INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or “target” for pages apparently lacking from the document photographed is “Missing Page(s)”. If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of “sectioning” the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.
Wang, Tong

DUALITY PROPERTIES AND SEQUENTIAL GRADIENT-RESTORATION ALGORITHMS FOR OPTIMAL CONTROL PROBLEMS

Rice University

University
Microfilms
International
300 N. Zeeb Road, Ann Arbor, MI 48106

Ph.D. 1985
PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark √.

1. Glossy photographs or pages ______
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy ______
5. Pages with black marks, not original copy ______
6. Print shows through as there is text on both sides of page ______
7. Indistinct, broken or small print on several pages √
8. Print exceeds margin requirements ______
9. Tightly bound copy with print lost in spine ______
10. Computer printout pages with indistinct print ______
11. Page(s) ________ lacking when material received, and not available from school or author.
12. Page(s) ________ seem to be missing in numbering only as text follows.
13. Two pages numbered ______. Text follows.
14. Curling and wrinkled pages ______
15. Dissertation contains pages with print at a slant, filmed as received ______
16. Other______________________________________________________

______________________________________________________

University
Microfilms
International
RICE UNIVERSITY

DUALITY PROPERTIES AND SEQUENTIAL
GRADIENT-RESTORATION ALGORITHMS FOR OPTIMAL CONTROL PROBLEMS

by

TONG WANG

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

APPROVED, THESIS COMMITTEE:

A. Miele, Professor
Mechanical Engineering Department
Chairman

Frederic A. Wierum, Professor
Mechanical Engineering Department

Derek C. Dyson, Professor
Chemical Engineering Department

Houston, Texas
April 1985
ABSTRACT
Duality Properties and Sequential
Gradient-Restoration Algorithms for Optimal Control Problems
by
Tong Wang

This thesis considers duality properties and their application to the sequential gradient-restoration algorithms (SGRA) for optimal control problems. Two problems are studied: (P1) the basic problem and (P2) the general problem. In Problem (P1), the minimization of a functional is considered, subject to differential constraints and final constraints, the initial state being given; in Problem (P2), the minimization of a functional is considered, subject to differential constraints, nondifferential constraints, initial constraints, and final constraints. Depending on whether the primal formulation is used or the dual formulation is used, one obtains a primal sequential gradient-restoration algorithm (PSGRA) and a dual sequential gradient-restoration algorithm (DSGRA).

With particular reference to Problem (P2), it is found convenient to split the control vector into an independent control vector and a dependent control vector, the latter having the same dimension as the nondifferential constraint vector. This modification enhances the computational efficiency of both the primal formulation and the dual formulation.

The basic property of the dual formulation is that the Lagrange multipliers associated with the gradient phase and the restoration phase of SGRA minimize a special functional, quadratic in multipliers, subject to the multiplier differential equations and boundary conditions, for given state, control, and parameter. This duality property yields considerable computational benefits in that
the auxiliary optimal control problems associated with the gradient phase and the restoration phase of SGRA can be reduced to mathematical programming problems involving a finite number of parameters as unknowns.

Several numerical examples are solved using both the primal formulation and the dual formulation.

Key Words. Numerical methods, mathematical programming, optimal control, primal formulation, dual formulation, sequential gradient-restoration algorithms.
ACKNOWLEDGEMENTS

The author would like to express his sincere gratitude to Dr. Angelo Miele for introducing him to the field of optimal control theory and its application and for his advice, guidance, and encouragement throughout the present work. The author would like to express his appreciation to Ms. P. Thomas for the excellent typing of the thesis. This research was supported in part by the Jet Propulsion Laboratory.

DEDICATION

The author would like to dedicate this work to his wife Qi Yueying, who has always been a source of encouragement and understanding.
TABLE OF CONTENTS

1. Introduction  
   Page 1
2. Basic Problem, Optimality Conditions  
   Page 4
3. Basic Problem, Gradient Phase, Primal Formulation  
   Page 7
4. Basic Problem, Gradient Phase, Dual Formulation  
   Page 10
5. Basic Problem, Gradient Phase, Computational Implications  
   Page 13
6. Basic Problem, Restoration Phase, Primal Formulation  
   Page 17
7. Basic Problem, Restoration Phase, Dual Formulation  
   Page 20
8. Basic Problem, Restoration Phase, Computational Implications  
   Page 23
9. Basic Problem, Dual Sequential Gradient-Restoration Algorithm  
   Page 26
10. General Problem, Optimality Conditions  
    Page 33
11. General Problem, Gradient Phase, Primal Formulation  
    Page 38
12. General Problem, Gradient Phase, Dual Formulation  
    Page 42
13. General Problem, Gradient Phase, Computational Implications  
    Page 46
14. General Problem, Restoration Phase, Primal Formulation  
    Page 50
15. General Problem, Restoration Phase, Dual Formulation  
    Page 54
16. General Problem, Restoration Phase, Computational Implications  
    Page 58
17. General Problem, Dual Sequential Gradient-Restoration Algorithm  
    Page 61
18. Comments  
    Page 68
19. Experimental Conditions  
    Page 69
20. Numerical Examples  
    Page 71
21. Discussion and Conclusions  
    Tables  
    References  
    Page 83
    Page 85
    Page 99
1. **Introduction**

This thesis considers duality properties and their application to sequential gradient-restoration algorithms (SGRA) for optimal control problems. Two problems are studied: (P1) the basic problem and (P2) the general problem. In Problem (P1), the minimization of a functional is considered, subject to differential constraints and final constraints, the initial state being given; in Problem (P2), the minimization of a functional is considered, subject to differential constraints, nondifferential constraints, initial constraints, and final constraints.

Sequential-gradient restoration algorithms involve a sequence of two-phase cycles, each cycle including a gradient phase and a restoration phase. In a complete gradient-restoration cycle, the value of the functional is decreased, while the constraints are satisfied to a predetermined accuracy; in the gradient phase, the value of the augmented functional is decreased, while avoiding excessive constraint violation; in the restoration phase, the constraint error is decreased, while avoiding excessive change in the value of the functional.

Depending on whether the primal formulation is used or the dual formulation is used, one obtains a primal sequential gradient-restoration algorithm (PSGRA) and a dual sequential gradient-restoration algorithm (DSGRA). This statement applies to both Problem (P1) and Problem (P2).

Problem (P2) is considerably more complicated than Problem (P1), because of the presence of the nondifferential constraints to be satisfied everywhere along the interval of integration. With particular reference to Problem (P2), it is found convenient to split the control vector into an independent control vector and a dependent control vector, the latter having the same dimension as the nondifferential constraint vector. This modification enhances the computational efficiency of both the primal formulation and the dual formulation.
The basic property of the dual formulation is that the Lagrange multipliers associated with the gradient phase and the restoration phase of SGRA minimize a special functional, quadratic in multipliers, subject to the multiplier differential equations and boundary conditions, for given state, control, and parameter. This duality property yields considerable computational benefits in that the auxiliary optimal control problems associated with the gradient phase and the restoration phase of SGRA can be reduced to mathematical programming problems involving a finite number of parameters as unknowns.

Previous Research. Previous research in the area of problems discussed in this thesis can be found in Refs. 1-27. Specifically, Refs. 1-3 discuss sequential gradient-restoration algorithms for optimal control problems, and Refs. 4-7 discuss combined gradient-restoration algorithms. Modified quasilinearization algorithms for optimal control problems are treated in Refs. 18-20. A basic ingredient of the algorithms presented in Refs. 1-7 and 18-20 is the method of particular solutions for solving linear, two-point boundary-value problems on a digital computer; this is discussed in Refs. 15-17.

Concerning duality properties for mathematical programming problems and optimal control problems, see Refs. 21-27. Algorithmic duality for optimal control problems is discussed in Refs. 8-14. The present thesis is an outgrowth of previous work by Miele and Wang, discussed in Refs. 9-14.

Notations. Vector-matrix notation is used for conciseness. All vectors are column vectors.

Let \( t \) denote the independent variable, and let \( x(t), u(t), v(t), \pi \) denote the dependent variables. The time \( t \) is a scalar; the state \( x(t) \) is an \( n \)-vector; the control \( u(t) \) is an \( m \)-vector; the control \( v(t) \) is a \( c \)-vector; and the parameter \( \pi \) is a \( p \)-vector.
Let \( f(x,u,v,\pi,t) \) denote a scalar function of the arguments \( x,u,v,\pi,t \). The symbol \( f_x \) denotes the \( n \)-vector function whose components are the partial derivatives of the scalar function \( f \) with respect to the components of the vector \( x \). Analogous definitions hold for the symbols \( f_u, f_v, f_\pi \).

Similar definitions are employed for the partial derivatives \( h_x, h_\pi, g_x, g_\pi \) of the scalar functions \( h(x,\pi), g(x,\pi) \).

Let \( S(x,u,v,\pi,t) \) denote a \( c \)-vector function of the arguments \( x,u,v,\pi,t \). The symbol \( S_x \) denotes the \( nxn \) matrix function whose elements are the partial derivatives of the components of the vector function \( S \) with respect to the components of the vector \( x \). Analogous definitions hold for the symbols \( S_u, S_v, S_\pi \).

Similar definitions are employed for the partial derivatives \( \phi_x, \phi_u, \phi_v, \phi_\pi, \omega_x, \omega_\pi, \psi_x, \psi_\pi \) of the vector functions \( \phi(x,u,v,\pi,t), \omega(x,\pi), \psi(x,\pi) \).

The dot sign denotes derivative with respect to the time, that is, \( \dot{x} = \frac{dx}{dt} \). The symbol \( T \) denotes transposition of vector or matrix. The subscript \( 0 \) denotes the initial point, and the subscript \( 1 \) denotes the final point.

The symbol \( N(y) = y^T y \) denotes the quadratic norm of a vector \( y \).
2. **Basic Problem, Optimality Conditions**

**Problem (P1).** We consider the problem of minimizing the functional

\[ I = \int_{0}^{1} f(x,u,\pi,t) \, dt + [g(x,\pi)]_1, \quad (1) \]

with respect to the n-vector state \( x(t) \), the m-vector control \( u(t) \), and the p-vector parameter \( \pi \) which satisfy the constraints

\[ \dot{x} + \phi(x,u,\pi,t) = 0, \quad 0 \leq t \leq 1, \quad (2a) \]

\[ (x)_{0} = \text{given}, \quad (2b) \]

\[ [\psi(x,\pi)]_1 = 0. \quad (2c) \]

In the above equations, \( f \) is a scalar; \( g \) is a scalar; \( \phi \) is a n-vector; and \( \psi \) is a b-vector, \( b \leq n \). We assume that the first and second derivatives of the functions \( f, g, \phi, \psi \) with respect to the vectors \( x,u,\pi \) exist and are continuous. We also assume that the nxb matrix \( \psi_{\dot{x}} \) has rank \( b \) at final point and that the constrained minimum exists.

From calculus of variations, it is known that Problem (P1) is of the Bolza type. It can be recast as that of minimizing the augmented functional

\[ J = I + L, \quad (3) \]

subject to (2), where \( L \) denotes the Lagrangian functional

\[ L = \int_{0}^{1} \lambda^T (\dot{x} + \phi) \, dt + (\mu^T \psi)_1. \quad (4) \]

In Eq. (4), \( \lambda(t) \) denotes an n-vector Lagrange multiplier and \( \mu \) denotes a b-vector Lagrange multiplier.

For Problem (P1), let the Hamiltonian \( H \) and the augmented function \( G \) be defined by
\[ H = f + \lambda^T \phi, \quad (5a) \]
\[ G = g + \mu^T \psi. \quad (5b) \]

Then, the first-order optimality conditions for Problem (P1) take the form

\[ \dot{\lambda} - H_x = 0, \quad 0 \leq t \leq 1, \quad (6a) \]
\[ H_u = 0, \quad 0 \leq t \leq 1, \quad (6b) \]
\[ \int_0^1 H_t \, dt + (G_n)_1 = 0, \quad (6c) \]
\[ (\lambda + G_x)_1 = 0. \quad (6d) \]

On account of Eqs. (5), the explicit form of Eqs. (6) is the following:

\[ \dot{\lambda} - f_x - \phi_x \lambda = 0, \quad 0 \leq t \leq 1, \quad (7a) \]
\[ f_u + \phi_u \lambda = 0, \quad 0 \leq t \leq 1, \quad (7b) \]
\[ \int_0^1 (f_t + \phi_t \lambda) \, dt + (g_n + \psi_n \mu)_1 = 0, \quad (7c) \]
\[ (\lambda + g_x + \psi_x \mu)_1 = 0. \quad (7d) \]

Summarizing, we seek the functions \( x(t), u(t), \pi \) and the multipliers \( \lambda(t), \mu \) such that the feasibility equations (2) and the optimality conditions (7) are satisfied.

**Performance Indexes.** The form of Eqs. (2) and (7) suggests that the following scalar performance indexes are useful in computational work:

\[ P = \int_0^1 N(\dot{x} + \phi) \, dt + N(\psi)_1, \quad (8a) \]

In Eq. (8a), it is implicitly assumed that the initial condition (2b) is satisfied.
\[ Q = \int_0^1 N(\dot{\lambda} - f_x - \phi_x \lambda)dt + \int_0^1 N(f_u + \phi_u \lambda)dt \]
\[ + N\left[ \int_0^1 (f_n + \phi_n \lambda)dt + \left( g_n + \psi_n \mu \right)_1 \right] + N(\lambda + g_x + \psi_x \mu)_1, \]  
(8b)

\[ R = P + Q. \]  
(8c)

Here, \( P \) denotes the error in the constraints, \( Q \) the error in the optimality conditions, and \( R \) the total error in the system. Therefore, numerical convergence can be characterized by the relations

\[ P \leq \varepsilon_1, \]  
(9a)

\[ Q \leq \varepsilon_2, \]  
(9b)

or by the relation

\[ R \leq \varepsilon_3, \]  
(10)

where \( \varepsilon_1, \varepsilon_2, \varepsilon_3 \) are preselected, small, positive numbers.
3. **Basic Problem, Gradient Phase, Primal Formulation**

The gradient phase of the sequential gradient-restoration algorithm involves a single iteration and is designed to decrease the augmented functional, while avoiding excessive constraint violation. The gradient iteration is started whenever Ineq. (9a) is satisfied.

Let \( x(t), u(t), \pi \) denote the nominal functions. Let \( \tilde{x}(t), \tilde{u}(t), \tilde{\pi} \) denote the varied functions. Let \( \Delta x(t), \Delta u(t), \Delta \pi \) denote the displacements leading from the nominal functions to the varied functions. By definition, the following relations hold:

\[
\tilde{x}(t) = x(t) + \Delta x(t), \quad (11a)
\]

\[
\tilde{u}(t) = u(t) + \Delta u(t), \quad (11b)
\]

\[
\tilde{\pi} = \pi + \Delta \pi. \quad (11c)
\]

The displacements \( \Delta x(t), \Delta u(t), \Delta \pi \) are computed by solving the following auxiliary minimization problem.

**Problem (GP).** Minimize the first variation of the functional (1), with respect to the vectors \( \Delta x(t), \Delta u(t), \Delta \pi \) which satisfy the linearized form of the constraints (2) plus a quadratic isoperimetric constraint imposed on the vectors \( \Delta u(t), \Delta \pi \). Therefore, we minimize the functional

\[
I_p = \int_0^1 (f_x^T \Delta x + f_u^T \Delta u + f_{\pi}^T \Delta \pi) dt + (g_x^T \Delta x + g_u^T \Delta u + g_{\pi}^T \Delta \pi)_1, \quad (12)
\]

with respect to the vectors \( \Delta x(t), \Delta u(t), \Delta \pi \) which satisfy the constraints

\[
\Delta x + \phi_x^T \Delta x + \phi_u^T \Delta u + \phi_{\pi}^T \Delta \pi = 0, \quad 0 \leq t \leq 1, \quad (13a)
\]

\[
(\Delta x)_0 = 0, \quad (13b)
\]
\[(\psi_\lambda^T\Delta x + \psi_\pi^T\Delta \pi)_1 = 0, \quad \text{(13c)}\]

and
\[
\int_0^1 \Delta u^T \Delta u \, dt + \Delta \pi^T \Delta \pi - \text{const} = 0. \quad \text{(14)}
\]

From calculus of variations, it is known that Problem (GP) is of the Bolza type. It can be recast as that of minimizing the augmented functional
\[
J_p = I_p + L_p, \quad \text{(15)}
\]

subject to (13)-(14), where \(L_p\) denotes the Lagrangian functional
\[
L_p = \int_0^1 \lambda^T (\dot{\Delta} x + \phi_\lambda^T \Delta x + \phi_u^T \Delta u + \phi_\pi^T \Delta \pi) \, dt + \mu^T (\psi_\lambda^T \Delta x + \psi_\pi^T \Delta \pi)_1
+ (1/2\alpha) \int_0^1 \Delta u^T \Delta u \, dt + \Delta \pi^T \Delta \pi - \text{const}. \quad \text{(16)}
\]

In Eq. (16), \(\lambda(t)\) denotes an \(n\)-vector Lagrange multiplier, \(\mu\) a \(b\)-vector Lagrange multiplier, and \(1/2\alpha\) a scalar Lagrange multiplier.

For Problem (GP), let the Hamiltonian \(H\) and the augmented functions \(K, G\) be defined by
\[
H = f_\lambda^T \Delta x + f_u^T \Delta u + f_\pi^T \Delta \pi + \lambda^T (\phi_\lambda^T \Delta x + \phi_u^T \Delta u + \phi_\pi^T \Delta \pi) + (1/2\alpha) \Delta u^T \Delta u, \quad \text{(17a)}
\]
\[
K = (1/2\alpha) \Delta \pi^T \Delta \pi, \quad \text{(17b)}
\]
\[
G = g_\lambda^T \Delta x + g_\pi^T \Delta \pi + \mu^T (\psi_\lambda^T \Delta x + \psi_\pi^T \Delta \pi). \quad \text{(17c)}
\]

Then, the first-order optimality conditions for Problem (GP) take the form
\[
\dot{\lambda} - H_{\Delta x} = 0, \quad 0 \leq t \leq 1, \quad \text{(18a)}
\]
\[
H_{\Delta u} = 0, \quad 0 \leq t \leq 1, \quad \text{(18b)}
\]
\[ \int_0^1 H_{\Delta t} \, dt + K_{\Delta t} + (G_{\Delta t})_1 = 0, \]  
(18c)

\[ (\lambda + G_{\Delta x})_1 = 0. \]  
(18d)

On account of Eqs. (17), the explicit form of Eqs. (18) is the following:

\[ \lambda - f_x - \phi_x \lambda = 0, \quad 0 \leq t \leq 1, \]  
(19a)

\[ f_u + \phi_u \lambda + \Delta u / \alpha = 0, \quad 0 \leq t \leq 1, \]  
(19b)

\[ \int_0^1 (f_t + \phi_t \lambda) \, dt + (q_t + \psi_t \mu)_1 + \Delta t / \alpha = 0, \]  
(19c)

\[ (\lambda + g_x + \psi_x \mu)_1 = 0. \]  
(19d)

Summarizing, we seek the functions \( \Delta x(t), \Delta u(t), \Delta t \) and the multipliers \( \lambda(t), \mu, 1/2 \alpha \) such that the feasibility equations (13)-(14) and the optimality conditions (19) are satisfied.
4. Basic Problem, Gradient Phase, Dual Formulation

Let $y(t)$, $z$ denote the vectors defined by

$$f_u + \phi_u \lambda + y = 0, \quad (20a)$$

$$\int_0^1 (f_\pi + \phi_\pi \lambda) dt + (g_\pi + \psi_\pi u)_1 + z = 0. \quad (20b)$$

It is interesting to note that the vectors $\lambda(t)$, $\mu$ and $y(t)$, $z$ can also be obtained by solving the following auxiliary minimization problem.

**Problem (GD).** Minimize the functional

$$I_D = (1/2) \int_0^1 y^T y dt + z^T z, \quad (21)$$

with respect to the vectors $\lambda(t)$, $\mu$ and $y(t)$, $z$ which satisfy the constraints

$$\lambda - f_x - \phi_x \lambda = 0, \quad 0 \leq t \leq 1, \quad (22a)$$

$$f_u + \phi_u \lambda + y = 0, \quad 0 \leq t \leq 1, \quad (22b)$$

$$\int_0^1 (f_\pi + \phi_\pi \lambda) dt + (g_\pi + \psi_\pi u)_1 + z = 0, \quad (22c)$$

$$(\lambda + g_x + \psi_x u)_1 = 0. \quad (22d)$$

From calculus of variations, it is known that Problem (GD) is of the Bolza type. It can be recast as that of minimizing the augmented functional

$$J_D = I_D + L_D, \quad (23)$$

subject to (22), where $L_D$ denotes the Lagrangian functional

$$L_D = \int_0^1 \left[ x^T (\lambda - f_x - \phi_x \lambda) - y^T (f_u + \phi_u \lambda + y) - z^T (f_\pi + \phi_\pi \lambda) dt \right.$$  

$$- z^T (g_\pi + \psi_\pi u)_1 - z^T z - \psi_x (\lambda + g_x + \psi_x u)_1. \quad (24)$$
In Eq. (24), $\lambda_*(t)$ denotes an n-vector Lagrange multiplier, $y_*(t)$ an m-vector Lagrange multiplier, $z_*$ a p-vector Lagrange multiplier, and $\nu_*$ an n-vector Lagrange multiplier.

