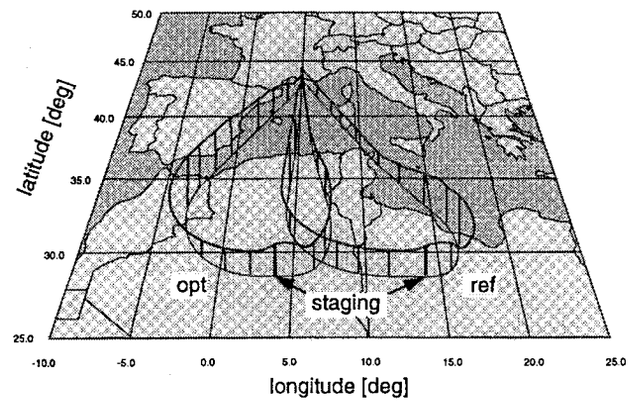

Fig. 6 Carrier aircraft reference and optimized trajectory.

reserves for prolonged cruise flights considered in Ref. 25, which in our case have been traded for increased payload capabilities. Second, some restart points are marked in Fig. 5a. At these stages the optimization process has been interrupted and subsequently restarted to verify vehicle design modifications needed, e.g., to accommodate varying fuel requirements of varied flights. Though this process has been automated in our design and mass estimation procedure, it must be monitored and controlled by the user to avoid unreasonable designs.

The westward shift of the location of stage separation as shown by the ground track and altitude profile in Fig. 6 is a result of significant fuel savings (−36.3%) during the flyback phase of the booster stage, which exceeds the increasing LH_2 fuel requirements (+4.25%) of the ascent flight phase. Note, however, that in practice it may be desirable to retain the stage 2 launch capability at the easterly location

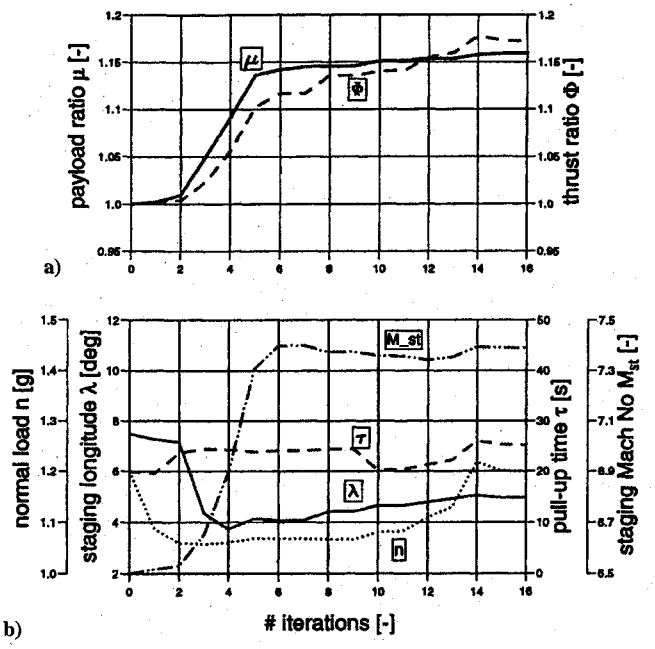

Fig. 7 a) Normalized payload improvements and b) main parameter development during optimization.

for an enlarged launch window, so that this potential for payload improvement may not actually be exploited. Nevertheless, this example demonstrates the capability of the decomposition method to effect significant changes in flight-path shaping.

The next example extends the previous results by inclusion of the installed thrust of the turboramjet propulsion system of the stage 1 vehicle, representing a major system design parameter. Figure 7a depicts the normalized payload μ and the thrust ratio ϕ (defined by the installed thrust divided by the reference thrust at take-off conditions) vs the main iteration numbers. The additional control parameters of the master optimization level are shown in Fig. 7b.

Once again, the algorithm successfully finds an improved solution. An increase of the installed engine thrust by approximately 17% allows higher staging Mach numbers because of better acceleration capabilities of the carrier vehicle and results in an payload increase of roughly 15%. A linear increase of engine weights with thrust has been assumed for vehicle mass estimates.

Parallel Computations

Gradient calculations on both optimization levels require most of the computational time. Therefore, this part has been parallelized in a first step toward fully exploiting the implicit data parallelism of the numerical differencing technique used for gradient approximation. Performance investigations of the parallel decomposition version were conducted on a cluster of six IBM RS/6000-550 workstations called PARIS (PArallel RISc).[26] The message-passing programming model PVM[27] has been used to handle the communication between the various processes on this parallel distributed memory platform. This workstation cluster had been installed with one goal being to test techniques for combining workstations and to obtain experience with software tools such as PVM and the potential improvements of parallel codes in preparation for later use on the more powerful Cray T3D parallel computer, which subsequently became available to us.