For Problem (GD), let the Hamiltonian $H$ and the augmented functions $K, G$ be defined by

$$H = (1/2)y^T y - \lambda^T_x (f_x + \phi_x \lambda) - y^T_u (f_u + \phi_u \lambda + y) - z^T_\pi (f_\pi + \phi_\pi \lambda), \quad (25a)$$

$$K = (1/2)z^T z - z^T_\pi z, \quad (25b)$$

$$G = -z^T_\pi (g_\pi + \psi_\pi \mu) - \nu^T_\pi (\lambda + g_x + \psi_x \mu). \quad (25c)$$

Then, the first-order optimality conditions for Problem (GD) take the form

$$\dot{\lambda}_* - H_{\lambda} = 0, \quad 0 \leq t \leq 1, \quad (26a)$$

$$H_y = 0, \quad 0 \leq t \leq 1, \quad (26b)$$

$$(G_{\mu})_1 = 0, \quad (26c)$$

$$K_z = 0, \quad (26d)$$

$$(\lambda_*)_0 = 0, \quad (26e)$$

$$(\lambda_* + G_{\lambda})_1 = 0. \quad (26f)$$

On account of Eqs. (25), the explicit form of Eqs. (26) is the following:

$$\dot{\lambda}_* + \phi_x^T \lambda_* + \phi_u^T y_* + \phi_\pi^T z_* = 0, \quad 0 \leq t \leq 1, \quad (27a)$$

$$y - y_* = 0, \quad 0 \leq t \leq 1, \quad (27b)$$

$$(\psi_x^T \nu_* + \psi_\pi^T z_*)_1 = 0. \quad (27c)$$
\[ z - z_\# = 0, \]  
\[ (\lambda_\#)_0 = 0, \] \hspace{1cm} (27d) \hspace{1cm} (27e) \hspace{1cm} (27f) 
\[ (\lambda_\# - \nu_\#)_1 = 0. \]

Let the following substitutions be employed:

\[ \lambda_\# = \Delta x(t)/\alpha, \] \hspace{1cm} 0 \leq t \leq 1, \hspace{1cm} (28a) 
\[ y_\# = \Delta u(t)/\alpha, \] \hspace{1cm} 0 \leq t \leq 1, \hspace{1cm} (28b) 
\[ z_\# = \Delta T/\alpha, \] \hspace{1cm} (28c) 
\[ \nu_\# = \Delta x(1)/\alpha. \] \hspace{1cm} (28d)

Then, one can readily verify that Eqs. (27) and (22) of Problem (GD) reduce to Eqs. (13) and (19) of Problem (GP). Clearly, after the transformation (28) is applied, the solution of Problem (GD) yields the solution of Problem (GP) and vice versa. This means that the multipliers \( \lambda(t), \mu \) associated with the gradient phase of SGRA are endowed with a duality property: They also minimize the quadratic functional (21), subject to (22), for given state, control, and parameter.
5. Basic Problem, Gradient Phase, Computational Implications

In this section, we exploit the previously established duality property and show that the execution of a gradient iteration can be reduced to solving a mathematical programming problem involving a finite number of parameters as unknowns. Hence, the algorithmic efficiency of the gradient phase of SGRA can be enhanced.

First, we consider Eqs. (22a) and (22d). We observe that, if \( \mu \) is assigned, \( \lambda(1) \) can be computed with (22d) and \( \lambda(t) \) can be computed by backward integration of (22a). Next, we execute \( b+1 \) backward integrations, using Eqs. (22a) and (22d) in combination with the following choices for the multiplier \( \mu \):

\[
\begin{align*}
\mu_1 &= \delta_1, \\
\mu_2 &= \delta_2, \\
&\vdots \\
\mu_b &= \delta_b, \\
\mu_{b+1} &= 0.
\end{align*}
\]  

(29a)

(29b)

In (29a), \( \delta_1, \delta_2, \ldots, \delta_b \) denote the vectors corresponding to the columns of the identity matrix of order \( b \); in (29b), \( 0 \) denotes the null vector of dimension \( b \).

Let \( \lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t), \lambda_{b+1}(t) \) denote the particular solutions of Eqs. (22a) and (22d), corresponding to the choices (29) for the multiplier \( \mu \). Let \( \tilde{\mu} \) denote the bx(b+1) matrix

\[
\tilde{\mu} = [\mu_1, \mu_2, \ldots, \mu_b, \mu_{b+1}];
\]

(30a)

Let \( \tilde{\lambda}(t) \) denote the nx(b+1) matrix

\[
\tilde{\lambda}(t) = [\lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t), \lambda_{b+1}(t)];
\]

(30b)

---

\(^2\)Clearly, \( \tilde{\mu} = [1,0] \), where \( I \) denotes the identity matrix of order \( b \).
and let \( k \) denote the \((b+1)\)-vector

\[
k = [k_1, k_2, \ldots, k_b, k_{b+1}]^T. \tag{30c}
\]

If the method of particular solution is employed (Refs. 15-17), the general solution of Eqs. (22a) and (22d) can be written in the form

\[
\mu = \tilde{\mu}k, \tag{31a}
\]

\[
\lambda(t) = \tilde{\lambda}(t)k, \tag{31b}
\]

with the following understanding: the components of the vector \( k \) must satisfy the normalization condition

\[
U^Tk = 1, \tag{32}
\]

where

\[
U = [1, 1, \ldots, 1, 1]^T \tag{33}
\]

denotes the \((b+1)\)-vector whose components are all equal to one.

Next, we combine Eqs. (22b), (22c) with Eqs. (31) and obtain the relations

\[
y = -f_{\mu} - \phi_{\mu} \tilde{\lambda}k, \quad 0 \leq t \leq 1, \tag{34a}
\]

\[
z = -\left[ \int_0^t f_{\mu} dt + (g_{\mu})_1 \right] - \left[ \int_0^t \phi_{\mu} \tilde{\lambda} dt + (\psi_{\mu})_1 \right] \tilde{\mu}k. \tag{34b}
\]

These relations show that \( y(t), z \) depend only on the parameter \( k \).

Finally, upon combining (21) and (34), we obtain the following quadratic function of \( k \):

\[
I_D = (1/2)k^T M k + D^T k + (1/2)H, \tag{35}
\]

where
\[ U^T k - 1 = 0. \]  \hfill (36)

Here, the matrix \( M \), the vector \( D \), and the scalar \( H \) are known. They are defined by

\[ M = \int_0^1 (\phi_u \lambda) (\phi_u \lambda) dt + \int_0^1 \phi_{\pi} \lambda dt + (\psi_{\pi})_1 \lambda \mu dt, \]  \hfill (37a)
\[ D = \left[ \int_0^1 \phi_{\pi} \lambda dt + (\psi_{\pi})_1 \lambda \right] \left[ \int_0^1 f_{\pi} dt + (g_{\pi})_1 \right] + \int_0^1 (\phi_u \lambda) f_u dt, \]  \hfill (37b)
\[ H = \left[ \int_0^1 f_{\pi} dt + (g_{\pi})_1 \right] \left[ \int_0^1 f_{\pi} dt + (g_{\pi})_1 \right] + \int_0^1 f_u f_u dt. \]  \hfill (37c)

Because of the duality property, the parameter \( k \) can be obtained by minimizing (35), subject to (36). Clearly, this auxiliary minimization problem is a mathematical programming problem.

Let \( \lambda \) denote a scalar Lagrange multiplier associated with the constraint (36). Let \( G_\lambda \) denote the augmented function

\[ G_\lambda = (1/2) k^T M k + D^T k + (1/2) H + \lambda (U^T k - 1). \]  \hfill (38)

With this understanding, the first-order optimality condition of the auxiliary minimization problem takes the form

\[ G_{\lambda k} = 0. \]  \hfill (39)

Hence, the values of \( k, \lambda \) are determined by solving the following linear algebraic system:

\[ M k + U \lambda + D = 0, \]  \hfill (40a)
\[ U^T k - 1 = 0, \]  \hfill (40b)
whose dimension is \( b+2 \). Once \( k, \beta \) are known, the multipliers \( u \) and \( \lambda(t) \) are determined with Eqs. (31). Then, \( \Delta u(t)/\alpha, \Delta \pi/\alpha \) are obtained with Eqs. (19b), (19c). Finally \( \Delta x(t)/\alpha \) is determined by forward integration of (13a), subject to (13b).

We note that, except for the determination of the stepsize, the gradient iteration is completed. The stepsize \( \alpha \) can be determined a posteriori using some suitable search procedure.
6. **Basic Problem, Restoration Phase, Primal Formulation**

The restoration phase of the sequential gradient-restoration algorithm involves one or more iterations and is designed to force constraint satisfaction to a predetermined accuracy. The restoration phase is terminated whenever Ineq. (9a) is satisfied.

Each restorative iteration is started with nominal functions $x(t)$, $u(t)$, $\pi$ violating at least one of Eqs. (2a), (2c), thereby violating Ineq. (9a). It leads from the nominal functions $x(t)$, $u(t)$, $\pi$ to the varied functions $\tilde{x}(t)$, $\tilde{u}(t)$, $\tilde{\pi}$ through the displacements $\Delta x(t)$, $\Delta u(t)$, $\Delta \pi$. By definition,

$$\tilde{x}(t) = x(t) + \Delta x(t), \quad (41a)$$

$$\tilde{u}(t) = u(t) + \Delta u(t), \quad (41b)$$

$$\tilde{\pi} = \pi + \Delta \pi. \quad (41c)$$

The displacements $\Delta x(t)$, $\Delta u(t)$, $\Delta \pi$ are computed by solving the following auxiliary minimization problem.

**Problem (RP).** Minimize the norm squared of the vectors $\Delta u(t)$, $\Delta \pi$, with respect to the vectors $\Delta x(t)$, $\Delta u(t)$, $\Delta \pi$ which satisfy the linearized form of the constraints (2). Therefore, we minimize the functional

$$I_p = (1/2\alpha)[\int_0^1 \Delta u^T \Delta u dt + \Delta \pi^T \Delta \pi], \quad (42)$$

with respect to the vectors $\Delta x(t)$, $\Delta u(t)$, $\Delta \pi$ which satisfy the constraints

$$\Delta \dot{x} + \phi_x^T \Delta x + \phi_u^T \Delta u + \phi_{\pi}^T \Delta \pi + \alpha(\tilde{x} + \phi) = 0, \quad 0 \leq t \leq 1, \quad (43a)$$

$$(\Delta x)_0 = 0, \quad (43b)$$

$$(\psi_x^T \Delta x + \psi_{\pi}^T \Delta \pi + \alpha \psi)_1 = 0. \quad (43c)$$
The symbol $\alpha$ in (42)-(43) denotes the restoration stepsize, $0 < \alpha \leq 1$.

From calculus of variations, it is known Problem (RP) is of the Bolza type. It can be recast as that of minimizing the augmented functional

$$J_P = I_P + L_P,$$

subject to (43), where $L_P$ denotes the Lagrangian functional

$$L_P = \int_0^1 \lambda^T [\Delta \dot{x} + \phi_x^T \Delta x + \phi_u^T \Delta u + \phi_{\Delta t}^T \Delta u + \alpha(\dot{x} + \phi)] dt$$

$$+ \mu^T (\psi_x^T \Delta x + \psi_{\Delta t}^T \Delta u + \alpha \psi)_1.$$  \hfill (45)

In Eq. (45), $\lambda(t)$ denotes an $n$-vector Lagrange multiplier and $\mu$ a $b$-vector Lagrange multiplier.

For Problem (RP), let the Hamiltonian $H$ and the augmented functions $K, G$ be defined by

$$H = (1/2 \alpha) \Delta u^T \Delta u + \lambda^T [\phi_x^T \Delta x + \phi_u^T \Delta u + \phi_{\Delta t}^T \Delta u + \alpha(\dot{x} + \phi)],$$  \hfill (46a)

$$K = (1/2 \alpha) \Delta t^T \Delta t,$$  \hfill (46b)

$$G = \mu^T (\psi_x^T \Delta x + \psi_{\Delta t}^T \Delta u + \alpha \psi).$$  \hfill (46c)

Then, the first-order optimality conditions for Problem (RP) take the form

$$\dot{\lambda} - H_{\Delta x} = 0, \quad 0 \leq t \leq 1,$$  \hfill (47a)

$$H_{\Delta u} = 0, \quad 0 \leq t \leq 1,$$  \hfill (47b)

$$\int_0^1 H_{\Delta t} dt + K_{\Delta t} + (G_{\Delta t})_1 = 0,$$  \hfill (47c)

$$(\lambda + G_{\Delta x})_1 = 0.$$  \hfill (47d)
On account of Eqs. (46), the explicit form of Eqs. (47) is the following:

\[ \dot{\lambda} - \phi_x \lambda = 0, \quad 0 \leq t \leq 1, \quad (48a) \]

\[ \phi_u \lambda + \Delta u/\alpha = 0, \quad 0 \leq t \leq 1, \quad (48b) \]

\[ \int_0^1 \phi_{\pi} \lambda dt + (\psi_{\pi} \mu)_1 + \Delta \pi / \alpha = 0, \quad (48c) \]

\[ (\lambda + \psi_x \mu)_1 = 0. \quad (48d) \]

Summarizing, we seek the functions \( \Delta x(t), \Delta u(t), \Delta \pi \) and the multipliers \( \lambda(t), \mu \) such that the feasibility equations (43) and the optimality conditions (48) are satisfied.
7. Basic Problem, Restoration Phase, Dual Formulation

Let \( y(t), z \) denote the vectors defined by

\[
\phi_u \lambda + y = 0, \tag{49a}
\]

\[
\int_0^1 \phi_t \lambda dt + (\psi_{\mu \lambda})_1 + z = 0. \tag{49b}
\]

It is interesting to note that the vectors \( \lambda(t), \mu \) and \( y(t), z \) can also be obtained by solving the following auxiliary minimization problem.

**Problem (RD).** Minimize the functional

\[
I_D = \frac{1}{2} [\int_0^1 y^T y dt + z^T z] - \int_0^1 \lambda^T (\dot{x} + \phi) dt + (\mu^T \psi)_1, \tag{50}
\]

with respect to the vectors \( \lambda(t), \mu \) and \( y(t), z \) which satisfy the constraints

\[
\dot{\lambda} - \phi_x \lambda = 0, \quad 0 \leq t \leq 1, \tag{51a}
\]

\[
\phi_u \lambda + y = 0, \quad 0 \leq t \leq 1, \tag{51b}
\]

\[
\int_0^1 \phi_t \lambda dt + (\psi_{\mu \lambda})_1 + z = 0, \tag{51c}
\]

\[
(\lambda + \psi_{\mu \lambda})_1 = 0. \tag{51d}
\]

From calculus of variations, it is known that Problem (RD) is of the Bolza type. It can be recast as that of minimizing the augmented functional

\[
J_D = I_D + L_D, \tag{52}
\]

subject to (51), where \( L_D \) denotes the Lagrangian functional

\[
L_D = \int_0^1 [\lambda^T (\dot{x} - \phi_x \lambda) - y^T (\phi_u \lambda + y) - z^T \phi_t \lambda] dt
\]

\[
- z^T (\psi_{\mu \lambda})_1 - z^T z - \psi^T (\lambda + \psi_{\mu \lambda})_1. \tag{53}
\]
In Eq. (53), $\lambda_*(t)$ denotes an $n$-vector Lagrange multiplier, $y_*(t)$ an $m$-vector Lagrange multiplier, $z_*$ a $p$-vector Lagrange multiplier, and $\nu_*$ an $n$-vector Lagrange multiplier.

For Problem (RD), let the Hamiltonian $H$ and the augmented functions $K$, $G$ be defined by
\begin{align}
H &= (1/2)y^T y - \lambda^T (\dot{x} + \phi) - \lambda^T \phi X \lambda - y^T (\phi_u \lambda + y) - z^T \phi_z \lambda, \quad (54a) \\
K &= (1/2)z^T z - z^T z, \quad (54b) \\
G &= -\mu^T \psi - z^T \psi \mu - \nu^T (\lambda + \psi \mu). \quad (54c)
\end{align}

Then, the first-order optimality conditions for Problem (RD) take the form
\begin{align}
\dot{\lambda}_* - H_\lambda &= 0, \quad 0 \leq t \leq 1, \quad (55a) \\
H_y &= 0, \quad 0 \leq t \leq 1, \quad (55b) \\
(G_\mu)_1 &= 0, \quad (55c) \\
K_z &= 0, \quad (55d) \\
(\lambda_*)_0 &= 0, \quad (55e) \\
(\lambda_* + G_\lambda)_1 &= 0. \quad (55f)
\end{align}

On account of Eqs. (54), the explicit form of Eqs. (55) is the following:
\begin{align}
\dot{\lambda}_* + \phi^T \lambda_* + \phi^T y_* + \phi^T z_* + (\dot{x} + \phi) &= 0, \quad 0 \leq t \leq 1, \quad (56a) \\
y - y_* &= 0, \quad 0 \leq t \leq 1, \quad (56b) \\
(\psi^T \lambda_* + \psi^T z_* + \psi)_1 &= 0. \quad (56c)
\end{align}
\[ z - z_* = 0, \quad (56d) \]
\[ (\lambda_*)_0 = 0, \quad (56e) \]
\[ (\lambda_* - \nu_*)_1 = 0. \quad (56f) \]

Let the following substitutions be employed:

\[ \lambda_* = \Delta x(t)/\alpha, \quad 0 \leq t \leq 1, \quad (57a) \]
\[ y_* = \Delta u(t)/\alpha, \quad 0 \leq t \leq 1, \quad (57b) \]
\[ z_* = \Delta v/\alpha, \quad (57c) \]
\[ \nu_* = \Delta x(1)/\alpha. \quad (57d) \]

Then, one can readily verify that Eqs. (56) and (51) of Problem (RD) reduce to Eqs. (43) and (48) of Problem (RP). Clearly, after the transformation (57) is applied, the solution of Problem (RD) yields the solution of Problem (RP) and vice versa. This means that the multipliers \( \lambda(t), \mu \) associated with the restoration phase of SGRA are endowed with a duality property: They also minimize the quadratic functional (50), subject to (51), for given state, control, and parameter.
8. **Basic Problem, Restoration Phase, Computational Implications**

In this section, we exploit the previously established duality property and show that the execution of a restorative iteration can be reduced to solving a mathematical programming problem involving a finite number of parameters as unknowns. Hence, the algorithmic efficiency of the restoration phase of SGRA can be enhanced.

First, we consider Eqs. (5la) and (51d). We observe that, if \( \mu \) is assigned, \( \lambda(1) \) can be computed with (51d) and \( \lambda(t) \) can be computed by backward integration of (51a). Next, we execute \( b \) backward integrations, using Eqs. (51a) and (51d) in combination with the following choices for the multiplier \( \mu \):

\[
\mu_1 = \delta_1, \quad \mu_2 = \delta_2, \ldots, \mu_b = \delta_b. \tag{58}
\]

In (58), \( \delta_1, \delta_2, \ldots, \delta_b \) denote the vectors corresponding to the columns of the identity matrix of order \( b \).

Let \( \lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t) \) denote the solutions of Eqs. (51a) and (51d), corresponding to the choices (58) for the multiplier \( \mu \). Let \( \tilde{\mu} \) denote the \( b \times b \) matrix\(^3\)

\[
\tilde{\mu} = [\mu_1, \mu_2, \ldots, \mu_b]; \tag{59a}
\]

let \( \tilde{\lambda}(t) \) denote the \( n \times b \) matrix

\[
\tilde{\lambda}(t) = [\lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t)]; \tag{59b}
\]

and let \( k \) denote the \( b \)-vector

\[
k = [k_1, k_2, \ldots, k_b]^T. \tag{59c}
\]

\(^3\)Clearly, \( \tilde{\mu} = I \), where \( I \) denotes the identity matrix of order \( b \).
If the method of complementary functions is employed, the general solution
of Eqs. (51a) and (51d) can be written in the form

\[ u = \tilde{u}k, \quad (60a) \]
\[ \lambda(t) = \tilde{\lambda}(t)k. \quad (60b) \]

Next, we combine Eqs. (51b), (51c) with Eqs. (60) and obtain the relations

\[ y = -\phi_0 \tilde{\lambda}k, \quad 0 \leq t \leq 1, \quad (61a) \]
\[ z = -\int_0^1 \phi_n \tilde{\lambda} dt + (\psi_1)\tilde{\mu}k. \quad (61b) \]

These relations show that \( y(t), z \) depend only on the parameter \( k \).

Finally, upon combining (50) and (61), we obtain the following quadratic
function of \( k \):

\[ I_D = (1/2)k^T Mk + D^T k. \quad (62) \]

Here, the matrix \( M \) and the vector \( D \) are known. They are defined by

\[ M = \int_0^1 (\phi_u \tilde{\lambda})^T (\phi_u \tilde{\lambda}) dt + \int_0^1 \phi_n \tilde{\lambda} dt + (\psi_1)\tilde{\mu}]^T [\int_0^1 \phi_n \tilde{\lambda} dt + (\psi_1)\tilde{\mu}], \quad (63a) \]
\[ D = -\int_0^1 \lambda^T (\dot{\chi} + \phi) dt - (\tilde{\mu}^T \psi)_1. \quad (63b) \]

Because of the duality property, the parameter \( k \) can be obtained by
minimizing (62). Clearly, this auxiliary minimization problem is a mathematical
programming problem, whose first-order optimality condition takes the form

\[ I_{Dk} = 0. \quad (64) \]

Hence, \( k \) is determined by solving the following linear algebraic system:

\[ Mk + D = 0, \quad (65) \]
whose dimension is $b$. Once $k$ is known, the multipliers $\mu$ and $\lambda(t)$ are determined with Eqs. (60). Then, $\Delta u(t)/\alpha$, $\Delta u/\alpha$ are obtained with Eqs. (48b), (48c). Finally, $\Delta x(t)/\alpha$ is determined by forward integration of (43a), subject to (43b).

We note that, except for the determination of the stepsize, the restorative iteration is completed. The stepsize $\alpha$ can be determined a posteriori using some suitable search procedure.
9. **Basic Problem, Dual Sequential Gradient-Restoration Algorithm**

The sequential gradient-restoration algorithm involves a sequence of two-phase cycles, each cycle including a gradient phase and a restoration phase. In a complete gradient-restoration cycle, the value of the functional is decreased, while the constraints are satisfied to a predetermined accuracy; in the gradient phase, the value of the augmented functional is decreased, while avoiding excessive constraint violation; in the restoration phase, the constraint error is decreased, while avoiding excessive change in the value of the functional.

It must be noted that, while the gradient phase involves a single iteration, the restoration phase might involve several iterations. The decision of whether to execute a gradient iteration or a restorative iteration is based on the measurement of a single scalar quantity, the constraint error \( P \), given by Eq. (8a). If the constraint error satisfies Ineq. (9a), a gradient iteration is executed; if the constraint error violates Ineq. (9a), a restorative iteration is executed.

For both gradient iterations and restorative iterations, the following terminology is employed: \( x(t), u(t), \pi \) denote the nominal functions; \( \bar{x}(t), \bar{u}(t), \bar{\pi} \) denote the varied functions; \( \Delta x(t), \Delta u(t), \Delta \pi \) denote the perturbations leading from the nominal functions to the varied functions; and \( A(t), B(t), C \) denote the perturbations per unit stepsize \( \alpha \). Then, the following relations hold:

\[
\bar{x}(t) = x(t) + \Delta x(t) = x(t) + \alpha A(t), \tag{66a}
\]

\[
\bar{u}(t) = u(t) + \Delta u(t) = u(t) + \alpha B(t), \tag{66b}
\]

\[
\bar{\pi} = \pi + \Delta \pi = \pi + \alpha C. \tag{66c}
\]
Thus, each iteration involves the following operations: (i) the determination of the functions A(t), B(t), C; and (ii) the determination of the stepsize $\alpha$.

Depending on whether the primal formulation is used or the dual formulation is used, one obtains a primal sequential gradient-restoration algorithm (PSGRA) and a dual sequential gradient-restoration algorithm (DSGRA). Since PSGRA has been discussed in Refs. 1-3, we restrict our attention to DSGRA.

**Gradient Iteration.** Its objective is to reduce the augmented functional $J$, while the constraints are satisfied to first order.

**Step 1.** Assume nominal functions $x(t)$, $u(t)$, $\pi$ which satisfy the constraints (2) within the predetermined accuracy (9a).

**Step 2.** For the nominal functions, compute the vectors $f_x$, $f_u$, $f_\pi$ and the matrices $\phi_x$, $\phi_u$, $\phi_\pi$ along the interval of integration; compute the vectors $g_x$, $g_\pi$ and the matrices $\psi_x$, $\psi_\pi$ at the final point.

**Step 3.** Execute $b+1$ backward integrations of the multiplier equations, using (22a) and (22d) in combination with the choices (29) for the multiplier $\mu$. Determine the particular solutions $\lambda_1(t)$, $\lambda_2(t)$, ..., $\lambda_b(t)$, $\lambda_{b+1}(t)$. Determine the matrices $\bar{\mu}$ and $\bar{\lambda}(t)$, given by (30a) and (30b).

**Step 4.** Using the matrices $\bar{\mu}$ and $\bar{\lambda}(t)$ of Step 3, compute the matrix $M$, the vector $D$, and the scalar $H$, given by Eqs. (37). Determine the vector $k$ and the scalar $\beta$ by solving the linear algebraic system (40).

**Step 5.** With the vector $k$ known, compute the multipliers $\mu$ and $\lambda(t)$ using Eqs. (31). Alternatively, compute the multiplier $\mu$ with (31a) and the multiplier $\lambda(t)$ by backward integration of the multiplier equations, using (22a) and (22d) in combination with (31a).

**Step 6.** Determine the control change per unit stepsize $B(t)$ and the parameter change per unit stepsize $C$ using the relations
\[ B = -f_u - \phi_u^\lambda, \quad 0 \leq t \leq 1, \quad (67a) \]
\[ C = -\int_0^1 (f_{\pi} + \phi_{\pi}^\lambda) dt - (g_{\pi} + \psi_{\pi} u) \lambda. \quad (67b) \]

Determine the state change per unit stepsize \( A(t) \) by forward integration of the linearized state equation, using the relations
\[ \dot{A} + \phi_{\lambda}^T A + \phi_u^T B + \phi_{\pi}^T C = 0, \quad 0 \leq t \leq 1, \quad (68a) \]
\[ (A)_0 = 0, \quad (68b) \]
in combination with Eqs. (67).

**Step 7.** With the functions \( A(t), B(t), C \) known, the one-parameter family of varied functions (66) can be formed. For this family, the functionals \( I, J, P \) take the following form:
\[ \tilde{I} = \tilde{I}(\alpha), \quad \tilde{J} = \tilde{J}(\alpha), \quad \tilde{P} = \tilde{P}(\alpha). \quad (69) \]

Then, the stepsize \( \alpha \) is computed by a one-dimensional search on the function \( \tilde{J}(\alpha) \) until the following relations are satisfied:
\[ \tilde{J}(\alpha) < \tilde{J}(0), \quad (70a) \]
\[ \tilde{P}(\alpha) \leq P_*, \quad (70b) \]

where \( P_* \) is a preselected number, not necessarily small.

The simplest way of ensuring satisfaction of (70) is to employ a bisection process, starting from the reference stepsize \( \alpha = \alpha_0 \). In turn, the reference stepsize \( \alpha_0 \) can be obtained by the combination of a scanning process and a cubic interpolation process. With the scanning process, one brackets the minimum point of the function \( \tilde{J}(\alpha) \). With the cubic interpolation process, either single-step or multi-step, one obtains an approximation to the reference
stepsize $\alpha_0$. This is the stepsize which yields the minimum of the cubic approximation to $\tilde{J}(\alpha)$.

The details of the one-dimensional search can be found in Refs. 1-3 and related publications. They are omitted here, for the sake of brevity.

**Step 8.** Once the stepsize $\alpha$ is known, compute the varied functions $\tilde{x}(t)$, $\tilde{u}(t)$, $\tilde{\pi}$ with Eqs. (66). In this way, the gradient iteration is completed.

**Restorative Iteration.** Its objective is to reduce the constraint error $P$, while the constraints are satisfied to first order and the norm squared of the vector $\Delta u(t)$, $\Delta \pi$ is minimized.

**Step 1.** Assume nominal functions $x(t)$, $u(t)$, $\pi$ which satisfy the constraint (2b), but violate at least one of the constraints (2a), (2c).

**Step 2.** For the nominal functions, compute the vector $\dot{x} + \phi$ and the matrices $\phi_x$, $\phi_u$, $\phi_\pi$ along the interval of integration; compute the vector $\psi$ and the matrices $\psi_x$, $\psi_\pi$ at the final point.

**Step 3.** Execute $b$ backward integrations of the multiplier equations, using (51a) and (51d) in combination with the choices (58) for the multiplier $\mu$. Determine the complementary solutions $\lambda_1(t)$, $\lambda_2(t)$, ..., $\lambda_b(t)$. Determine the matrices $\tilde{\mu}$ and $\tilde{\lambda}(t)$, given by (59a) and (59b).

**Step 4.** Using the matrices $\tilde{\mu}$ and $\tilde{\lambda}(t)$ of Step 3, compute the matrix $M$ and the vector $D$ given by Eqs. (63). Determine the vector $k$ by solving the linear algebraic system (65).

**Step 5.** With the vector $k$ known, compute the multipliers $\mu$ and $\lambda(t)$ using Eqs. (60). Alternatively, compute the multiplier $\mu$ with (60a) and the multiplier $\lambda(t)$ by backward integration of the multiplier equations, using (51a) and (51d) in combination with (60a).

**Step 6.** Determine the control change per unit stepsize $B(t)$ and the parameter change per unit stepsize $C$ using the relations
\[ B = -\phi_u \lambda, \quad 0 \leq t \leq 1, \quad (71a) \]

\[ C = -\int_0^1 \phi_n \lambda dt - \left( \psi_n \mu \right)_1. \quad (71b) \]

Determine the state change per unit stepsize \( A(t) \) by forward integration of the linearized state equation, using the relations

\[ \dot{A} + \phi_x^T A + \phi_u^T B + \phi_n^T C + (\dot{x} + \phi) = 0, \quad 0 \leq t \leq 1, \quad (72a) \]

\[ (A)_0 = 0, \quad (72b) \]

in combination with Eqs. (71).

**Step 7.** With the functions \( A(t), B(t), C \) known, the one-parameter family of varied functions (66) can be formed. For this family, the functional \( P \) takes the following form:

\[ \tilde{P} = \tilde{P}(\alpha). \quad (73) \]

Then, the stepsize \( \alpha \) is computed by a one-dimensional search on the function (73) until the following relation is satisfied:

\[ \tilde{P}(\alpha) < \tilde{P}(0). \quad (74) \]

The simplest way of ensuring satisfaction of (74) is to employ a bisection process, starting from the reference stepsize \( \alpha = \alpha_0 \). Here, the correct reference stepsize is \( \alpha_0 = 1 \), in that it yields one-step restoration if the constraints (2) are linear.

**Step 8.** Once the stepsize \( \alpha \) is known, compute the varied functions \( \bar{x}(t), \bar{u}(t), \bar{y} \) with Eqs. (66). In this way, the restorative iteration is completed.
**Gradient Phase.** The gradient phase includes a single gradient iteration. Hence, the gradient phase is the same as the gradient iteration discussed previously.

**Restoration Phase.** The restoration phase might include several restorative iterations. In each restorative iteration, the constraint error is reduced in accordance with Ineq. (74). The restoration phase is terminated whenever the constraint error reaches a level compatible with Ineq. (9a).

**Gradient-Restoration Cycle.** As stated before, a complete gradient-restoration cycle includes a gradient phase and a restoration phase. After a restoration phase is completed, one must verify whether the following inequality is satisfied:

\[ I < \hat{I}; \tag{75} \]

here, \( I \) denotes the value of the functional (1) at the end of the present restoration phase and \( \hat{I} \) denotes the value of the functional (1) at the end of the previous restoration phase. If Ineq. (75) is satisfied, one starts the next cycle of the sequential gradient-restoration algorithm. If Ineq. (75) is violated, one returns to the previous gradient phase and reduces the gradient stepsize (using a bisection process) until, after restoration, the functional \( I \) finally decreases.

**Starting Condition.** The present algorithm must be started with nominal functions \( x(t), u(t), \pi \) which satisfy the constraint (2b), but not necessarily the constraints (2a) and (2c). If the nominal functions violate Ineq. (9a), the algorithm starts with a restoration phase; hence, the first cycle is a half cycle involving only a restoration phase. If the nominal functions satisfy Ineq. (9a), the algorithm starts with a gradient phase; hence, the first
cycle is a complete cycle, involving both a gradient phase and a restoration phase.

**Stopping Condition.** The present algorithm is terminated whenever Ineqs. (9a) and (9b) are satisfied simultaneously. Note that Ineq. (9a) is verified at the end of a restoration phase/beginning of a gradient phase. On the other hand, Ineq. (9b) must be verified at the beginning of a gradient phase, after the multipliers $\lambda(t)$, $\mu$ are computed and before the search for the gradient stepsize is executed.
10. **General Problem, Optimality Conditions**

**Problem (P2).** We consider the problem of minimizing the functional

\[ I = \int_0^1 f(x,u,v,\pi, t) dt + [h(x,\pi)]_0 + [g(x,\pi)]_1, \]  

(76)

with respect to the \(n\)-vector state \(x(t)\), the \(m\)-vector control \(u(t)\), the \(c\)-vector control \(v(t)\), and the \(p\)-vector parameter \(\pi\) which satisfy the constraints

\[ \dot{x} + \phi(x,u,v,\pi, t) = 0, \quad 0 \leq t \leq 1, \]  

(77a)

\[ S(x,u,v,\pi, t) = 0, \quad 0 \leq t \leq 1, \]  

(77b)

\[ [\omega(x,\pi)]_0 = 0, \]  

(77c)

\[ [\psi(x,\pi)]_1 = 0. \]  

(77d)

In the above equations, \(f\) is a scalar; \(h\) is a scalar; \(g\) is a scalar; \(\phi\) is an \(n\)-vector; \(S\) is a \(c\)-vector; \(\omega\) is an \(a\)-vector, \(a \leq n\); and \(\psi\) is a \(b\)-vector, \(b \leq n\). We assume that the first and second derivatives of the functions \(f,h,g,\phi,S,\omega,\psi\) with respect to the vectors \(x,u,v,\pi\) exist and are continuous.

We also assume that the \(n \times a\) matrix \(\omega_x\) has rank \(a\) at the initial point, that the \(n \times b\) matrix \(\psi_x\) has rank \(b\) at the final point, that the \(c \times c\) matrix \(S_v\) has rank \(c\) everywhere along the interval of integration, and that the constrained minimum exists.

From calculus of variations, it is known that Problem (P2) is of the Bolza type. It can be recast as that of minimizing the augmented functional

\[ J = I + L, \]  

(78)

subject to (77), where \(L\) denotes the Lagrangian functional.
\[ L = \int_0^1 \lambda^T (\dot{x} + \phi) dt + \int_0^1 \rho^T S dt + (\sigma^T \omega)_0 + (\mu^T \psi)_1. \]  \hspace{1cm} (79)

In Eq. (79), \( \lambda(t) \) denotes an \( n \)-vector Lagrange multiplier, \( \rho(t) \) denotes a \( c \)-vector Lagrange multiplier, \( \sigma \) denotes an \( a \)-vector Lagrange multiplier, and \( \mu \) denotes a \( b \)-vector Lagrange multiplier.

For Problem (P2), let the Hamiltonian \( H \) and the augmented functions \( N, G \) be defined by

\[ H = f + \lambda^T \phi + \rho^T S, \] \hspace{1cm} (80a)

\[ N = h + \sigma^T \omega, \] \hspace{1cm} (80b)

\[ G = g + \mu^T \psi. \] \hspace{1cm} (80c)

Then, the first-order optimality conditions for Problem (P2) take the form

\[ \dot{\lambda} - H_x = 0, \hspace{1cm} 0 \leq t \leq 1, \] \hspace{1cm} (81a)

\[ H_u = 0, \hspace{1cm} 0 \leq t \leq 1, \] \hspace{1cm} (81b)

\[ H_v = 0, \hspace{1cm} 0 \leq t \leq 1, \] \hspace{1cm} (81c)

\[ \int_0^1 H_u dt + (N_u)_0 + (G_u)_1 = 0, \] \hspace{1cm} (81d)

\[ (-\lambda + N_x)_0 = 0, \] \hspace{1cm} (81e)

\[ (\lambda + G_x)_1 = 0. \] \hspace{1cm} (81f)

On account of Eqs. (80), the explicit form of Eqs. (81) is the following:

\[ \dot{\lambda} - f_x - \phi_x \lambda - S_x \rho = 0, \hspace{1cm} 0 \leq t \leq 1, \] \hspace{1cm} (82a)

\[ f_u + \phi_u \lambda + S_u \rho = 0, \hspace{1cm} 0 \leq t \leq 1, \] \hspace{1cm} (82b)
\[ f_v + \phi_v \lambda + S_v \rho = 0, \quad 0 \leq t \leq 1, \quad (82c) \]

\[ \int_0^1 (f_\pi + \phi_\pi \lambda + S_\pi \rho) dt + (h_\pi + \omega_\pi \sigma)_0 + (g_\pi + \psi_\pi \mu)_1 = 0, \quad (82d) \]

\[ (-\lambda + h_x + \omega_x \sigma)_0 = 0, \quad (82e) \]

\[ (\lambda + g_x + \psi_x \mu)_1 = 0. \quad (82f) \]

Summarizing, we seek the functions \( x(t), u(t), v(t), \pi \) and the multipliers \( \lambda(t), \rho(t), \sigma, \mu \) such that the feasibility equations (77) and the optimality conditions (82) are satisfied.

**Alternative Form.** Under the assumption that the matrix \( S_v \) is nonsingular, Eq. (82c) yields the following solution for the Lagrange multiplier \( \rho \) associated with the nondifferential constraint (77b):

\[ \rho = -S_v^{-1}(f_v + \phi_v \lambda), \quad 0 \leq t \leq 1. \quad (83) \]

As a consequence, the optimality conditions (82) can be rewritten in the following alternative form:

\[ \dot{\lambda} - F_x - \phi_x \lambda = 0, \quad 0 \leq t \leq 1, \quad (84a) \]

\[ F_u + \phi_u \lambda = 0, \quad 0 \leq t \leq 1, \quad (84b) \]

\[ \int_0^1 (F_\pi + \phi_\pi \lambda) dt + (h_\pi + \omega_\pi \sigma)_0 + (g_\pi + \psi_\pi \mu)_1 = 0, \quad (84c) \]

\[ (-\lambda + h_x + \omega_x \sigma)_0 = 0, \quad (84d) \]

\[ (\lambda + g_x + \psi_x \mu)_1 = 0. \quad (84e) \]

Here, \( F_x, F_u, F_\pi \) denote the vectors...
\[ F_X = f_X - S_X S_V^{-1} f_V, \]  
(85a)

\[ F_u = f_u - S_u S_V^{-1} f_V, \]  
(85b)

\[ F_\pi = f_\pi - S_\pi S_V^{-1} f_V, \]  
(85c)

and \( \phi_X, \phi_u, \phi_\pi \) denote the matrices

\[ \phi_X = \phi_X - S_X S_V^{-1} \phi_V, \]  
(86a)

\[ \phi_u = \phi_u - S_u S_V^{-1} \phi_V, \]  
(86b)

\[ \phi_\pi = \phi_\pi - S_\pi S_V^{-1} \phi_V. \]  
(86c)

Summarizing, we seek the functions \( x(t), u(t), v(t), \pi \) and the multipliers \( \lambda(t), \sigma, \mu \) such that the feasibility equations (77) and the optimality conditions (84) are satisfied. After Eqs. (77) and (84) are solved, \( \rho(t) \) is computed with (83).

In the following sections, we refer to the form (84) of the optimality conditions.

**Performance Indexes.** The form of Eqs. (77) and (84) suggests that the following scalar performance indexes are useful in computational work:

\[ P = \int_0^1 N(\dot{x} + \phi) dt + \int_0^1 N(S) dt + N(\omega)_0 + N(\psi)_1, \]  
(87a)

\[ Q = \int_0^1 N(\dot{\lambda} - F_X - \phi_X \lambda) dt + \int_0^1 N(F_u + \phi_u \lambda) dt \]

\[ + N\left[ \int_0^1 (F_\pi + \phi_\pi \lambda) dt + (h_\pi + \omega_\pi \sigma)_0 + (g_\pi + \psi_\pi \mu)_1 \right] \]

\[ + N(-\lambda + h_X + \omega_X \sigma)_0 + N(\lambda + g_X + \psi_X \mu)_1, \]  
(87b)

\[ R = P + Q. \]  
(87c)
Here, $P$ denotes the error in the constraints, $Q$ the error in the optimality conditions, and $R$ the total error in the system. Therefore, numerical convergence can be characterized by the relations

$$P \leq \varepsilon_1, \quad (88a)$$

$$Q \leq \varepsilon_2, \quad (88b)$$

or by the relation

$$R \leq \varepsilon_3, \quad (89)$$

where $\varepsilon_1$, $\varepsilon_2$, $\varepsilon_3$ are preselected, small, positive numbers.
11. **General Problem, Gradient Phase, Primal Formulation**

The gradient phase of the sequential gradient-restoration algorithm involves a single iteration and is designed to decrease the augmented functional, while avoiding excessive constraint violation. The gradient iteration is started whenever Ineq. (88a) is satisfied.