Speedup

The speed-up Sp of a parallel program describes the time advantages that parallel evaluation offers as compared to the sequential technique:

$$Sp^{pr} = \frac{t_{\text{elp}}^{(1)}}{t_{\text{elp}}^{(pr)}} \approx \frac{t_{\text{CPU}}^{(1)}}{t_{\text{CPU}}^{(pr)} + t_{\text{COM}}^{(pr)}} \leq pr \tag{17}$$

Ideally, the speed-up factor Sp^{pr} corresponds to the number of computer knots pr, which means that a program is computed pr times faster on a parallel computer employing pr processors than on a sequential one employing one processor. However, these optimum conditions can only be achieved if it is possible to partition the sequential problem into time-balanced parts, and if the communication requirements are vanishing. In the present state the parallel program version consists of a parallel part (gradient evaluation) and a sequential one (line search and search direction evaluation). Thus, the CPU time can be written assuming load balance of the parallel part as $t_{\text{CPU}}^{(pr)} = t_{\text{ser}}^{(pr)} + (t_{\text{par}}/pr)$, and Eq. (17) becomes

$$Sp^{(pr)} = \frac{t_{\text{ser}}^{(1)} + t_{\text{par}}^{(1)}}{t_{\text{ser}}^{(pr)} + (t_{\text{par}}/pr) + t_{\text{COM}}^{(pr)}} \tag{18}$$

The communication time $t_{\text{COM}}^{(pr)}$, which expresses the time effort to establish the communication link between the processors plus the data rate of the communication, could be neglected in our case compared to the CPU-time requirements. Under the assumption that all gradient calculations of an optimization step require the same CPU time (load balance), and that the search direction evaluation time is small compared to the time for line search (i.e., the procedure of finding an optimal step-size scalar during one optimization step) and gradient evaluation, Eq. (18) can be transformed to

$$Sp^{(pr)} = \frac{Ls + n_g n_{\text{par}}}{Ls + n_g n_{\text{pg}}} \tag{19}$$

where Ls and n_g are the number of line-search procedures and number of path integrations per gradient calculation, respectively. The symbol n_{par} denotes the number of optimization parameters, and n_{pg} the number of gradient evaluations per processor.

Performance Comparison

This subsection presents the results of benchmark tests taken from the optimization example employing five main optimization parameters. The numbers of control variables used within the three subproblems are given in Table 1. Ideally, the number of parameters should determine the number of parallel knots that are required for parallel gradient calculation. Since the PARIS cluster consists of only six workstations, more processes have been simulated on a single physical knot in those cases where the parameter number exceeds the number of workstations.

There are two possibilities to measure the computational time on PARIS. The first one determines the time difference that the whole cluster requires for the computation. This approach specifies the real computational time $t_{\text{elp}}^{(pr)}$ of a program on the cluster. In the second case, the elapsed time of the master process (initiating all the other parallel processes) is determined. Provided that more than six processes are initiated and the master process performs the line-search procedure, this time determines the possible speedup on a hypothetical cluster employing the same number of physical knots as processes are initiated.

The speedup (19) for the subproblem optimizations has been evaluated and measured with the setup of Table 1. For the first flight segment, eight processes are required for gradient evaluation, and each process has to calculate two complete flight-path integrations (central difference approximations), whereas in the second flight phase seven processes determine the gradient information (two gradients per process).

Figure 8 illustrates the results of the speed-up measurement (second method) with respect to the number of line-search steps during one optimization iteration step and compares them with the theoretical speedup, which has been evaluated for the three subproblem phases by means of Eq. (19). Since the measured speedups (indicated by markers) are close to the theoretical values (lines) in all

Table 1 Setup for the speed-up evaluations

Flight phase	n_{par}	n_g	n_{pg}
1	8	2	1
2	14	2	2
3	6	2	2

Table 2 Speedup on the PARIS cluster for flight phase 1 optimization

Required number of line-search steps/iteration	t_{elp}, s		
	Parallel	Sequential	Sp
1	18.88	62.99	3.336
2	23.09	67.49	2.932
4	29.34	74.09	2.525
5	33.55	77.86	2.321
7	40.53	85.36	2.106

Table 3 Speedup for the main optimization level

Case	t_{elp}, min		
	Parallel	Sequential	Sp
1: main function	16.0	42.75	2.672
2: main iteration steps	167.0	722.95	4.329