Let \( x(t), u(t), v(t), \pi \) denote the nominal functions. Let \( \bar{x}(t), \bar{u}(t), \bar{v}(t), \bar{\pi} \) denote the varied functions. Let \( \Delta x(t), \Delta u(t), \Delta v(t), \Delta \pi \) denote the displacements leading from the nominal functions to the varied functions. By definition, the following relations hold:

\[
\begin{align*}
\bar{x}(t) &= x(t) + \Delta x(t), \\
\bar{u}(t) &= u(t) + \Delta u(t), \\
\bar{v}(t) &= v(t) + \Delta v(t), \\
\bar{\pi} &= \pi + \Delta \pi.
\end{align*}
\]

The displacements \( \Delta x(t), \Delta u(t), \Delta v(t), \Delta \pi \) are computed by solving the following auxiliary minimization problem.

**Problem (GP).** Minimize the first variation of the functional (76), with respect to the vectors \( \Delta x(t), \Delta u(t), \Delta v(t), \Delta \pi \) which satisfy the linearized form of the constraints (77) plus a quadratic isoperimetric constraint imposed on the vectors \( \Delta u(t), \Delta \pi, \Delta x(0) \). Therefore, we minimize the functional

\[
I_p = \int_0^1 (f_x^T \Delta x + f_u^T \Delta u + f_v^T \Delta v + f_{\pi}^T \Delta \pi) dt + (h_x^T \Delta x + h_u^T \Delta u)_0 + (g_x^T \Delta x + g_u^T \Delta u)_1,
\]

with respect to the vectors \( \Delta x(t), \Delta u(t), \Delta v(t), \Delta \pi \) which satisfy the constraints

\[
\Delta \dot{x} + \phi_x^T \Delta x + \phi_u^T \Delta u + \phi_v^T \Delta v + \phi_{\pi}^T \Delta \pi = 0, \quad 0 \leq t \leq 1,
\]
\[ S_{\Delta x}^T + S_{\Delta u}^T + S_{\Delta v}^T + S_{\Delta \eta}^T = 0, \quad 0 \leq t \leq 1, \quad (92b) \]
\[ (\omega_{\Delta x}^T + \omega_{\Delta \eta}^T)_0 = 0, \quad (92c) \]
\[ (\psi_{\Delta x}^T + \psi_{\Delta \eta}^T)_1 = 0, \quad (92d) \]

and

\[ \int_0^1 \Delta u^T \Delta u dt + \Delta \eta^T \Delta \eta + (\Delta x^T \Delta x)_0 - \text{const} = 0. \quad (93) \]

**Alternative Form.** Under the assumption that the matrix \( S_v \) is nonsingular, Eq. (92b) yields the following solution for the dependent control displacement:

\[ \Delta v = - (S_v^{-1})^T (S_{\Delta x}^T + S_{\Delta u}^T + S_{\Delta \eta}^T), \quad 0 \leq t \leq 1. \quad (94) \]

As a consequence, upon eliminating \( \Delta v \) from Eqs. (91)-(92) and upon recalling the definitions (85)-(86), Problem (GP) is reformulated as follows: Minimize the functional

\[ I_P = \int_0^1 \left( F_{\Delta x}^T + F_{\Delta u}^T + F_{\Delta \eta}^T \right) dt + (h_{\Delta x}^T + h_{\Delta \eta}^T)_0 + (g_{\Delta x}^T + g_{\Delta \eta}^T)_1, \quad (95) \]

with respect to the vectors \( \Delta x(t), \Delta u(t), \Delta \eta \) which satisfy the constraints

\[ \Delta \dot{x} + \phi_{\Delta x}^T \Delta x + \phi_{\Delta u}^T \Delta u + \phi_{\Delta \eta}^T \Delta \eta = 0, \quad 0 \leq t \leq 1, \quad (96a) \]
\[ (\omega_{\Delta x}^T + \omega_{\Delta \eta}^T)_0 = 0, \quad (96b) \]
\[ (\psi_{\Delta x}^T + \psi_{\Delta \eta}^T)_1 = 0, \quad (96c) \]

and

\[ \int_0^1 \Delta u^T \Delta u dt + \Delta \eta^T \Delta \eta + (\Delta x^T \Delta x)_0 - \text{const} = 0. \quad (97) \]

**From calculus of variations, it is known that Problem (GP) is of the Bolza type.** It can be recast as that of minimizing the augmented functional
\[ J_p = I_p + L_p, \]
subject to (96)-(97), where \( L_p \) denotes the Lagrangian functional

\[
L_p = \int_0^1 \lambda^T (\Delta x + \Phi^T \Delta u + \Phi^T \Delta T) dt + \sigma^T (\omega_x \Delta x + \omega_\Delta \Delta T) + \\
+ \mu^T (\psi_x \Delta x + \psi_\Delta \Delta T) t + (1/2 \alpha) \int_0^1 \Delta u \Delta u dt + \Delta \Delta T + (\Delta x \Delta x) - \text{const}. \]

(99)

In Eq. (99), \( \lambda(t) \) denotes an \( n \)-vector Lagrange multiplier, \( \sigma \) an \( a \)-vector Lagrange multiplier, \( \mu \) a \( b \)-vector Lagrange multiplier, and \( 1/2 \alpha \) a scalar Lagrange multiplier.

For Problem (GP), let the Hamiltonian \( H \) and the augmented functions \( K, N, G \) be defined as follows:

\[
H = F^T_x \Delta x + F^T_u \Delta u + F^T_\phi \Delta T + \lambda^T (\Phi^T \Delta x + \Phi^T \Delta u + \Phi^T \Delta T) + (1/2 \alpha) \Delta u \Delta u, \]

(100a)

\[
K = (1/2 \alpha) \Delta \Delta T, \]

(100b)

\[
N = h^T_x \Delta x + h^T_\phi \Delta T + \sigma^T (\omega_x \Delta x + \omega_\Delta \Delta T) + (1/2 \alpha) \Delta x \Delta x, \]

(100c)

\[
G = g^T_x \Delta x + g^T_\phi \Delta T + \mu^T (\psi_x \Delta x + \psi_\phi \Delta T). \]

(100d)

Then, the first-order optimality conditions of Problem (GP) take the form

\[
\dot{\lambda} = H_{\Delta x}, \quad 0 \leq t \leq 1, \quad (101a)
\]

\[
H_{\Delta u} = 0, \quad 0 \leq t \leq 1, \quad (101b)
\]

\[
\int_0^1 H_{\Delta \Delta T} dt + K_{\Delta \Delta T} + (N_{\Delta \Delta T})_0 + (G_{\Delta \Delta T})_1 = 0, \quad (101c)
\]

\[
(-\lambda + N_{\Delta x})_0 = 0, \quad (101d)
\]
\[(\lambda + G_{\Delta x})_1 = 0.\] (101e)

On account of Eqs. (100), the explicit form of Eqs. (101) is the following:

\[\dot{\lambda} - F_x - \Phi_x \lambda = 0, \quad 0 \leq t \leq 1, \] (102a)

\[F_u + \Phi_u \lambda + \Delta u/\alpha = 0, \quad 0 \leq t \leq 1, \] (102b)

\[\int_0^1 (F_{u \lambda} + \Phi_{u \lambda}) dt + (h_{u \lambda} + \omega_{u \lambda})_0 + (q_{u \lambda} + \psi_{u \lambda})_1 + \Delta u/\alpha = 0, \] (102c)

\[(-\lambda + h_x + \omega_x \sigma + \Delta x/\alpha)_0 = 0, \] (102d)

\[(\lambda + g_x + \psi_x \mu)_1 = 0. \] (102e)

Summarizing, we seek the functions \(\Delta x(t), \Delta u(t), \Delta \tau\) and the multipliers \(\lambda(t),\sigma, \mu, 1/2\alpha\) such that the feasibility equations (96)-(97) and the optimality conditions (102) are satisfied. After Eqs. (96), (97), (102) are solved, \(\Delta v(t)\) is obtained with (94).
12. General Problem, Gradient Phase, Dual Formulation

Let $y(t), z, w$ denote the vectors defined by

\begin{align}
F_u + \phi_u \lambda + y &= 0, \\
\int_0^1 (F_\pi + \phi_\pi \lambda) dt + (h_\pi + \omega_\pi)0 + (g_\pi + \psi_\pi u)_1 + z &= 0, \\
(-\lambda + h_x + \omega_x)0 + w &= 0.
\end{align}

\hspace{1cm} (103a) \hspace{1cm} (103b) \hspace{1cm} (103c)

It is interesting to note that the vectors $\lambda(t), \sigma, u$ and $y(t), z, w$ can also be obtained by solving the following auxiliary minimization problem.

Problem (GD). Minimize the functional

\[ I_D = \left(1/2\right) \int_0^1 y^T y dt + z^T z + w^T w, \]

\hspace{1cm} (104)

with respect to the vectors $\lambda(t), \sigma, u$ and $y(t), z, w$ which satisfy the constraints

\begin{align}
\dot{\lambda} - F_x - \phi_x \lambda &= 0, \\
F_u + \phi_u \lambda + y &= 0, \\
\int_0^1 (F_\pi + \phi_\pi \lambda) dt + (h_\pi + \omega_\pi)0 + (g_\pi + \psi_\pi u)_1 + z &= 0, \\
(-\lambda + h_x + \omega_x)0 + w &= 0, \\
(\lambda + g_x + \psi_x u)_1 &= 0.
\end{align}

\hspace{1cm} (105a) \hspace{1cm} (105b) \hspace{1cm} (105c) \hspace{1cm} (105d) \hspace{1cm} (105e)

From calculus of variations, it is known that Problem (GD) is of the Bolza type. It can be recast as that of minimizing the augmented functional

\[ J_D = I_D + L_D, \]

\hspace{1cm} (106)
subject to (105), where \( L_D \) denotes the Lagrangian functional

\[
L_D = \int_0^1 \left[ \lambda^*_x (\lambda - F_x - \phi_x \lambda^*) - y^*_x (F_u + \phi_u \lambda + y) - z^*_x (F_\pi + \phi_\pi \lambda) \right] dt
- z^*_x (h_\pi + \omega_\pi \sigma)_0
- z^*_x (g_\pi + \psi_\pi \mu)_1
- z^*_x z
- w^*_x (-\lambda + h_X + \omega_X \sigma)_0
- w^*_x w - \nu^*_x (\lambda + g_X + \psi_X \mu)_1.
\]  
(107)

In Eq. (107), \( \lambda^*_x(t) \) denotes an \( n \)-vector Lagrange multiplier, \( y^*_x(t) \) an \( m \)-vector Lagrange multiplier, \( z^*_x \) a \( p \)-vector Lagrange multiplier, \( w^*_x \) an \( n \)-vector Lagrange multiplier, and \( \nu^*_x \) an \( n \)-vector Lagrange multiplier.

For Problem (GD), let the Hamiltonian \( H \) and the augmented functions \( K, N, G \) be defined by

\[
H = (1/2)y^T y - \lambda^*_x (F_x + \phi_x \lambda) - y^*_x (F_u + \phi_u \lambda + y) - z^*_x (F_\pi + \phi_\pi \lambda),
\]  
(108a)

\[
K = (1/2)(z^T z + w^T w) - (z^*_x z + w^*_x w),
\]  
(108b)

\[
N = -z^*_x (h_\pi + \omega_\pi \sigma) - w^*_x (-\lambda + h_X + \omega_X \sigma),
\]  
(108c)

\[
G = -z^*_x (g_\pi + \psi_\pi \mu) - \nu^*_x (\lambda + g_X + \psi_X \mu).
\]  
(108d)

Then, the first-order optimality conditions of Problem (GD) take the form

\[
\dot{\lambda}^*_x - H_\lambda = 0, \quad 0 \leq t \leq 1,
\]  
(109a)

\[
H_y = 0, \quad 0 \leq t \leq 1,
\]  
(109b)

\[
(N_0)_0 = 0,
\]  
(109c)

\[
(G_1)_0 = 0,
\]  
(109d)

\[
K_z = 0,
\]  
(109e)
\[ K_w = 0, \quad (109f) \]
\[ (-\lambda_* + N_\lambda)_0 = 0, \quad (109g) \]
\[ (\lambda_* + G_\lambda)_1 = 0. \quad (109h) \]

On account of Eqs. (108), the explicit form of Eqs. (109) is the following:

\[ \begin{align*}
\dot{\lambda}_* + \phi_x^T \lambda_* + \phi_u^T y_* + \phi_z^T z_* &= 0, \quad 0 \leq t \leq 1, \\
y - y_* &= 0, \quad 0 \leq t \leq 1, \\
(\omega_x^T w_* + \omega_z^T z_*)_0 &= 0, \\
(\psi_x^T v_* + \psi_z^T z_*)_1 &= 0, \\
z - z_* &= 0, \\
w - w_* &= 0, \\
(\lambda_* - w_*)_0 &= 0, \\
(\lambda_* - v_*)_1 &= 0.
\end{align*} \quad (110) \]

Let the following substitutions be employed:

\[ \begin{align*}
\lambda_* &= \Delta x(t)/\alpha, \quad 0 \leq t \leq 1, \\
y_* &= \Delta u(t)/\alpha, \quad 0 \leq t \leq 1, \\
z_* &= \Delta \omega/\alpha, \\
w_* &= \Delta x(0)/\alpha, \\
v_* &= \Delta x(1)/\alpha.
\end{align*} \quad (111) \]
Then, one can readily verify that Eqs. (110) and (105) of Problem (GD) reduce to Eqs. (92) and (102) of Problem (GP). Clearly, after the transformation (111) is applied, the solution of Problem (GD) yields the solution of Problem (GP) and vice versa. This means that the multipliers $\lambda(t)$, $\sigma$, $\mu$ associated with the gradient phase of SGRA are endowed with a duality property: They also minimize the quadratic functional (104), subject to (105), for given state, control, and parameter.
13. General Problem, Gradient Phase, Computational Implications

In this section, we exploit the previously established duality property and show that the execution of a gradient iteration can be reduced to solving a mathematical programming problem involving a finite number of parameters as unknowns. Hence, the algorithmic efficiency of the gradient phase of SGRA can be enhanced.

First, we consider Eqs. (105a) and (105e). We observe that, if $\mu$ is assigned, $\lambda(1)$ can be computed with (105e) and $\lambda(t)$ can be computed by backward integration of (105a). Next, we execute $b+1$ backward integrations, using Eqs. (105a) and (105e) in combination with the following choices for the multiplier $\mu$:

$$\mu_1 = \delta_1, \quad \mu_2 = \delta_2, \ldots, \quad \mu_b = \delta_b,$$

$$\mu_{b+1} = 0.$$  \hspace{1cm} (112a)

In (112a), $\delta_1, \delta_2, \ldots, \delta_b$ denote the vectors corresponding to the columns of the identity matrix of order $b$; in (112b), 0 denotes the null vector of dimension $b$.

Let $\lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t), \lambda_{b+1}(t)$ denote the particular solutions of Eqs. (105a) and (105e), corresponding to the choices (112) for the multiplier $\mu$. Let $\tilde{\mu}$ denote the $bx(b+1)$ matrix

$$\tilde{\mu} = [\mu_1, \mu_2, \ldots, \mu_b, \mu_{b+1}];$$  \hspace{1cm} (113a)

let $\tilde{\lambda}(t)$ denote the $nx(b+1)$ matrix

$$\tilde{\lambda}(t) = [\lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t), \lambda_{b+1}(t)];$$  \hspace{1cm} (113b)

Clearly, $\tilde{\mu} = [I, 0]$, where $I$ denotes the identity matrix of order $b$. 
and let \( k \) denote the \((b+1)\)-vector

\[
    k = [k_1, k_2, \ldots, k_b, k_{b+1}]^T.
\]  

(113c)

If the method of particular solution is employed (Refs. 15-17), the general solution of Eqs. (105a) and (105e) can be written in the form

\[
    \mu = \tilde{\mu} k, \tag{114a}
\]

\[
    \lambda(t) = \tilde{\lambda}(t) k, \tag{114b}
\]

with the following understanding: the components of the vector \( k \) must satisfy

the normalization condition

\[
    U^T k = 1, \tag{115}
\]

where

\[
    U = [1, 1, \ldots, 1, 1]^T \tag{116}
\]

denotes the \((b+1)\)-vector whose components are all equal to one.

Next, we combine Eqs. (105b), (105c), (105d) with Eqs. (114) and obtain

the relations

\[
    y = -F_u - \Phi_u \tilde{\lambda} k, \tag{117a}
\]

\[
    z = -\left[ \int_0^1 F_{\pi} dt + (h_\pi)_0 + (g_\pi)_1 \right] - \left[ \int_0^1 \phi_\pi \tilde{\lambda} dt + (\psi_\pi)_1 \tilde{\mu} \right] k - (\omega_\pi)_0 \sigma, \tag{117b}
\]

\[
    w = -(h_\chi)_0 + (\tilde{\chi})_0 k - (\omega_\chi)_0 \sigma. \tag{117c}
\]

These relations show that \( y(t), z, w \) depend only on the parameters \( k, \sigma \).
Finally, upon combining (104) and (117), we obtain the following quadratic function of \( k, \sigma \):

\[
I_D = (1/2)k^T M_k + (1/2)\sigma^T N_\sigma + k^T L_\sigma + D^T_k + E^T \sigma + (1/2)H,
\]

where

\[
U^T k - 1 = 0.
\]

Here, the matrices \( M, N, L \), the vectors \( D, E \), and the scalar \( H \) are known.

They are defined by

\[
M = \int_0^1 (\phi_u \tilde{\lambda})^T (\phi_u \tilde{\lambda}) dt + (\tilde{\lambda}^T \tilde{\lambda})_0
\]

\[
+ \int_0^1 \phi_u \tilde{\lambda} dt + (\psi_\pi \tilde{\mu})^T \int_0^1 \phi_u \tilde{\lambda} dt + (\psi_\pi \tilde{\mu}) \right]
\]

\[
N = (\omega_\pi^T \omega_\pi + \omega_\pi^T \omega_\pi)_0,
\]

\[
L = \int_0^1 \phi_\pi \tilde{\lambda} dt + (\psi_\pi \tilde{\mu})^T (\omega_\pi)_0 - (\tilde{\lambda}^T \omega_\pi)_0,
\]

\[
D = \int_0^1 \phi_\pi \tilde{\lambda} dt + (\psi_\pi \tilde{\mu})^T \int_0^1 F_\pi dt + (h_\pi)_0 + (g_\pi)_1
\]

\[
+ \int_0^1 (\phi_u \tilde{\lambda})^T F_u dt - (\tilde{\lambda}^T h_\pi)_0,
\]

\[
E = (\omega_\pi^T)_0 \int_0^1 F_\pi dt + (h_\pi)_0 + (g_\pi)_1 \right]
\]

\[
H = \int_0^1 F_\pi dt + (h_\pi)_0 + (g_\pi)_1 \right]
\]

\[
+ \int_0^1 F_u^T F_u dt + (h_\pi^T h_\pi)_0.
\]
Because of the duality property, the parameters $k$, $\sigma$ can be obtained by minimizing (118), subject to (119). Clearly, this auxiliary minimization problem is a mathematical programming problem.

Let $\beta$ denote a scalar Lagrange multiplier associated with the constraint (119). Let $G_D$ denote the augmented function

$$G_D = (1/2)k^T Mk + (1/2)\sigma^T N\sigma + k^T L\sigma + D^T k + E^T \sigma + (1/2)H + \beta(U^T k - 1).$$  \hspace{1cm} (121)

With this understanding, the first-order optimality conditions of the auxiliary minimization problem take the form

$$G_{Dk} = 0, \quad G_{D\sigma} = 0. \hspace{1cm} (122)$$

Hence, the values of $k$, $\sigma$, $\beta$ are determined by solving the following linear algebraic system:

$$Mk + L\sigma + U\beta + D = 0, \hspace{1cm} (123a)$$

$$L^T k + N\sigma + E = 0, \hspace{1cm} (123b)$$

$$U^T k - 1 = 0, \hspace{1cm} (123c)$$

whose dimension is $a+b+2$. Once $k$, $\sigma$, $\beta$ are known, the multipliers $\mu$ and $\lambda(t)$ are determined with Eqs. (114). Then, $\Delta u(t)/\alpha$, $\Delta v/\alpha$, $\Delta x(0)/\alpha$ are obtained with Eqs. (102b), (102c), (102d). Finally, $\Delta x(t)/\alpha$ is determined by forward integration of (96a).

We note that, except for the determination of the stepsize, the gradient iteration is completed. The stepsize $\alpha$ can be determined a posteriori using some suitable search procedure.
14. General Problem, Restoration Phase, Primal Formulation

The restoration phase of the sequential gradient-restoration algorithm involves one or more iterations and is designed to force constraint satisfaction to a predetermined accuracy. The restoration phase is terminated whenever Ineq. (88a) is satisfied.

Each restorative iteration is started with nominal functions $x(t)$, $u(t)$, $v(t)$, $\pi$ violating at least one of Eqs. (77), thereby violating Ineq. (88a). It leads from the nominal functions $x(t)$, $u(t)$, $v(t)$, $\pi$ to the varied functions $\bar{x}(t)$, $\bar{u}(t)$, $\bar{v}(t)$, $\bar{\pi}$ through the displacements $\Delta x(t)$, $\Delta u(t)$, $\Delta v(t)$, $\Delta \pi$. By definition,

\begin{align}
\bar{x}(t) &= x(t) + \Delta x(t), & (124a) \\
\bar{u}(t) &= u(t) + \Delta u(t), & (124b) \\
\bar{v}(t) &= v(t) + \Delta v(t), & (124c) \\
\bar{\pi} &= \pi + \Delta \pi. & (124d)
\end{align}

The displacements $\Delta x(t)$, $\Delta u(t)$, $\Delta v(t)$, $\Delta \pi$ are computed by solving the following auxiliary minimization problem.

**Problem (RP).** Minimize the norm squared of the vectors $\Delta u(t)$, $\Delta \pi$, $\Delta x(0)$, with respect to the vectors $\Delta x(t)$, $\Delta u(t)$, $\Delta v(t)$, $\Delta \pi$ which satisfy the linearized form of the constraints (77). Therefore, we minimize the functional

\begin{equation}
I_p = (1/2\alpha) \int_0^1 \Delta u^T \Delta u dt + \Delta \pi^T \Delta \pi + (\Delta x^T \Delta x)_0,
\end{equation}

with respect to the vectors $\Delta x(t)$, $\Delta u(t)$, $\Delta v(t)$, $\Delta \pi$ which satisfy the constraints

\begin{align}
\dot{\Delta x} + \phi_x^T \Delta x + \phi_u^T \Delta u + \phi_v^T \Delta v + \phi_{\pi}^T \Delta \pi + \alpha(\dot{x} + \phi) &= 0, & 0 \leq t \leq 1, & (126a)
\end{align}
\[ S_x^T \Delta x + S_u^T \Delta u + S_v^T \Delta v + S_{\pi}^T \Delta \pi + \alpha S = 0, \quad 0 \leq t \leq 1, \quad (126b) \]
\[ (\alpha_x^T \Delta x + \alpha_u^T \Delta u + \alpha \omega)_0 = 0, \quad (126c) \]
\[ (\psi_x^T \Delta x + \psi_u^T \Delta u + \alpha \psi)_1 = 0. \quad (126d) \]

The symbol \( \alpha \) in (125)-(126) denotes the restoration stepsize, \( 0 < \alpha \leq 1 \).

**Alternative Form.** Under the assumption that the matrix \( S_v \) is nonsingular, Eq. (126b) yields the following solution for the dependent control displacement:

\[ \Delta v = -(S_v^{-1})^T (S_x^T \Delta x + S_u^T \Delta u + S_{\pi}^T \Delta \pi + \alpha S), \quad 0 \leq t \leq 1. \quad (127) \]

Let the following definitions be introduced:

\[ \hat{\phi} = \phi - (S_v^{-1} \phi_v)^T S \quad (128) \]

and

\[ \phi_x = \phi_x - S_x S_v^{-1} \phi_v, \quad (129a) \]
\[ \phi_u = \phi_u - S_u S_v^{-1} \phi_v, \quad (129b) \]
\[ \phi_{\pi} = \phi_{\pi} - S_{\pi} S_v^{-1} \phi_v. \quad (129c) \]

As a consequence, upon eliminating \( \Delta v \) from Eqs. (126) and upon recalling the definitions (128)-(129), Problem (RP) is reformulated as follows: Minimize the functional

\[ I_p = (1/2\alpha) \left\{ \int_0^1 \Delta u^T \Delta u dt + \Delta \pi^T \Delta \pi + (\Delta x^T \Delta x)_0 \right\}, \quad (130) \]

with respect to the vectors \( \Delta x(t), \Delta u(t), \Delta \pi \) which satisfy the constraints

\[ \Delta \dot{x} + \phi_x^T \Delta x + \phi_u^T \Delta u + \phi_{\pi}^T \Delta \pi + \alpha (\dot{x} + \hat{\phi}) = 0, \quad 0 \leq t \leq 1, \quad (131a) \]
\[(\omega_\Delta^T x + \omega_\perp^T \Delta t + \alpha \omega)_0 = 0, \quad (131b)\]
\[(\psi_\Delta x + \psi_\perp^T \Delta t + \alpha \psi)_1 = 0. \quad (131c)\]

From calculus of variations, it is known that Problem (RP) is of the Bolza type. It can be recast as that of minimizing the augmented functional

\[J_P = I_P + L_P, \quad (132)\]

subject to (131), where \(L_P\) denotes the Lagrangian functional

\[L_P = \int_0^1 \lambda^T [\Delta \dot{x} + \phi_\Delta^T \Delta u + \phi_\perp^T \Delta t + \alpha (\dot{x} + \dot{\phi})] dt + \sigma^T (\omega_\Delta x + \omega_\perp^T \Delta t + \alpha \omega)_0 + \mu^T (\psi_\Delta x + \psi_\perp^T \Delta t + \alpha \psi)_1. \quad (133)\]

In Eq. (133), \(\lambda(t)\) denotes an \(n\)-vector Lagrange multiplier, \(\sigma\) an \(a\)-vector Lagrange multiplier, and \(\mu\) a \(b\)-vector Lagrange multiplier.

For Problem (RP), let the Hamiltonian \(H\) and the augmented functions \(K, N, G\) be defined by

\[H = (1/2 \alpha) \Delta u^T \Delta u + \lambda^T [\phi_\Delta^T \Delta x + \phi_\Delta^T \Delta u + \phi_\perp^T \Delta t + \alpha (\ddot{x} + \ddot{\phi})], \quad (134a)\]
\[K = (1/2 \alpha) \Delta t^T \Delta t, \quad (134b)\]
\[N = (1/2 \alpha) \Delta x^T \Delta x + \sigma^T (\omega_\Delta x + \omega_\perp^T \Delta t + \alpha \omega), \quad (134c)\]
\[G = \mu^T (\psi_\Delta^T \Delta x + \psi_\perp^T \Delta t + \alpha \psi). \quad (134d)\]

Then, the first-order optimality conditions for Problem (RP) take the form

\[\dot{\lambda} = H_{\Delta x}^T = 0, \quad 0 \leq t \leq 1, \quad (135a)\]
\[H_{\Delta u} = 0, \quad 0 \leq t \leq 1, \quad (135b)\]
\[ \int_0^1 H_{\Delta t} \, dt + K_{\Delta t} + (N_{\Delta t})_0 + (G_{\Delta t})_1 = 0, \]  
(135c)

\[ (-\lambda + N_{\Delta x})_0 = 0, \]  
(135d)

\[ (\lambda + G_{\Delta x})_1 = 0. \]  
(135e)

On account of Eqs. (134), the explicit form of Eqs. (135) is the following:

\[ \dot{\lambda} - \Phi_x \lambda = 0, \quad 0 \leq t \leq 1, \]  
(136a)

\[ \Phi_u \lambda + \Delta u / \alpha = 0, \quad 0 \leq t \leq 1, \]  
(136b)

\[ \int_0^1 \Phi_{\Delta t} \lambda \, dt + (\omega_{\Delta t})_0 + (\psi_{\Delta t})_1 + \Delta \tau / \alpha = 0, \]  
(136c)

\[ (-\lambda + \omega_x \sigma + \Delta x / \alpha)_0 = 0, \]  
(136d)

\[ (\lambda + \psi_x \mu)_1 = 0. \]  
(136e)

Summarizing, we seek the functions \( \Delta x(t), \Delta u(t), \Delta \tau \) and the multipliers \( \lambda(t), \sigma, \mu \) such that the feasibility equations (131) and the optimality conditions (136) are satisfied. After Eqs. (131) and (136) are solved, \( \Delta v(t) \) is obtained with (127).
15. **General Problem, Restoration Phase, Dual Formulation**

Let \( y(t), z, w \) denote the vectors defined by

\[
\Phi_u \lambda + y = 0, \quad \text{(137a)}
\]

\[
\int_0^1 \Phi_\pi \lambda \, dt + (\omega_\pi \sigma)_0 + (\psi_\pi \mu)_1 + z = 0, \quad \text{(137b)}
\]

\[
(-\lambda + \omega X \sigma)_0 + w = 0. \quad \text{(137c)}
\]

It is interesting to note that the vectors \( \lambda(t), \sigma, \mu \) and \( y(t), z, w \) can also be obtained by solving the following auxiliary minimization problem.

**Problem (RD).** Minimize the functional

\[
I_0 = (1/2) \int_0^1 y^T y \, dt + z^T z + w^T w
\]

\[
- [\int_0^1 \lambda^T (\dot{\lambda} + \hat{\lambda}) \, dt + (\sigma^T \omega)_0 + (\mu^T \psi)_1], \quad \text{(138)}
\]

with respect to the vectors \( \lambda(t), \sigma, \mu \) and \( y(t), z, w \) which satisfy the constraints

\[
\dot{\lambda} - \Phi_X \lambda = 0, \quad 0 \leq t \leq 1, \quad \text{(139a)}
\]

\[
\Phi_u \lambda + y = 0, \quad 0 \leq t \leq 1, \quad \text{(139b)}
\]

\[
\int_0^1 \Phi_\pi \lambda \, dt + (\omega_\pi \sigma)_0 + (\psi_\pi \mu)_1 + z = 0, \quad \text{(139c)}
\]

\[
(-\lambda + \omega X \sigma)_0 + w = 0, \quad \text{(139d)}
\]

\[
(\lambda + \psi_X \mu)_1 = 0. \quad \text{(139e)}
\]

From calculus of variations, it is known that Problem (RD) is of the Bolza type. It can be recast as that of minimizing the augmented functional
\[ J_D = I_D + L_D, \quad (140) \]

subject to (139), where \( L_D \) denotes the Lagrangian functional

\[
L_D = \int_0^1 [\lambda^T X - \lambda^T \phi_X \lambda] dt - \lambda^T (\psi X \mu) 1 - z^T \lambda - w^T (- \lambda + \psi X \mu) 1. \quad (141)
\]

In Eq. (141), \( \lambda(t) \) denotes an \( n \)-vector Lagrange multiplier, \( y(t) \) an \( m \)-vector Lagrange multiplier, \( z \) a \( p \)-vector Lagrange multiplier, \( w \) an \( n \)-vector Lagrange multiplier, and \( \nu \) an \( n \)-vector Lagrange multiplier.

For Problem (RD), let the Hamiltonian \( H \) and the augmented functions \( K, N, G \) be defined by

\[
H = (1/2)y^T y - \lambda^T (\dot{\lambda} + \phi) - \lambda^T \phi_X \lambda - y^T (\phi_X \lambda + y) - z^T \phi_X \lambda, \quad (142a)
\]
\[
K = (1/2)(z^T z + w^T w) - z^T \lambda - w^T \lambda, \quad (142b)
\]
\[
N = -\sigma^T w - z^T \psi \sigma - w^T (- \lambda + \psi \sigma), \quad (142c)
\]
\[
G = -\mu^T \psi - z^T \psi \mu - \nu^T (\lambda + \psi \mu). \quad (142d)
\]

Then, the first-order optimality conditions of Problem (RD) take the form

\[
\dot{\lambda} + H = 0, \quad 0 \leq t \leq 1, \quad (143a)
\]
\[
H_y = 0, \quad 0 \leq t \leq 1, \quad (143b)
\]
\[
(N_y)_0 = 0, \quad (143c)
\]
\[
(G_y)_1 = 0, \quad (143d)
\]
\[ K_z = 0, \quad (143e) \]
\[ K_w = 0, \quad (143f) \]
\[ (-\lambda_* + N_x)_{0} = 0, \quad (143g) \]
\[ (\lambda_* + G_{\lambda})_{1} = 0. \quad (143h) \]

On account of Eqs. (142), the explicit form of Eqs. (143) is the following:

\[ \dot{x}_* + \phi_T \lambda_* + \phi_T y_* + \phi_T z_* + (\dot{x} + \hat{o}) = 0, \quad 0 \leq t \leq 1, \quad (144a) \]
\[ y - y_* = 0, \quad 0 \leq t \leq 1, \quad (144b) \]
\[ (\omega_T w_* + \omega_T z_* + \omega)_{0} = 0, \quad (144c) \]
\[ (\psi_T \nu_* + \psi_T z_* + \psi)_{1} = 0, \quad (144d) \]
\[ z - z_* = 0, \quad (144e) \]
\[ w - w_* = 0, \quad (144f) \]
\[ (\lambda_* - w_*)_{0} = 0, \quad (144g) \]
\[ (\lambda_* - \nu_*)_{1} = 0. \quad (144h) \]

Let the following substitutions be employed:

\[ \lambda_* = \Delta x(t)/\alpha, \quad 0 \leq t \leq 1, \quad (145a) \]
\[ y_* = \Delta u(t)/\alpha, \quad 0 \leq t \leq 1, \quad (145b) \]
\[ z_* = \Delta \tau/\alpha, \quad (145c) \]
\[ w_* = \Delta x(0)/\alpha, \quad (145d) \]

\[ v_* = \Delta x(1)/\alpha. \quad (145e) \]

Then, one can readily verify that Eqs. (144) and (139) of Problem (RD) reduce to Eqs. (131) and (136) of Problem (RP). Clearly, after the transformation (145) is applied, the solution of Problem (RD) yields the solution of Problem (RP) and vice versa. This means that the multipliers \( \lambda(t), \sigma, \mu \) associated with the restoration phase of SGRA are endowed with a duality property: They also minimize the quadratic functional (138), subject to (139), for given state, control, and parameter.
16. General Problem, Restoration Phase, Computational Implications

In this section, we exploit the previously established duality property and show that the execution of a restorative iteration can be reduced to solving a mathematical programming problem involving a finite number of parameters as unknowns. Hence, the algorithmic efficiency of the restoration phase of SGRA can be enhanced.

First, we consider Eqs. (139a) and (139e). We observe that, if $\mu$ is assigned, $\lambda(1)$ can be computed with (139e) and $\lambda(t)$ can be computed by backward integration of (139a). Next, we execute $b$ backward integrations, using Eqs. (139a) and (139e) in combination with the following choices for the multiplier $\mu$:

$$\mu_1 = \delta_1, \quad \mu_2 = \delta_2, \ldots, \quad \mu_b = \delta_b.$$  \hspace{1cm} (146)

In (146), $\delta_1, \delta_2, \ldots, \delta_b$ denote the vectors corresponding to the columns of the identity matrix of order $b$.

Let $\lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t)$ denote the solutions of Eqs. (139a) and (139e), corresponding to the choices (146) for the multiplier $\mu$. Let $\tilde{\mu}$ denote the $b \times b$ matrix\footnote{Clearly, $\tilde{\mu} = I$, where $I$ denotes the identity matrix of order $b$.}

$$\tilde{\mu} = [\mu_1, \mu_2, \ldots, \mu_b];$$  \hspace{1cm} (147a)

Let $\tilde{X}(t)$ denote the $nxb$ matrix

$$\tilde{X}(t) = [\lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t)];$$  \hspace{1cm} (147b)

and let $k$ denote the $b$-vector

$$k = [k_1, k_2, \ldots, k_b]^T.$$  \hspace{1cm} (147c)
If the method of complementary functions is employed, the general solution of Eqs. (139a) and (139e) can be written in the form

\[ \mu = \tilde{\mu}k, \]  
\[ \lambda(t) = \tilde{\lambda}(t)k. \]  

Next, we combine Eqs. (139b), (139c), (139d) with Eqs. (148) and obtain the relations

\[ y = -\phi_u \tilde{\lambda}_k, \]  
\[ z = -\int_0^1 \phi_u \tilde{\lambda}_k dt + (\psi_\pi)_1 \tilde{\mu} = (\omega_\pi)_0 \sigma, \]  
\[ w = (\tilde{\lambda})_0 k - (\omega_x)_0 \sigma. \]

These relations show that \( y(t), z, w \) depend only on the parameters \( k, \sigma \).

Finally, upon combining (138) and (149), we obtain the following quadratic function of \( k, \sigma \):

\[ I_D = (1/2)k^T M k + (1/2)\sigma^T N \sigma + k^T L \sigma + D^T k + E^T \sigma. \]

Here, the matrices \( M, N, L \) and the vectors \( D, E \) are known. They are defined by

\[ M = \int_0^1 (\phi_u \tilde{\lambda})^T (\phi_u \tilde{\lambda}) dt + (\tilde{\lambda}^T \lambda)_0 \]  
\[ + \left[ \int_0^1 \phi_u \tilde{\lambda}_k dt + (\psi_\pi)_1 \tilde{\mu} \right]^T \left[ \int_0^1 \phi_u \tilde{\lambda}_k dt + (\psi_\pi)_1 \tilde{\mu} \right], \]  
\[ N = (\omega_x^T \omega_x + \omega^T \omega)_0, \]  
\[ L = \left[ \int_0^1 \phi_u \tilde{\lambda}_k dt + (\psi_\pi)_1 \tilde{\mu} \right]^T (\omega_\pi)_0 - (\tilde{\lambda}^T \omega_x)_0. \]
\[ D = -\int_0^1 \lambda^T(\dot{x} + \dot{\psi}) dt - (\tilde{\mu}^T \psi)_1, \]  
\[ E = -\omega_0. \]  

Because of the duality property, the parameters \( k, \sigma \) can be obtained by minimizing (150). Clearly, this auxiliary minimization problem is a mathematical programming problem, whose first-order optimality conditions take the form

\[ I_Dk = 0, \quad I_D\sigma = 0. \]  

Hence, the values of \( k, \sigma \) are determined by solving the following linear algebraic system:

\[ Mk + L\sigma + D = 0, \]  
\[ L^T k + N\sigma + E = 0, \]  

whose dimension is \( a+b \). Once \( k, \sigma \) are known, the multipliers \( \mu \) and \( \lambda(t) \) are determined with Eqs. (148). Then, \( \Delta u(t)/\alpha, \Delta x/\alpha, \Delta x(0)/\alpha \) are obtained with Eqs. (136b), (136c), (136d). Finally, \( \Delta x(t)/\alpha \) is determined by forward integration of (131a).

We note that, except for the determination of the stepsize, the restorative iteration is completed. The stepsize \( \alpha \) can be determined a posteriori using some suitable search procedure.
17. **General Problem, Dual Sequential Gradient-Restoration Algorithm**

The sequential-gradient restoration algorithm involves a sequence of two-phase cycles, each cycle including a gradient phase and a restoration phase. In a complete gradient-restoration cycle, the value of the functional is decreased, while the constraints are satisfied to a predetermined accuracy; in the gradient phase, the value of the augmented functional is decreased, while avoiding excessive constraint violation; in the restoration phase, the constraint error is decreased, while avoiding excessive change in the value of the functional.

It must be noted that, while the gradient phase involves a single iteration, the restoration phase might involve several iterations. The decision of whether to execute a gradient iteration or a restorative iteration is based on the measurement of a single scalar quantity, the constraint error $P$, given by Eq. (87a). If the constraint error satisfies Ineq. (88a), a gradient iteration is executed; if the constraint error violates Ineq. (88a), a restorative iteration is executed.

For both gradient iterations and restorative iterations, the following terminology is employed: $x(t)$, $u(t)$, $\pi$ denote the nominal functions; $\tilde{x}(t)$, $\tilde{u}(t)$, $\tilde{\pi}$ denote the varied functions; $\Delta x(t)$, $\Delta u(t)$, $\Delta \pi$ denote the perturbations leading from the nominal functions to the varied functions; and $A(t)$, $B(t)$, $C$ denote the perturbations per unit stepsize $\alpha$. Then, the following relations hold:

\[
\tilde{x}(t) = x(t) + \Delta x(t) = x(t) + \alpha A(t), \quad (154a)
\]

\[
\tilde{u}(t) = u(t) + \Delta u(t) = u(t) + \alpha B(t), \quad (154b)
\]

\[
\tilde{\pi} = \pi + \Delta \pi = \pi + \alpha C. \quad (154c)
\]
Thus, each iteration involves the following operations: (i) the determination of the functions $A(t), B(t), C$; and (ii) the determination of the stepsize $\alpha$.

Depending on whether the primal formulation is used or the dual formulation is used, one obtains a primal sequential gradient-restoration algorithm (PSGRA) and a dual sequential gradient-restoration algorithm (DSGRA). Since PSGRA has been discussed in Refs. 1-3, we restrict our attention to DSGRA.

Gradient Iteration. Its objective is to reduce the augmented functional $J$, while the constraints are satisfied to first order.

Step 1. Assume nominal functions $x(t), u(t), \pi$ which satisfy the constraints (77) within the predetermined accuracy (88a).\(^6\)

Step 2. For the nominal functions, compute the vectors $F_x, F_u, F_\pi$ and the matrices $\phi_x, \phi_u, \phi_\pi$ along the interval of integration; compute the vectors $h_x, h_\pi$ and the matrices $\omega_x, \omega_\pi$ at the initial point; compute the vectors $g_x, g_\pi$ and the matrices $\psi_x, \psi_\pi$ at the final point.

Step 3. Execute $b+1$ backward integrations of the multiplier equations, using (105a) and (105e) in combination with the choices (112) for the multiplier $\mu$. Determine the particular solutions $\lambda_1(t), \lambda_2(t), \ldots, \lambda_b(t), \lambda_{b+1}(t)$. Determine the matrices $\tilde{\mu}$ and $\tilde{\lambda}(t)$, given by (113a) and (113b).