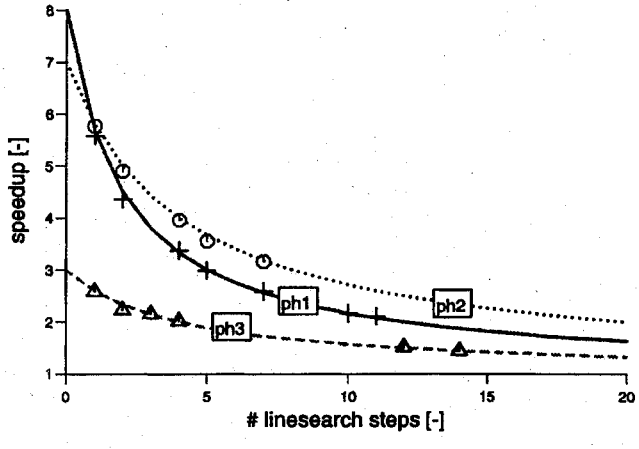

Fig. 8 Theoretical and measured speedup during subsegment optimization.

three flight segments, the previously supposed presumptions are true for Eq. (19) concerning the subsegment optimization. The curves depict the typical trend for the speedup: decreasing from the ideal value to 1 as the portions of sequential computation increase.

Note that these speed-up values are academic values assuming a parallel platform that provides sufficient numbers of knots. Real speed-up values obtained on the PARIS cluster are exemplified in Table 2 for the first flight phase (denoted ph1 in Fig. 8).

Of course, the real speedup is worse than the ideal value based on Eq. (19). The rather large difference, especially for low line-search steps, is caused by the violation of the load-balance assumption. In the example above, four of the six knots are always idle after their work, while the remaining two evaluate the seventh and eighth parameter gradients. However, the difference decrease with increasing number of line-search steps (compared to the results in Fig. 8), because then the sequential part of the program has a dominant influence on the speedup.

Finally, speedup has been measured on PARIS for the master optimization level 1 using the parallel gradient evaluation on both optimization levels. No theoretical speedup could be determined because the characteristics of the main-function computation allow no time prediction, because of the varying numbers of line-search steps during sublevel optimization. Table 3 summarizes these results.

In the first case the computational time has been measured for one main-function evaluation while eight processes have been simulated on PARIS to evaluate the gradient information of the subsegment targets. The real speedup of 2.672 is not as good as expected, since the sequential part of the program is still large. The speedup improves and reaches a value of 4.329 as soon as the main gradient calculation is included in the main optimization cycle. In this case, 40 processes (5 for main parameter gradient evaluation times 8 for sublevel optimization) are initiated on the cluster. This relatively good performance ($Sp_{max} = 6.0$) has its reason in the large number of processes that have been distributed among the six workstations of PARIS. Thus, almost all processors are busy during the main objective gradient evaluation, and idle times are encountered only during the main-optimization line-search steps.

Summary

An optimization method has been developed and applied to the mission of a Sänger-type launch vehicle, considering both the flight path and system design optimization tasks. Previously, this problem showed sensitivity to numerical ill-conditioning, and care must be taken when selecting scaling weights to formulate a well-behaved problem. The procedure discussed employs a two-level decomposition approach that splits the problem into a sequence of flight segments to be optimized separately on a subproblem level. A superior optimization cycle subsequently combines these subsegments while determing optimal initial and boundary conditions as well as optimal design variables. NLP methods are used in both decomposition levels to solve the particular optimization tasks.

The results presented in this article demonstrate the capability of the decomposition method to successfully optimize the entire mission and major vehicle design variables. The approach has emerged as a promising tool in simultaneous system design and trajectory optimization. However, a large computational cost is involved. Therefore, a second effort explored the potential of a parallel-processing workstation cluster for the calculation of the derivatives of functions and constraints required for optimization. Via parallelism, the solution time has been reduced up to four times by employing six processors, indicating substantial time improvements to be expected on computers with a large number of processors. Investigations are currently being conducted to implement the program package on a Cray T3D computer providing 32 processors and to improve the parallel performance by reducing the serial part of the procedure.

Acknowledgment

The support of this work by the Deutsche Forschungsgemeinschaft as part of the Sonderforschungsbereich 259 "High Temperature Problems of Reusable Space Transportation Systems" at Stuttgart University is gratefully acknowledged.

References

[1]Weingartner, S., "SÄNGER—The Reference Concept of the German Hypersonics Technology Program," AIAA Paper 93-5161, Dec. 1993.

[2]Maita, M., "Space Plane Concept Overview," AIAA Paper 93-5162, Dec. 1993.

[3]Lozino-Lozinsky, G., Sokorodelov, V., and Plokhikh, V., "International Reusable Aerospace System MAKS—Present State of the Art and Perspective," AIAA Paper 93-5165, Dec. 1993.

[4]Stanley, D. O., Wilhite, A. W., Englund, W. C., and Laube, J. R., "Comparison of Single-Stage and Two-Stage Airbreathing Launch Vehicles," *Journal of Spacecraft and Rockets*, Vol. 29, No. 5, 1992, pp. 735–740.

[5]Freeman, D. C., Talay, T. A., Stanley, D. O., Lepsch, R. A., and Wilhite, A. W., "Design Options for Advanced Manned Launch Systems," *Journal of Spacecraft and Rockets*, Vol. 32, No. 2, 1995, pp. 241–249.