Step 4. Using the matrices $\tilde{\mu}$ and $\tilde{\lambda}(t)$ of Step 3, compute the matrices $M, N, L$, the vectors $D, E$, and the scalar $H$, given by Eqs. (120). Determine the vectors $k, \sigma$ and the scalar $\beta$ by solving the linear algebraic system (123).

Step 5. With the vector $k$ known, compute the multipliers $\mu$ and $\lambda(t)$ using Eqs. (114). Alternatively, compute the multiplier $\mu$ with (114a) and the multiplier $\lambda(t)$ by backward integration of the multiplier equations, using (105a) and (105e) in combination with (114a).

\(^6\)For given $x(t), u(t), \pi$, the dependent control $v(t)$ is computed in such a way that Eq. (77b) is satisfied. For instance, $v(t)$ can be computed using the modified quasilinearization algorithm.
**Step 6.** Determine the control change per unit stepsize \( B(t) \), the parameter change per unit stepsize \( C \), and the initial state change per unit stepsize \( A(0) \) using the relations

\[
B = -(F_u + \Phi_u \lambda), \quad 0 \leq t \leq 1, \quad (155a)
\]

\[
C = - \int_0^1 (F_\pi + \Phi_\pi \lambda) dt - (h_\pi + \omega_\pi \sigma)_0 - (q_\pi + \psi_\pi \mu)_1, \quad (155b)
\]

\[
(A)_0 = -(-\lambda + h_x + \omega_x \sigma)_0. \quad (155c)
\]

Determine the state change per unit stepsize \( A(t) \) by forward integration of the linearized state equation, using the relation

\[
\dot{A} + \Phi_x^T A + \Phi_u^T B + \Phi_\pi^T C = 0, \quad 0 \leq t \leq 1, \quad (156)
\]

in combination with Eqs. (155).

**Step 7.** With the functions \( A(t) \), \( B(t) \), \( C \) known, the one-parameter family of varied functions (154) can be formed. For this family, the functionals \( I, J, P \) take the following form:

\[
\tilde{I} = \tilde{I}(\alpha), \quad \tilde{J} = \tilde{J}(\alpha), \quad \tilde{P} = \tilde{P}(\alpha). \quad (157)
\]

Then, the stepsize \( \alpha \) is computed by a one-dimensional search on the function \( \tilde{J}(\alpha) \) until the following relations are satisfied:

\[
\tilde{J}(\alpha) < \tilde{J}(0), \quad (158a)
\]

\[
\tilde{P}(\alpha) \leq P_*, \quad (158b)
\]

where \( P_* \) is a preselected number, not necessarily small.

The simplest way of ensuring satisfaction of (158) is to employ a bisection process, starting from the reference stepsize \( \alpha = \alpha_0 \). In turn, the reference
stepsize $\alpha_0$ can be obtained by the combination of a scanning process and a cubic interpolation process. With the scanning process, one brackets the minimum point of the function $\tilde{J}(\alpha)$. With the cubic interpolation process, either single-step or multi-step, one obtains an approximation to the reference stepsize $\alpha_0$. This is the stepsize which yields the minimum of the cubic approximation to $\tilde{J}(\alpha)$.

The details of the one-dimensional search can be found in Refs. 1-3 and related publications. They are omitted here, for the sake of brevity.

**Step 8.** Once the stepsize $\alpha$ is known, compute the varied functions $\tilde{x}(t)$, $\tilde{u}(t)$, $\tilde{w}$ with Eqs. (154). In this way, the gradient iteration is completed.

**Restorative Iteration.** Its objective is to reduce the constraint error $P$, while the constraints are satisfied to first order and the norm squared of the vectors $\Delta u(t)$, $\Delta w$, $\Delta x(0)$ is minimized.

**Step 1.** Assume nominal functions $x(t)$, $u(t)$, $w$ which satisfy the constraint (77b), but violate at least one of the constraints (77a), (77c), (77d).\footnote{For given $x(t)$, $u(t)$, $w$, the dependent control $v(t)$ is computed in such a way that Eq. (77b) is satisfied. For instance, $v(t)$ can be computed using the modified quasilinearization algorithm.}

**Step 2.** For the nominal functions, compute the vector $\dot{x} + \dot{\phi}$ and the matrices $\Phi_x$, $\Phi_u$, $\Phi_w$ along the interval of integration; compute the vector $\omega$ and the matrices $\omega_x$, $\omega_w$ at the initial point; compute the vector $\psi$ and the matrices $\psi_x$, $\psi_w$ at the final point.

**Step 3.** Execute $b$ backward integrations of the multiplier equations, using (139a) and (139e) in combination with the choices (146) for the multiplier $\mu$. Determine the complementary solutions $\lambda_1(t)$, $\lambda_2(t)$, ..., $\lambda_b(t)$. Determine the matrices $\tilde{\mu}$ and $\tilde{\lambda}(t)$, given by (147a) and (147b).

**Step 4.** Using the matrices $\tilde{\mu}$ and $\tilde{\lambda}(t)$ of Step 3, compute the matrices $M$, $N$, $L$ and the vectors $D$, $E$, given by Eqs. (151). Determine the vectors $k$, $\sigma$.
by solving the linear algebraic system (153).

**Step 5.** With the vector \( k \) known, compute the multipliers \( \mu \) and \( \lambda(t) \) using Eqs. (148). Alternatively, compute the multiplier \( \mu \) with (148a) and the multiplier \( \lambda(t) \) by backward integration of the multiplier equations, using (139a) and (139e) in combination with (148a).

**Step 6.** Determine the control change per unit stepsize \( B(t) \), the parameter change per unit stepsize \( C \), and the initial state change per unit stepsize \( A(0) \) using the relations

\[
B = -\phi_u \lambda, \quad 0 \leq t \leq 1, \quad (159a)
\]
\[
C = -\int_0^1 \phi_\pi \lambda dt - (\omega_\pi \sigma)_0 - (\psi_\pi \mu)_1, \quad (159b)
\]
\[
(A)_0 = -(-\lambda + \omega_x \sigma)_0. \quad (159c)
\]

Determine the state change per unit stepsize \( A(t) \) by forward integration of the linearized state equation, using the relation

\[
\dot{A} + \phi_x^T A + \phi_u^T B + \phi_\pi^T C + (\dot{x} + \hat{\theta}) = 0, \quad 0 \leq t \leq 1, \quad (160)
\]

in combination with Eqs. (159).

**Step 7.** With the functions \( A(t), B(t), C \) known, the one-parameter family of varied functions (154) can be formed. For this family, the functional \( P \) takes the following form:

\[
\tilde{P} = \tilde{P}(\alpha). \quad (161)
\]

Then, the stepsize \( \alpha \) is computed by a one-dimensional search on the function (161) until the following relation is satisfied:

\[
\tilde{P}(\alpha) < \tilde{P}(0). \quad (162)
\]
The simplest way of ensuring satisfaction of (162) is to employ a bisection process, starting from the reference stepsize $\alpha = \alpha_0$. Here, the correct reference stepsize is $\alpha_0 = 1$, in that it yields one-step restoration if the constraints (77) are linear.

**Step B.** Once the stepsize $\alpha$ is known, compute the varied functions $\bar{x}(t), \bar{u}(t), \bar{\pi}$ with Eqs. (154). In this way, the restorative iteration is completed.

**Gradient Phase.** The gradient phase includes a single gradient iteration. Hence, the gradient phase is the same as the gradient iteration discussed previously.

**Restoration Phase.** The restoration phase might include several restorative iterations. In each restorative iteration, the constraint error is reduced in accordance with Ineq. (162). The restoration phase is terminated whenever the constraint error reaches a level compatible with Ineq. (88a).

**Gradient-Restoration Cycle.** As stated before, a complete gradient-restoration cycle includes a gradient phase and a restoration phase. After a restoration phase is completed, one must verify whether the following inequality is satisfied:

$$I < \hat{I};$$  \hspace{1cm} (163)

Here, $I$ denotes the value of the functional (76) at the end of the present restoration phase and $\hat{I}$ denotes the value of the functional (76) at the end of the previous restoration phase. If Ineq. (163) is satisfied, one starts the next cycle of the sequential gradient-restoration algorithm. If Ineq. (163) is violated, one returns to the previous gradient phase and reduces the gradient stepsize (using a bisection process) until, after restoration, the functional $I$ finally decreases.
**Starting Condition.** The present algorithm must be started with nominal functions $x(t), u(t), \pi$ which satisfy the constraint (77b), but not necessarily the constraints (77a), (77c), (77d). If the nominal functions violate Ineq. (88a), the algorithm starts with a restoration phase; hence, the first cycle is a half cycle involving only a restoration phase. If the nominal functions satisfy Ineq. (88a), the algorithm starts with a gradient phase; hence, the first cycle is a complete cycle, involving both a gradient phase and a restoration phase.

**Stopping Condition.** The present algorithm is terminated whenever Ineqs. (88a) and (88b) are satisfied simultaneously. Note that Ineq. (88a) is verified at the end of a restoration phase/beginning of a gradient phase. On the other hand, Ineq. (88b) must be verified at the beginning of a gradient phase, after the multipliers $\lambda(t), \mu$ are computed and before the search for the gradient stepsize is executed.
18. Comments

In Sections 2-17, primal-dual properties were derived employing only first-order conditions. It can be verified readily that both the Legendre-Clebsch condition and the Weierstrass condition (minimum principle) are satisfied for all of the auxiliary minimization problems studied in connection with both the primal formulation and the dual formulation. It can also be verified that the solution of each auxiliary minimization problem is unique. These topics are omitted for the sake of brevity.

It is clear that some simple relations hold between the values of the functionals associated with the primal formulation and the values of the functionals associated with the dual formulation. These relations are stated below without proof.

For the gradient phase, the functionals $I_p$ and $I_D$, associated with Problems (GP) and (GD), satisfy the relation

$$ I_p + 2\alpha I_D = 0, \tag{164} $$

where $\alpha$ is the stepsize of the gradient phase.

For the restoration phase, the functionals $I_p$ and $I_D$, associated with Problems (RP) and (RD), satisfy the relation

$$ I_p + \alpha I_D = 0, \tag{165} $$

where $\alpha$ is the stepsize of the restoration phase.
19. **Experimental Conditions**

In order to evaluate the theory, 12 examples were solved. The sequential gradient-restoration algorithms associated with Problems (P1) and (P2) were programmed in FORTRAN IV; a FORTRAN G1 compiler was used; the numerical results were obtained in double-precision arithmetic using both the primal formulation (PSGRA) and the dual formulation (DSGRA).

Computations were performed at Rice University using an NAS-AS-9000 computer. For each example, the interval of integration was divided into 100 steps. The differential equations were integrated using Hamming's modified predictor-corrector method with a special Runge-Kutta starting procedure. The definite integrals I, J, P, Q were computed using a modified Simpson's rule.

**Convergence Conditions.** The parameters $\varepsilon_1$, $\varepsilon_2$ appearing in Ineqs. (9) and (88) were set at the levels

\[
\varepsilon_1 = E-08, \quad (166a)
\]

\[
\varepsilon_2 = E-04. \quad (166b)
\]

The tolerance level (166a) characterizes the convergence condition of the restoration phase; the tolerance levels (166a) and (166b), employed in combination, characterize the convergence conditions of the algorithm as a whole.

**Gradient Stepsize.** The search technique for the gradient stepsize $\alpha$ uses a scanning process, followed by a multi-step cubic interpolation process, followed by a bisection process. The multi-step cubic interpolation process was stopped whenever the following inequality is satisfied:

\[
|\tilde{J}_\alpha(\alpha)/\tilde{J}_\alpha(0)| \leq E-03. \quad (167a)
\]
The number of Hermitian search steps was subject to the upper bound

\[ N_S \leq 5. \]  \hspace{1cm} (167b)

The bisection process was started with the stepsize \( \alpha_0 \) which satisfies either Ineq. (167a) or \( N_S = 5 \). The bisection process was stopped whenever Ineqs. (70) or (158) are satisfied, with

\[ P_* = 10. \]  \hspace{1cm} (168)

**Restoration Stepwise.** The search technique for the restoration stepsize \( \alpha \) uses a bisection process, starting from the reference stepsize \( \alpha_0 = 1 \). The bisection process was stopped whenever Ineq. (74) or (162) is satisfied.
20. **Numerical Examples**

In order to evaluate the theory, 12 examples were solved. Examples 1-4 deal with the basic problem; Examples 4-8 deal with the general problem without nondifferential constraints; Examples 9-12 deal with the general problem with nondifferential constraints.

In the description of the examples, scalar notation is used. In particular, the symbols $x_i(t), i=1,2,..., n$, denote the components of the state vector; the symbols $u_i(t), i=1,2,..., m$, denote the components of the independent control vector; the symbols $v_i(t), i=1,2,..., c$, denote the components of the dependent control vector; the symbols $w_i(t), i=1,2,..., m+c$, denote the components of the total control vector; and the symbols $\pi_i, i=1,2,..., p$, denote the components of the parameter vector.

For all of the examples, a time normalization is used in order to simplify the numerical computations. Specifically, the actual time $\theta$ is replaced by the normalized time

$$ t = \theta/\tau, \quad (169) $$

which is defined in such a way that $t = 0$ at the initial point and $t = 1$ at the final point. The actual final time $\tau$, if it is free, is regarded as a component of the parameter vector $\pi$ being optimized. In this way, an optimal control problem with variable final time is converted into an optimal control problem with fixed final time.

Concerning the convergence history, the terminology is as follows: $N_c$ denotes the cycle number; $N_g$ is the number of gradient iterations per cycle; $N_r$ is the number of restorative iterations per cycle; $N$ is the total number of iterations; $P$ is the constraint error; $Q$ is the error in the optimality conditions; and $I$ the value of the functional being minimized.
Example 1. This example involves (i) a quadratic functional, (ii) two nonlinear differential equations, (iii) boundary conditions of the fixed endpoint type, and (iv) fixed final time $\tau = 1$:

$$I = \int_0^1 (1 + x_1^2 + x_2^2 + u_1^2)dt,$$  \hfill (170a)

$$\dot{x}_1 = u_1 - x_2^2, \quad \dot{x}_2 = u_1 - x_1x_2,$$  \hfill (170b)

$$x_1(0) = 0, \quad x_2(0) = 1,$$  \hfill (170c)

$$x_1(1) = 1, \quad x_2(1) = 2.$$  \hfill (170d)

The assumed nominal functions are

$$x_1(t) = t, \quad x_2(t) = 1 + t, \quad u_1(t) = 1.$$  \hfill (171)

The numerical results for both PSGRA and DSGRA are given in Tables 1-2. Convergence to the desired stopping conditions occurs in $N_c = 3$ cycles and $N = 7$ iterations, which include 2 gradient iterations and 5 restorative iterations. The CPU time is 0.71 sec for PSGRA and 0.48 sec for DSGRA.

Example 2. This example involves (i) a nonquadratic functional, (ii) two nonlinear differential equations, (iii) boundary conditions of the fixed endpoint type, and (iv) fixed final time $\tau = 1$:

$$I = \int_0^1 (-2\cos u_1)dt,$$  \hfill (172a)

$$\dot{x}_1 = 2\sin u_1 - 1, \quad \dot{x}_2 = x_1,$$  \hfill (172b)

$$x_1(0) = 0, \quad x_2(0) = 0,$$  \hfill (172c)

$$x_1(1) = 0, \quad x_2(1) = 0.3.$$  \hfill (172d)
The assumed nominal functions are

\[ x_1(t) = 0, \quad x_2(t) = 0.3t, \quad u_1(t) = 0. \]  \hspace{1cm} (173)

The numerical results for both PSGRA and DSGRA are given in Tables 3-4. Convergence to the desired stopping conditions occurs in \( N_c = 6 \) cycles and \( N = 13 \) iterations, which include 5 gradient iterations and 8 restorative iterations. The CPU time is 1.24 sec for PSGRA and 0.87 sec for DSGRA.

**Example 3.** This is a minimum time problem and involves (i) a linear functional, (ii) two nonlinear differential equations, (iii) boundary conditions of the fixed endpoint type, and (iv) free final time \( \tau \). After setting \( \pi_1 = \tau \), the problem is as follows:

\begin{align*}
I &= \pi_1, \quad \text{(174a)} \\
\dot{x}_1 &= \pi_1 u_1, \quad \dot{x}_2 = \pi_1 (u_1^2 - x_1^2), \quad \text{(174b)} \\
x_1(0) &= 0, \quad x_2(0) = 0, \quad \text{(174c)} \\
x_1(1) &= 1, \quad x_2(1) = 0. \quad \text{(174d)}
\end{align*}

The assumed nominal functions are

\[ x_1(t) = t, \quad x_2(t) = 0, \quad u_1(t) = 1, \quad \pi_1 = 1. \]  \hspace{1cm} (175)

The numerical results for both PSGRA and DSGRA are given in Tables 5-6. Convergence to the desired stopping conditions occurs in \( N_c = 3 \) cycles and \( N = 7 \) iterations, which include 2 gradient iterations and 5 restorative iterations. The CPU time is 0.73 sec for PSGRA and 0.51 sec for DSGRA.

**Example 4.** This is a minimum time problem and involves (i) a linear functional, (ii) three nonlinear differential equations, (iii) initial state
given, (iv) final state partly given and partly free, and (v) free final time $\tau$. After setting $\pi_1 = \tau$, the problem is as follows:

\begin{align*}
I &= \pi_1, \quad \text{(176a)} \\
\dot{x}_1 &= \pi_1 x_3 \cos u_1, \quad \dot{x}_2 = \pi_1 x_3 \sin u_1, \quad \dot{x}_3 = \pi_1 \sin u_1, \quad \text{(176b)} \\
x_1(0) &= 0, \quad x_2(0) = 0, \quad x_3(0) = 0, \quad \text{(176c)} \\
x_1(1) &= 1. \quad \text{(176d)}
\end{align*}

The assumed nominal functions are

\begin{equation}
x_1(t) = t, \quad x_2(t) = 0, \quad x_3(t) = 0, \quad u_1(t) = 1, \quad \pi_1 = 1. \quad \text{(177)}
\end{equation}

The numerical results for both PSGRA and DSGRA are given in Tables 7-8. Convergence to the desired stopping conditions occurs in $N_c = 4$ cycles and $N = 12$ iterations, which include 3 gradient iterations and 9 restorative iterations. The CPU time is 1.41 sec for PSGRA and 0.97 sec for DSGRA.

**Example 5.** This is a minimum time problem and involves (i) a linear functional, (ii) three nonlinear differential equations, (iii) a component of the initial state given and the remaining components subject to a nonlinear equation, (iv) final state partly given and partly free, and (v) free final time $\tau$. After setting $\pi_1 = \tau$, the problem is as follows:

\begin{align*}
I &= \pi_1, \quad \text{(178a)} \\
\dot{x}_1 &= \pi_1 u_1, \quad \dot{x}_2 = \pi_1 (x_1^2 - u_1^2), \quad \dot{x}_3 = \pi_1 (u_1 - x_2^2 + x_1), \quad \text{(178b)} \\
x_1(0) &= 0, \quad x_2(0) + x_3(0) = 1, \quad \text{(178c)} \\
x_2(1) &= 0, \quad x_3(1) = 2. \quad \text{(178d)}
\end{align*}
The assumed nominal functions are

\[ x_1(t) = 0, \quad x_2(t) = 0, \quad x_3(t) = 1 + t, \quad u_1(t) = 1, \quad \pi_1 = 1. \quad (179) \]

The numerical results for both PSGRA and DSGRA are given in Tables 9-10. Convergence to the desired stopping conditions occurs in \( N_c = 3 \) cycles and \( N = 9 \) iterations, which include 2 gradient iterations and 7 restorative iterations. The CPU time is 1.30 sec for PSGRA and 0.84 sec for DSGRA.

**Example 6.** This is a minimum time problem and involves (i) a linear functional, (ii) three nonlinear differential equations, (iii) a component of the initial state given and the remaining components subject to a nonlinear equation, (iv) a component of final state given and the remaining components subject to a nonlinear equation, and (v) free final time \( \tau \). After setting \( \pi_1 = \tau \), the problem is as follows:

\[ I = \pi_1, \quad \text{(180a)} \]

\[ \dot{x}_1 = \pi_1 u_1, \quad \dot{x}_2 = \pi_1 (x_1^2 - u_1^2), \quad \dot{x}_3 = \pi_1 (u_1 - x_2^2 + x_1), \quad \text{(180b)} \]

\[ x_1(0) = 0, \quad x_2^2(0) + x_3^2(0) = 1, \quad \text{(180c)} \]

\[ x_1(1)x_2(1) = 0, \quad x_3(1) = 2. \quad \text{(180d)} \]

The assumed nominal functions are

\[ x_1(t) = t, \quad x_2(t) = 0, \quad x_3(t) = 1 + t, \quad u_1(t) = 1, \quad \pi_1 = 1. \quad (181) \]

The numerical results for both PSGRA and DSGRA are given in Tables 11-12. Convergence to the desired stopping conditions occurs in \( N_c = 3 \) cycles and \( N = 9 \) iterations, which include 2 gradient iterations and 7 restorative iterations. The CPU time is 1.32 sec for PSGRA and 0.85 sec for DSGRA.
Example 7. This is a minimum time problem and involves (i) a linear functional, (ii) three nonlinear differential equations, (iii) a component of the initial state given and the remaining components subject to a nonlinear equation, (iv) final state partly given and partly free, and (v) free final time \( \tau \). After setting \( \pi_1 = \tau \), the problem is as follows:

\[
I = \pi_1, \tag{182a}
\]

\[
\dot{x}_1 = \pi_1 x_3 \cos u_1, \quad \dot{x}_2 = \pi_1 x_3 \sin u_1, \quad \dot{x}_3 = \pi_1 \sin u_1, \tag{182b}
\]

\[
x_1(0) = 0, \quad x_2(0) x_3(0) = 0, \tag{182c}
\]

\[
x_1(1) = 1. \tag{182d}
\]

The assumed nominal functions are

\[
x_1(t) = t, \quad x_2(t) = 1, \quad x_3(t) = 0, \quad u_1(t) = 1, \quad \pi_1 = 1. \tag{183}
\]

The numerical results for both PSGRA and DSGRA are given in Tables 13-14. Convergence to the desired stopping conditions occurs in \( N_c = 4 \) cycles and \( N = 12 \) iterations, which include 3 gradient iterations and 9 restorative iterations. The CPU time is 1.41 sec for PSGRA and 0.98 sec for DSGRA.

Example 8. This is a minimum time problem and involves (i) a linear functional, (ii) three nonlinear differential equations, (iii) a component of the initial state given and the remaining components subject to a nonlinear equation, (iv) a component of the final state free and the remaining components subject to a nonlinear equation, and (v) free final time \( \tau \). After setting \( \pi_1 = \tau \), the problem is as follows:

\[
I = \pi_1, \tag{184a}
\]
\[ \dot{x}_1 = \pi_1 x_3 \cos u_1, \quad \dot{x}_2 = \pi_1 x_3 \sin u_1, \quad \dot{x}_3 = \pi_1 \sin u_1, \] (184b)

\[ x_1(0) = 0, \quad x_2(0)x_3(0) = 0, \] (184c)

\[ x_1^2(1) + x_3(1) = 1. \] (184d)

The assumed nominal functions are

\[ x_1(t) = t, \quad x_2(t) = 1, \quad x_3(t) = 0, \quad u_1(t) = 1, \quad \pi_1 = 1. \] (185)

The numerical results for both PSGRA and DSGRA are given in Tables 15-16. Convergence to the desired stopping conditions occurs in \( N_c = 4 \) cycles and \( N = 11 \) iterations, which include 3 gradient iterations and 8 restorative iterations. The CPU time is 1.36 sec for PSGRA and 0.97 sec for DSGRA.

**Example 9.** This example involves (i) a quadratic functional, (ii) two nonlinear differential equations, (iii) a nonlinear nondifferential constraint, (iv) boundary conditions of the fixed endpoint type, and (v) fixed final time \( t = 1 \):

\[ I = \int_0^1 (1 + x_1^2 + x_2^2 + w_1^2) dt, \] (186a)

\[ \dot{x}_1 = w_2 - x_1, \quad \dot{x}_2 = w_1 - x_1 x_2, \] (186b)

\[ 2w_1 - w_2 - x_2^2 = 0, \] (186c)

\[ x_1(0) = 0, \quad x_2(0) = 1, \] (186d)

\[ x_1(1) = 1, \quad x_2(1) = 2. \] (186e)

The assumed nominal functions are

\[ x_1(t) = t, \quad x_2(t) = 1 + t, \quad w_1(t) = 1, \quad w_2(t) = 2 - (1 + t)^2. \] (187)

In this problem, one selects \( u_1 = w_1 \) and \( v_1 = w_2 \).
The numerical results are given in Tables 17-18 for both PSGRA and DSGRA. Convergence to the desired stopping conditions occurs in $N_c = 3$ cycles and $N = 7$ iterations, which include 2 gradient iterations and 5 restorative iterations. The CPU time is 0.81 sec for PSGRA and 0.59 sec for DSGRA.

**Example 10.** This example involves (i) a quadratic functional, (ii) a nonlinear differential equation, (iii) a linear state inequality constraint, (iv) boundary condition of the fixed endpoint type, and (v) fixed final time $\tau = 1$:

\begin{align*}
I &= \int_0^1 (x_1^2 + w_1^2) \, dt \\
\dot{x}_1 &= x_1^2 - w_1, \\
x_1(0) &= 0.9 \geq 0, \\
x_1(0) &= 1, \\
x_1(1) &= 1.
\end{align*}

(188a) (188b) (188c) (188d) (188e)

Upon introducing the auxiliary state variable $x_2$ and the auxiliary control variable $w_2$ defined by

\begin{align*}
x_1 - 0.9 &= x_2^2, \\
\dot{x}_2 &= w_2,
\end{align*}

(189a) (189b)

the above inequality constrained problem can be replaced with the following equality constrained problem:
\[ I = \int_0^1 (x_1^2 + w_2^2) \, dt, \quad (190a) \]
\[ x_1 = x_1^2 - w_1, \quad x_2 = w_2, \quad (190b) \]
\[ x_1^2 - w_1 - 2x_2^2 = 0, \quad (190c) \]
\[ x_1(0) = 1, \quad x_2(0) = \sqrt[3]{0.1}, \quad (190d) \]
\[ x_1(1) = 1. \quad (190e) \]

The assumed nominal functions are
\[ x_1(t) = 1, \quad x_2(t) = \sqrt[3]{0.1}, \quad w_1(t) = 1 - 2\sqrt[3]{0.1}, \quad w_2(t) = 1. \quad (191) \]

In this problem, one selects \( u_1 = w_2, \ v_1 = w_1 \).

The numerical results are given in Tables 19-20 for both PSGRA and DSGRA.

Convergence to the desired stopping conditions occurs in \( N_c = 7 \) cycles and \( N = 16 \) iterations, which include 6 gradient iterations and 10 restorative iterations. The CPU time is 1.49 sec for PSGRA and 1.15 sec for DSGRA.

Example 11. This example involves (i) a quadratic functional, (ii) two linear differential equations, (iii) a linear state inequality constraint, (iv) boundary conditions of fixed endpoint type, and (v) fixed final time \( t = 1 \):

\[ I = \int_0^1 w_2^2 \, dt, \quad (192a) \]
\[ x_1 = x_2, \quad x_2 = w_1, \quad (192b) \]
\[ 0.15 - x_1 \geq 0, \quad (192c) \]
\[ x_1(0) = 0, \quad x_2(0) = 1, \quad (192d) \]
\[ x_1(1) = 0, \quad x_2(1) = -1. \quad (192e) \]
Upon introducing the auxiliary state variables $x_3$, $x_4$ and the auxiliary control variable $w_2$ defined by

$$0.15 - x_1 = x_3^2, \quad \dot{x}_3 = x_4, \quad \dot{x}_4 = w_2,$$

(193)

the above inequality constrained problem can be replaced with the following equality constrained problem:

$$I = \int_0^1 w_1^2 dt,$$

(194a)

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = w_1, \quad \dot{x}_3 = x_4, \quad \dot{x}_4 = w_2,$$

(194b)

$$w_1 + 2x_3w_2 + 2x_4^2 = 0,$$

(194c)

$$x_1(0) = 0, \quad x_2(0) = 1, \quad x_3(0) = \sqrt{(0.15)}, \quad x_4(0) = -\sqrt{(1/0.60)},$$

(194d)

$$x_1(1) = 0, \quad x_2(1) = -1.$$  

(194e)

The assumed nominal functions are

$$x_1(t) = 0, \quad x_2(t) = 1-2t, \quad x_3(t) = \sqrt{(0.15)(1-2t)}, \quad x_4(t) = \sqrt{(1/0.60)(2t-1)},$$

(195a)

$$w_1(t) = -(1/0.30)(2t-1)^2, \quad w_2(t) = 0.$$  

(195b)

In this problem, one selects $u_1 = w_2$, $v_1 = w_1$.

The numerical results are given in Tables 21-22 for PSGRA and in Tables 23-24 for DSGRA. Convergence to the desired stopping conditions occurs in $N_c = 11$ cycles and $N = 22$ iterations, which include 10 gradient iterations and 12 restorative iterations. The CPU time is 4.72 sec for PSGRA and 3.36 sec for DSGRA.

**Example 12.** This is a minimum time problem and involves (i) a linear functional, (ii) two nonlinear differential equations, (iii) a state derivative
inequality constraint, (iv) boundary conditions of the fixed endpoint type, and (v) free final time $\tau$. After setting $\pi_1 = \tau$, the problem is as follows:

$$I = \pi_1,$$  \hspace{1cm} (196a)

$$\dot{x}_1 = \pi_1 w_1, \quad \dot{x}_2 = \pi_1 (w_1^2 - x_1^2 - 0.5),$$  \hspace{1cm} (196b)

$$\dot{x}_2 + 0.5 \geq 0,$$  \hspace{1cm} (196c)

$$x_1(0) = 0, \quad x_2(0) = 0,$$  \hspace{1cm} (196d)

$$x_1(1) = 1, \quad x_2(1) = -\pi/4.$$  \hspace{1cm} (196e)

Upon introducing the auxiliary control variable $w_2$ defined by

$$\dot{x}_2 + 0.5 - w_2^2 = 0,$$  \hspace{1cm} (197)

the above inequality constrained problem can be replaced with the following equality constrained problem:

$$I = \pi_1,$$  \hspace{1cm} (198a)

$$\dot{x}_1 = \pi_1 w_1, \quad \dot{x}_2 = \pi_1 (w_1^2 - x_1^2 - 0.5),$$  \hspace{1cm} (198b)

$$\dot{w}_1^2 - x_1^2 - w_2^2 = 0,$$  \hspace{1cm} (198c)

$$x_1(0) = 0, \quad x_2(0) = 0,$$  \hspace{1cm} (198d)

$$x_1(1) = 1, \quad x_2(1) = -\pi/4.$$  \hspace{1cm} (198e)

The assumed nominal functions are

$$x_1(t) = t, \quad x_2(t) = -(\pi/4)t, \quad w_1(t) = \sqrt{1 + t^2}, \quad w_2(t) = 1, \quad \pi_1 = 1.$$  \hspace{1cm} (199)
In this problem, one selects $u_1 = w_2$, $v_1 = w_1$.

The numerical results are given in Tables 25-26 for both PSGRA and DSGRA. Convergence to the desired stopping conditions occurs in $N_c = 5$ cycles and $N = 10$ iterations, which include 4 gradient iterations and 6 restorative iterations. The CPU time is 1.20 sec for PSGRA and 0.92 sec for DSGRA.
21. Discussion and Conclusions

This thesis considers duality properties and their application to
sequential gradient-restoration algorithms (SGRA) for optimal control problems.
Two problems are studied: (P1) the basic problem and (P2) the general problem.
In Problem (P1), the minimization of a functional is considered, subject to
differential constraints and final constraints, the initial state being given;
in Problem (P2), the minimization of a functional is considered, subject to
differential constraints, nondifferential constraints, initial constraints, and
final constraints. Depending on whether the primal formulation is used or
the dual formulation is used, one obtains a primal sequential gradient-restoration
algorithm (PSGRA) and a dual sequential gradient-restoration algorithm (DSGRA).

With particular reference to Problem (P2), it is found convenient to split
the control vector into an independent control vector and a dependent control
vector, the latter having the same dimension as the nondifferential constraint
vector. This modification enhances the computational efficiency of both the
primal formulation and the dual formulation.

The basic property of the dual formulation is that the Lagrange multipliers
associated with the gradient phase and the restoration phase of SGRA minimize a
special functional, quadratic in multipliers, subject to the multiplier differential
equations and boundary conditions, for given state, control, and parameter.
This duality property yields considerable computational benefits in that the
auxiliary optimal control problems associated with the gradient phase and
the restoration phase of SGRA can be reduced to mathematical programming problems
involving a finite number of parameters as unknowns.

Twelve numerical examples are solved using both the primal formulation
and the dual formulation (Tables 1-26). Summary results can be found in Tables
27-28.
Table 27 refers to problems without nondifferential constraints and shows the following quantities: the CPU time $T_p$ of the primal formulation (PSGRA), the CPU time $T_D$ of the dual formulation (DSGRA), and the ratio $T_D/T_p$. Clearly, the reduction in CPU time, due to using DSGRA in place of PSGRA, is in the range 29% to 36%, depending on the particular example.

Table 28 refers to problems with nondifferential constraints and shows the quantities $T_p$, $T_D$, as well as the ratio $T_D/T_p$. Clearly, the reduction of CPU time, due to using DSGRA in place of PSGRA, is in the range 23% to 29%, depending on the particular example.

Also shown in Table 28 is the CPU time $T_{pp}$ of the primal algorithm (PPSGRA) developed in Refs. 2-3, as well as the ratios $T_p/T_{pp}$ and $T_D/T_{pp}$. It must be noted that PPSGRA differs from PSGRA and DSGRA in two aspects: (i) the independent control vector $u(t)$ and the dependent control vector $v(t)$ are joined in a single vector in PPSGRA, while they are kept separated in PSGRA and DSGRA; and (ii) rigorous enforcement of the nondifferential constraint (77b) is not required at the beginning of each iteration of PPSGRA, while it is required for PSGRA and DSGRA. Clearly, the reduction in CPU time, due to using PSGRA in place of PPSGRA, is in the range 39% to 62%, depending on the particular examples. More importantly, the reduction in CPU time, due to using DSGRA in place of PPSGRA, is in the range 53% to 71%, depending on the particular example.
Table 1. Convergence history, PSGRA and DSGRA, Example 1.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.72E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.33E-10</td>
<td>0.97E+00</td>
<td>33.67701</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0.85E-13</td>
<td>0.50E-02</td>
<td>33.46606</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>0.52E-09</td>
<td>0.42E-04</td>
<td>33.46484</td>
</tr>
</tbody>
</table>

Table 2. Converged solution, PSGRA and DSGRA, Example 1.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>1.0000</td>
<td>-8.3428</td>
</tr>
<tr>
<td>0.1</td>
<td>-0.7862</td>
<td>0.2778</td>
<td>-6.3676</td>
</tr>
<tr>
<td>0.2</td>
<td>-1.3012</td>
<td>-0.2367</td>
<td>-3.8633</td>
</tr>
<tr>
<td>0.3</td>
<td>-1.5838</td>
<td>-0.5625</td>
<td>-1.4846</td>
</tr>
<tr>
<td>0.4</td>
<td>-1.6736</td>
<td>-0.7169</td>
<td>0.4683</td>
</tr>
<tr>
<td>0.5</td>
<td>-1.6003</td>
<td>-0.7107</td>
<td>1.9932</td>
</tr>
<tr>
<td>0.6</td>
<td>-1.3781</td>
<td>-0.5437</td>
<td>3.2523</td>
</tr>
<tr>
<td>0.7</td>
<td>-1.0080</td>
<td>-0.2056</td>
<td>4.4921</td>
</tr>
<tr>
<td>0.8</td>
<td>-0.4877</td>
<td>0.3180</td>
<td>6.0526</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1808</td>
<td>1.0416</td>
<td>8.4996</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000</td>
<td>2.0000</td>
<td>13.0497</td>
</tr>
</tbody>
</table>
Table 3. Convergence history, PSGRA and DSGRA, Example 2.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.11E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.18E-08</td>
<td>0.67E+00</td>
<td>-1.11665</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>0.18E-11</td>
<td>0.34E-01</td>
<td>-1.16519</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.12E-10</td>
<td>0.31E-02</td>
<td>-1.16923</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>0.58E-15</td>
<td>0.65E-03</td>
<td>-1.16950</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>12</td>
<td>0.20E-08</td>
<td>0.18E-03</td>
<td>-1.16961</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>13</td>
<td>0.36E-08</td>
<td>0.50E-04</td>
<td>-1.16964</td>
</tr>
</tbody>
</table>

Table 4. Converged solution, PSGRA and DSGRA, Example 2.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.3333</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0938</td>
<td>0.0047</td>
<td>1.3049</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1856</td>
<td>0.0187</td>
<td>1.2610</td>
</tr>
<tr>
<td>0.3</td>
<td>0.2742</td>
<td>0.0417</td>
<td>1.2005</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3575</td>
<td>0.0734</td>
<td>1.1131</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4310</td>
<td>0.1129</td>
<td>0.9784</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4842</td>
<td>0.1589</td>
<td>0.7517</td>
</tr>
<tr>
<td>0.7</td>
<td>0.4922</td>
<td>0.2083</td>
<td>0.3661</td>
</tr>
<tr>
<td>0.8</td>
<td>0.4141</td>
<td>0.2544</td>
<td>-0.1521</td>
</tr>
<tr>
<td>0.9</td>
<td>0.2381</td>
<td>0.2877</td>
<td>-0.6088</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0000</td>
<td>0.3000</td>
<td>-0.8959</td>
</tr>
</tbody>
</table>
### Table 5. Convergence history, PSGRA and DSGRA, Example 3.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.53E+00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.75E-16</td>
<td>0.53E-01</td>
<td>1.58101</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0.33E-08</td>
<td>0.13E-03</td>
<td>1.57080</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>0.29E-08</td>
<td>0.16E-05</td>
<td>1.57075</td>
</tr>
</tbody>
</table>

### Table 6. Converged solution, PSGRA and DSGRA, Example 3.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.9998</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1564</td>
<td>-0.1544</td>
<td>0.9875</td>
</tr>
<tr>
<td>0.2</td>
<td>0.3089</td>
<td>-0.2938</td>
<td>0.9508</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4539</td>
<td>-0.4043</td>
<td>0.8907</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5876</td>
<td>-0.4753</td>
<td>0.8087</td>
</tr>
<tr>
<td>0.5</td>
<td>0.7069</td>
<td>-0.4997</td>
<td>0.7068</td>
</tr>
<tr>
<td>0.6</td>
<td>0.8088</td>
<td>-0.4752</td>
<td>0.5875</td>
</tr>
<tr>
<td>0.7</td>
<td>0.8907</td>
<td>-0.4043</td>
<td>0.4539</td>
</tr>
<tr>
<td>0.8</td>
<td>0.9508</td>
<td>-0.2937</td>
<td>0.3092</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9875</td>
<td>-0.1544</td>
<td>0.1572</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0017</td>
</tr>
</tbody>
</table>
Table 7. Convergence history, PSGRA and DSGRA, Example 4.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.17E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0.44E-16</td>
<td>0.25E+00</td>
<td>1.83370</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>0.30E-09</td>
<td>0.42E-01</td>
<td>1.78267</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>0.40E-09</td>
<td>0.62E-03</td>
<td>1.77263</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>0.11E-16</td>
<td>0.92E-05</td>
<td>1.77246</td>
</tr>
</tbody>
</table>

Table 8. Converged solution, PSGRA and DSGRA, Example 4.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.5708</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0016</td>
<td>0.0156</td>
<td>0.1765</td>
<td>1.4133</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0129</td>
<td>0.0608</td>
<td>0.3487</td>
<td>1.2558</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0426</td>
<td>0.1312</td>
<td>0.5122</td>
<td>1.0985</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0975</td>
<td>0.2198</td>
<td>0.6630</td>
<td>0.9412</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1820</td>
<td>0.3180</td>
<td>0.7975</td>
<td>0.7841</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2976</td>
<td>0.4162</td>
<td>0.9124</td>
<td>0.6270</td>
</tr>
<tr>
<td>0.7</td>
<td>0.4428</td>
<td>0.5047</td>
<td>1.0047</td>
<td>0.4699</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6132</td>
<td>0.5748</td>
<td>1.0722</td>
<td>0.3127</td>
</tr>
<tr>
<td>0.9</td>
<td>0.8019</td>
<td>0.6197</td>
<td>1.1132</td>
<td>0.1550</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000</td>
<td>0.6347</td>
<td>1.1267</td>
<td>-0.0033</td>
</tr>
</tbody>
</table>
Table 9. Convergence history, PSGRA and DSGRA, Example 5.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.20E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0.75E-11</td>
<td>0.29E+00</td>
<td>1.01089</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>0.52E-14</td>
<td>0.10E-02</td>
<td>0.86486</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.42E-12</td>
<td>0.20E-04</td>
<td>0.86431</td>
</tr>
</tbody>
</table>

Table 10. Converged solution, PSGRA and DSGRA, Example 5.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>0.6456</td>
<td>0.7637</td>
<td>1.6129</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1305</td>
<td>0.4488</td>
<td>0.8742</td>
<td>1.4166</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2463</td>
<td>0.2968</td>
<td>0.9943</td>
<td>1.2686</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3507</td>
<td>0.1784</td>
<td>1.1197</td>
<td>1.1516</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4459</td>
<td>0.0873</td>
<td>1.2479</td>
<td>1.0530</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5330</td>
<td>0.0203</td>
<td>1.3771</td>
<td>0.9636</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6125</td>
<td>-0.0245</td>
<td>1.5061</td>
<td>0.8768</td>
</tr>
<tr>
<td>0.7</td>
<td>0.6845</td>
<td>-0.0480</td>
<td>1.6341</td>
<td>0.7875</td>
</tr>
<tr>
<td>0.8</td>
<td>0.7485</td>
<td>-0.0510</td>
<td>1.7598</td>
<td>0.6923</td>
</tr>
<tr>
<td>0.9</td>
<td>0.8039</td>
<td>-0.0344</td>
<td>1.8823</td>
<td>0.5896</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8501</td>
<td>0.0000</td>
<td>2.0000</td>
<td>0.4791</td>
</tr>
</tbody>
</table>
Table 11. Convergence history, PSGRA and DSGRA, Example 6.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.87E+00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0.54E-10</td>
<td>0.32E+00</td>
<td>1.02610</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>0.24E-12</td>
<td>0.70E-03</td>
<td>0.86466</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.80E-13</td>
<td>0.34E-05</td>
<td>0.86430</td>
</tr>
</tbody>
</table>