[6]Braun, R. D., Powell, R. W., Lepsch, R. A., Stanley, D. O., and Kroo, I. M., "Comparison of Two Multidisciplinary Optimization Strategies for Launch-Vehicle Design," *Journal of Spacecraft and Rockets*, Vol. 32, No. 3, 1995, pp. 404–410.

[7]Lepsch, R. A., Jr., Stanley, D. O., and Unal, R., "Dual-Fuel Propulsion in Single-Stage Advanced Manned Launch System Vehicle," *Journal of Spacecraft and Rockets*, Vol. 32, No. 3, 1995, pp. 417–425.

[8]Schöttle, U. M., and Hillesheimer, M., "Performance Optimization of an Airbreathing Launch Vehicle by a Sequential Trajectory Optimization and Vehicle Design Scheme," AIAA Paper 91-2655, Aug. 1991.

[9]Hillesheimer, M., Schöttle, U. M., and Messerschmid, E., "Optimization of Two-Stage Reusable Space Transportation Systems with Rocket and Airbreathing Propulsion Concepts," International Astronautical Federation, Paper 92-0863, Sept. 1992.

[10]Petersen, F. M., Cornick, D. E., and Brauer, G. L., "A Two-Level Trajectory Decomposition Algorithm Featuring Optimal Intermedia Target Selection," *Journal of Spacecraft and Rockets*, Vol. 14, No. 11, 1977, pp. 676–682.

[11]Beltracchi, T. J., "Decomposition Approach to Solving the All-Up Trajectory Optimization Problem," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 3, 1992, pp. 707–716.

[12]Nguyen, H. N., "FONSIZE: A Trajectory Optimization and Vehicle Sizing Program," AIAA Paper 93-1100, Feb. 1993.

[13]Betts, J. T., and Huffman, W. P., "Trajectory Optimization on a Parallel Processor," *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, 1991, pp. 431–439.

[14]Park, K., and Psiaki, M. L., "Design and Test of a Parallel Trajectory Optimization Algorithm," AIAA Paper 94-3636, Aug. 1994.

[15]Enright, P. J., and Conway, B. A., "Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002.

[16]Betts, J. T., and Cramer, E. J., "Application of Direct Transcription to Commercial Aircraft Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 1, 1995, pp. 151–159.

[17]Braur, G. L., Cornick, D. E., and Stevenson, R., "Capabilities and Applications of the Programme to Optimize Simulated Trajectories," NASA CR-2770, Feb. 1977.

[18]Schöttle, U. M., Grallert, H., and Hewitt, F. A., "Advanced Air-breathing Propulsion Concepts for Winged Launch Vehicles," International Astronautical Federation, Paper 88-248, Oct. 1988; also *Acta Astronautica*, Vol. 20, 1989, pp. 117–129.

[19]Hallman, W., "Sensitivity Analysis for Trajectory Optimization Problems," AIAA Paper 90-0471, Jan. 1990.

[20]Schittkowski, K., "NLPQL—A Fortran Subroutine for Solving Constrained Nonlinear Programing Problems," *Annals of Operations Research*, Vol. 5, 1985–1986, pp. 485–500.

[21]Schöttle, U. M., "Flight and Propulsion Optimization of Airbreathing Launch Systems," Ph.D. Thesis, Stuttgart Univ., Stuttgart, Germany, Nov. 1988 (in German).

[22]Fiacco, A. V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, edited by R. Bellman, Vol. 165, Mathematics in Science and Engineering, Academic, New York, 1983.

[23]Braun, R. D., and Kroo, I. M., "Post-Optimality Analysis in Aerospace Vehicle Design," AIAA Paper 93-3932, Aug. 1993.

[24]Kuczera, H., Hauck, H., and Sacher, P., "The German Hypersonics Technology Programme Status 1993 and Perspectives," AIAA Paper 93-5159, Dec. 1993.

[25]Anon., "SÄNGER Concept Study, Phase Ib from May 1, 1991–Jan. 31, 1993," Daimler-Benz Aerospace AG, Orbital- und Hydrotechnologie Bremen System GmbH, 1993 (in German); Final Rept. UR-V-193(93), Jan. 1993.

[26]Resch, M., Geiger, A., and Zikeli, J., "Message-Passing Systems on Workstation Clusters and Parallel Computers—The Impact of Software and Network Architectures on Applications," *High-Performance Computing and Networking*, edited by W. Gentzsch and U. Harms, Lecture Notes in Computer Science, Springer–Verlag, Berlin, 1994.

[27]Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., and Sunderam, V., "PVM 3—User's Guide and Reference Manual," Oak Ridge National Lab., ORNL/TM-12187, Oak Ridge, TN, May 1993.

J. A. Martin
Associate Editor