Table 12. Converged solution, PSGRA and DSGRA, Example 6.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>Ul</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>0.6488</td>
<td>0.7610</td>
<td>1.6249</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1313</td>
<td>0.4495</td>
<td>0.8722</td>
<td>1.4237</td>
</tr>
<tr>
<td>0.2</td>
<td>0.2476</td>
<td>0.2962</td>
<td>0.9929</td>
<td>1.2730</td>
</tr>
<tr>
<td>0.3</td>
<td>0.3523</td>
<td>0.1771</td>
<td>1.1188</td>
<td>1.1545</td>
</tr>
<tr>
<td>0.4</td>
<td>0.4477</td>
<td>0.0857</td>
<td>1.2473</td>
<td>1.0548</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5349</td>
<td>0.0186</td>
<td>1.3768</td>
<td>0.9645</td>
</tr>
<tr>
<td>0.6</td>
<td>0.6145</td>
<td>-0.0260</td>
<td>1.5061</td>
<td>0.8768</td>
</tr>
<tr>
<td>0.7</td>
<td>0.6864</td>
<td>-0.0493</td>
<td>1.6341</td>
<td>0.7866</td>
</tr>
<tr>
<td>0.8</td>
<td>0.7503</td>
<td>-0.0519</td>
<td>1.7599</td>
<td>0.6908</td>
</tr>
<tr>
<td>0.9</td>
<td>0.8056</td>
<td>-0.0349</td>
<td>1.8823</td>
<td>0.5874</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8516</td>
<td>0.0000</td>
<td>2.0000</td>
<td>0.4761</td>
</tr>
</tbody>
</table>
Table 13. Convergence history, PSGRA and DSGRA, Example 7.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.17E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0.44E-16</td>
<td>0.25E+00</td>
<td>1.83370</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>8</td>
<td>0.30E-09</td>
<td>0.42E-01</td>
<td>1.78267</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>0.40E-09</td>
<td>0.62E-03</td>
<td>1.77263</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>0.11E-16</td>
<td>0.92E-05</td>
<td>1.77246</td>
</tr>
</tbody>
</table>

Table 14. Converged solution, PSGRA and DSGRA, Example 7.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.5708</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0016</td>
<td>1.0156</td>
<td>0.1765</td>
<td>1.4133</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0129</td>
<td>1.0608</td>
<td>0.3487</td>
<td>1.2558</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0426</td>
<td>1.1312</td>
<td>0.5122</td>
<td>1.0985</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0975</td>
<td>1.2198</td>
<td>0.6630</td>
<td>0.9412</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1820</td>
<td>1.3180</td>
<td>0.7975</td>
<td>0.7841</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2976</td>
<td>1.4162</td>
<td>0.9124</td>
<td>0.6270</td>
</tr>
<tr>
<td>0.7</td>
<td>0.4428</td>
<td>1.5047</td>
<td>1.0047</td>
<td>0.4699</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6132</td>
<td>1.5748</td>
<td>1.0722</td>
<td>0.3127</td>
</tr>
<tr>
<td>0.9</td>
<td>0.8019</td>
<td>1.6197</td>
<td>1.1132</td>
<td>0.1550</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000</td>
<td>1.6347</td>
<td>1.1267</td>
<td>-0.0033</td>
</tr>
</tbody>
</table>
Table 15. Convergence history, PSGRA and DSGRA, Example 8.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.17E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.16E-13</td>
<td>0.46E-01</td>
<td>1.02942</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>0.26E-17</td>
<td>0.33E-02</td>
<td>1.00404</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.11E-10</td>
<td>0.80E-03</td>
<td>1.00064</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>0.13E-13</td>
<td>0.96E-04</td>
<td>1.00012</td>
</tr>
</tbody>
</table>

Table 16. Converged solution, PSGRA and DSGRA, Example 8.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>U1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>1.5802</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0000</td>
<td>1.0050</td>
<td>0.1000</td>
<td>1.5743</td>
</tr>
<tr>
<td>0.2</td>
<td>0.0000</td>
<td>1.0200</td>
<td>0.2000</td>
<td>1.5684</td>
</tr>
<tr>
<td>0.3</td>
<td>0.0001</td>
<td>1.0450</td>
<td>0.3000</td>
<td>1.5626</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0005</td>
<td>1.0800</td>
<td>0.4000</td>
<td>1.5567</td>
</tr>
<tr>
<td>0.5</td>
<td>0.0013</td>
<td>1.1250</td>
<td>0.5000</td>
<td>1.5508</td>
</tr>
<tr>
<td>0.6</td>
<td>0.0025</td>
<td>1.1800</td>
<td>0.6000</td>
<td>1.5449</td>
</tr>
<tr>
<td>0.7</td>
<td>0.0044</td>
<td>1.2450</td>
<td>0.7000</td>
<td>1.5390</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0070</td>
<td>1.3200</td>
<td>0.7999</td>
<td>1.5332</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0105</td>
<td>1.4049</td>
<td>0.8999</td>
<td>1.5274</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0149</td>
<td>1.4998</td>
<td>0.9998</td>
<td>1.5216</td>
</tr>
</tbody>
</table>
Table 17. Convergence history, PSGRA and DSGRA, Example 9.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.72E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.33E-10</td>
<td>0.97E+00</td>
<td>33.67701</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0.85E-13</td>
<td>0.50E-02</td>
<td>33.46606</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>7</td>
<td>0.52E-09</td>
<td>0.42E-04</td>
<td>33.46484</td>
</tr>
</tbody>
</table>

Table 18. Converged solution, PSGRA and DSGRA, Example 9.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>W1</th>
<th>W2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>1.0000</td>
<td>-8.3428</td>
<td>-17.6856</td>
</tr>
<tr>
<td>0.1</td>
<td>-0.7862</td>
<td>0.2778</td>
<td>-6.3676</td>
<td>-12.8124</td>
</tr>
<tr>
<td>0.2</td>
<td>-1.3012</td>
<td>-0.2367</td>
<td>-3.8633</td>
<td>-7.7826</td>
</tr>
<tr>
<td>0.3</td>
<td>-1.5838</td>
<td>-0.5625</td>
<td>-1.4846</td>
<td>-3.2856</td>
</tr>
<tr>
<td>0.4</td>
<td>-1.6736</td>
<td>-0.7169</td>
<td>0.4683</td>
<td>0.4226</td>
</tr>
<tr>
<td>0.5</td>
<td>-1.6003</td>
<td>-0.7107</td>
<td>1.9932</td>
<td>3.4812</td>
</tr>
<tr>
<td>0.6</td>
<td>-1.3781</td>
<td>-0.5437</td>
<td>3.2523</td>
<td>6.2089</td>
</tr>
<tr>
<td>0.7</td>
<td>-1.0080</td>
<td>-0.2056</td>
<td>4.4921</td>
<td>8.9419</td>
</tr>
<tr>
<td>0.8</td>
<td>-0.4877</td>
<td>0.3180</td>
<td>6.0526</td>
<td>12.0042</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1808</td>
<td>1.0416</td>
<td>8.4996</td>
<td>15.9142</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000</td>
<td>2.0000</td>
<td>13.0497</td>
<td>22.0994</td>
</tr>
</tbody>
</table>
Table 19. Convergence history, PSGRA and DSGRA, Example 10.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.14E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>4</td>
<td>0.13E-13</td>
<td>0.24E+00</td>
<td>1.76035</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>0.83E-27</td>
<td>0.42E-02</td>
<td>1.66015</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>0.83E-32</td>
<td>0.43E-03</td>
<td>1.65747</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>10</td>
<td>0.14E-31</td>
<td>0.19E-03</td>
<td>1.65707</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>0.76E-32</td>
<td>0.12E-03</td>
<td>1.65691</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>14</td>
<td>0.56E-32</td>
<td>0.10E-03</td>
<td>1.65679</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>16</td>
<td>0.83E-32</td>
<td>0.72E-04</td>
<td>1.65670</td>
</tr>
</tbody>
</table>

Table 20. Converged solution, PSGRA and DSGRA, Example 10.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>W1</th>
<th>W2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.0000</td>
<td>0.3162</td>
<td>1.7417</td>
<td>-1.1728</td>
</tr>
<tr>
<td>0.1</td>
<td>0.9409</td>
<td>0.2023</td>
<td>1.3363</td>
<td>-1.1144</td>
</tr>
<tr>
<td>0.2</td>
<td>0.9093</td>
<td>0.0965</td>
<td>1.0094</td>
<td>-0.9458</td>
</tr>
<tr>
<td>0.3</td>
<td>0.9006</td>
<td>0.0237</td>
<td>0.8346</td>
<td>-0.4987</td>
</tr>
<tr>
<td>0.4</td>
<td>0.9001</td>
<td>-0.0078</td>
<td>0.8075</td>
<td>-0.1670</td>
</tr>
<tr>
<td>0.5</td>
<td>0.9002</td>
<td>-0.0154</td>
<td>0.8104</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.6</td>
<td>0.9001</td>
<td>-0.0078</td>
<td>0.8127</td>
<td>0.1670</td>
</tr>
<tr>
<td>0.7</td>
<td>0.9006</td>
<td>0.0237</td>
<td>0.7874</td>
<td>0.4987</td>
</tr>
<tr>
<td>0.8</td>
<td>0.9093</td>
<td>0.0965</td>
<td>0.6443</td>
<td>0.9458</td>
</tr>
<tr>
<td>0.9</td>
<td>0.9409</td>
<td>0.2023</td>
<td>0.4344</td>
<td>1.1143</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000</td>
<td>0.3162</td>
<td>0.2583</td>
<td>1.1728</td>
</tr>
</tbody>
</table>
Table 21. Convergence history, PSGRA, Example 11.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.99E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0.30E-12</td>
<td>0.28E+00</td>
<td>6.11724</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.34E-22</td>
<td>0.28E-01</td>
<td>5.95995</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.10E-20</td>
<td>0.93E-02</td>
<td>5.93992</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>0.13E-21</td>
<td>0.36E-02</td>
<td>5.93375</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>0.33E-22</td>
<td>0.21E-02</td>
<td>5.93099</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>0.49E-23</td>
<td>0.12E-02</td>
<td>5.92944</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>0.75E-24</td>
<td>0.72E-03</td>
<td>5.92861</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>18</td>
<td>0.68E-08</td>
<td>0.65E-03</td>
<td>5.92794</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>19</td>
<td>0.48E-08</td>
<td>0.25E-03</td>
<td>5.92762</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>0.32E-24</td>
<td>0.61E-03</td>
<td>5.92714</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>22</td>
<td>0.17E-09</td>
<td>0.87E-04</td>
<td>5.92696</td>
</tr>
</tbody>
</table>

Table 22. Converged solution, PSGRA, Example 11.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>W1</th>
<th>W2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.3873</td>
<td>-1.2910</td>
<td>-4.4451</td>
<td>1.4353</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0794</td>
<td>0.6050</td>
<td>0.2657</td>
<td>-1.1386</td>
<td>-3.4566</td>
<td>1.6257</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1243</td>
<td>0.3084</td>
<td>0.1604</td>
<td>-0.9616</td>
<td>-2.4777</td>
<td>1.9596</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1444</td>
<td>0.1096</td>
<td>0.0752</td>
<td>-0.7293</td>
<td>-1.4813</td>
<td>2.7774</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1497</td>
<td>0.0143</td>
<td>0.0178</td>
<td>-0.4035</td>
<td>-0.4557</td>
<td>3.6614</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1500</td>
<td>0.0001</td>
<td>-0.0035</td>
<td>-0.0160</td>
<td>0.0277</td>
<td>4.0224</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1498</td>
<td>-0.0119</td>
<td>0.0152</td>
<td>0.3906</td>
<td>-0.4272</td>
<td>4.0151</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1447</td>
<td>-0.1092</td>
<td>0.0731</td>
<td>0.7471</td>
<td>-1.5378</td>
<td>2.8840</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1245</td>
<td>-0.3101</td>
<td>0.1598</td>
<td>0.9703</td>
<td>-2.4683</td>
<td>1.8318</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0795</td>
<td>-0.6060</td>
<td>0.2656</td>
<td>1.1409</td>
<td>-3.4495</td>
<td>1.5931</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0000</td>
<td>-1.0000</td>
<td>0.3873</td>
<td>1.2910</td>
<td>-4.4439</td>
<td>1.4337</td>
</tr>
</tbody>
</table>
### Table 23. Convergence history, DSGRA, Example 11.

<table>
<thead>
<tr>
<th>(N_c)</th>
<th>(N_g)</th>
<th>(N_r)</th>
<th>(N)</th>
<th>(P)</th>
<th>(Q)</th>
<th>(I)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.99E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>0.30E-12</td>
<td>0.28E+00</td>
<td>6.1724</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.71E-13</td>
<td>0.28E-01</td>
<td>5.95995</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.13E-13</td>
<td>0.93E-02</td>
<td>5.93992</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>0.29E-14</td>
<td>0.36E-02</td>
<td>5.93375</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>0.24E-15</td>
<td>0.21E-02</td>
<td>5.93099</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>0.73E-16</td>
<td>0.12E-02</td>
<td>5.92944</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>0.38E-17</td>
<td>0.73E-03</td>
<td>5.92861</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>0</td>
<td>18</td>
<td>0.68E-08</td>
<td>0.64E-03</td>
<td>5.92800</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>0</td>
<td>19</td>
<td>0.47E-08</td>
<td>0.25E-03</td>
<td>5.92761</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>0.83E-17</td>
<td>0.61E-03</td>
<td>5.92714</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>0</td>
<td>22</td>
<td>0.19E-09</td>
<td>0.88E-04</td>
<td>5.92687</td>
</tr>
</tbody>
</table>

### Table 24. Converged solution, DSGRA, Example 11.

<table>
<thead>
<tr>
<th>(t)</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(x_3)</th>
<th>(x_4)</th>
<th>(w_1)</th>
<th>(w_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.3873</td>
<td>-1.2910</td>
<td>-4.4450</td>
<td>1.4352</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0794</td>
<td>0.6050</td>
<td>0.2657</td>
<td>-1.1386</td>
<td>-3.4567</td>
<td>1.6258</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1243</td>
<td>0.3084</td>
<td>0.1604</td>
<td>-0.9615</td>
<td>-2.4778</td>
<td>1.9599</td>
</tr>
<tr>
<td>0.3</td>
<td>0.1444</td>
<td>0.1096</td>
<td>0.0752</td>
<td>-0.7293</td>
<td>-1.4812</td>
<td>2.7775</td>
</tr>
<tr>
<td>0.4</td>
<td>0.1497</td>
<td>0.0143</td>
<td>0.0178</td>
<td>-0.4035</td>
<td>-0.4557</td>
<td>3.6610</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1500</td>
<td>-0.0001</td>
<td>-0.0035</td>
<td>-0.0160</td>
<td>0.0277</td>
<td>4.0221</td>
</tr>
<tr>
<td>0.6</td>
<td>0.1498</td>
<td>-0.0119</td>
<td>0.0152</td>
<td>0.3905</td>
<td>-0.4272</td>
<td>4.0150</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1447</td>
<td>-0.1092</td>
<td>0.0731</td>
<td>0.7471</td>
<td>-1.5378</td>
<td>2.8848</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1245</td>
<td>-0.3101</td>
<td>0.1598</td>
<td>0.9703</td>
<td>-2.4684</td>
<td>1.8316</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0795</td>
<td>-0.6060</td>
<td>0.2656</td>
<td>1.1409</td>
<td>-3.4494</td>
<td>1.5929</td>
</tr>
<tr>
<td>1.0</td>
<td>0.0000</td>
<td>-1.0000</td>
<td>0.3873</td>
<td>1.2910</td>
<td>-4.4432</td>
<td>1.4330</td>
</tr>
</tbody>
</table>
Table 25. Convergence history, PSGRA and DSGRA, Example 12.

<table>
<thead>
<tr>
<th>Nc</th>
<th>Ng</th>
<th>Nr</th>
<th>N</th>
<th>P</th>
<th>Q</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0.17E+01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>0.20E-08</td>
<td>0.69E-02</td>
<td>1.82432</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>0.39E-09</td>
<td>0.74E-03</td>
<td>1.82247</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>0.78E-14</td>
<td>0.21E-03</td>
<td>1.82233</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>0.37E-16</td>
<td>0.10E-03</td>
<td>1.82229</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>10</td>
<td>0.67E-08</td>
<td>0.52E-04</td>
<td>1.82225</td>
</tr>
</tbody>
</table>

Table 26. Converged solution, PSGRA and DSGRA, Example 12.

<table>
<thead>
<tr>
<th>t</th>
<th>X1</th>
<th>X2</th>
<th>W1</th>
<th>W2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.5043</td>
<td>0.5043</td>
</tr>
<tr>
<td>0.1</td>
<td>0.0911</td>
<td>-0.0461</td>
<td>0.4921</td>
<td>0.4836</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1785</td>
<td>-0.0987</td>
<td>0.4656</td>
<td>0.4300</td>
</tr>
<tr>
<td>0.3</td>
<td>0.2601</td>
<td>-0.1621</td>
<td>0.4275</td>
<td>0.3392</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3333</td>
<td>-0.2400</td>
<td>0.3746</td>
<td>0.1710</td>
</tr>
<tr>
<td>0.5</td>
<td>0.4021</td>
<td>-0.3298</td>
<td>0.4021</td>
<td>0.0078</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4824</td>
<td>-0.4209</td>
<td>0.4824</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5789</td>
<td>-0.5121</td>
<td>0.5789</td>
<td>0.0002</td>
</tr>
<tr>
<td>0.8</td>
<td>0.6946</td>
<td>-0.6032</td>
<td>0.6946</td>
<td>0.0005</td>
</tr>
<tr>
<td>0.9</td>
<td>0.8334</td>
<td>-0.6943</td>
<td>0.8334</td>
<td>-0.0003</td>
</tr>
<tr>
<td>1.0</td>
<td>1.0000</td>
<td>-0.7854</td>
<td>1.0000</td>
<td>0.0004</td>
</tr>
</tbody>
</table>
Table 27. CPU time (sec) for problems without nondifferential constraints.

<table>
<thead>
<tr>
<th>Example</th>
<th>$T_P$</th>
<th>$T_D$</th>
<th>$T_D/T_P$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 1</td>
<td>0.71</td>
<td>0.48</td>
<td>0.68</td>
</tr>
<tr>
<td>Example 2</td>
<td>1.24</td>
<td>0.87</td>
<td>0.70</td>
</tr>
<tr>
<td>Example 3</td>
<td>0.73</td>
<td>0.51</td>
<td>0.70</td>
</tr>
<tr>
<td>Example 4</td>
<td>1.41</td>
<td>0.97</td>
<td>0.69</td>
</tr>
<tr>
<td>Example 5</td>
<td>1.30</td>
<td>0.84</td>
<td>0.65</td>
</tr>
<tr>
<td>Example 6</td>
<td>1.32</td>
<td>0.85</td>
<td>0.64</td>
</tr>
<tr>
<td>Example 7</td>
<td>1.41</td>
<td>0.98</td>
<td>0.70</td>
</tr>
<tr>
<td>Example 8</td>
<td>1.36</td>
<td>0.97</td>
<td>0.71</td>
</tr>
</tbody>
</table>

Table 28. CPU time (sec) for problems with nondifferential constraints.

<table>
<thead>
<tr>
<th>Example</th>
<th>$T_P$</th>
<th>$T_D$</th>
<th>$T_D/T_P$</th>
<th>$T_{pp}$</th>
<th>$T_P/T_{pp}$</th>
<th>$T_D/T_{pp}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example 9</td>
<td>0.81</td>
<td>0.59</td>
<td>0.73</td>
<td>1.45</td>
<td>0.56</td>
<td>0.41</td>
</tr>
<tr>
<td>Example 10</td>
<td>1.49</td>
<td>1.15</td>
<td>0.77</td>
<td>2.43</td>
<td>0.61</td>
<td>0.47</td>
</tr>
<tr>
<td>Example 11</td>
<td>4.72</td>
<td>3.36</td>
<td>0.71</td>
<td>8.76</td>
<td>0.54</td>
<td>0.38</td>
</tr>
<tr>
<td>Example 12</td>
<td>1.20</td>
<td>0.92</td>
<td>0.77</td>
<td>3.16</td>
<td>0.38</td>
<td>0.29</td>
</tr>
</tbody>
</table>
References


