British Journal of Mathematics & Computer Science
1(3): 165-180, 2011

SCIENCEDOMAIN *international*
www.sciencedomain.org

Comparative Analysis of Extended Conjugate Method (ECGM) and Euler-Lagrange (E-L) Algorithms for Reaction Diffusion Problem

J. O. Omolehin[1], K. Rauf[1*], O. Abikoye[2], O. Enikuomehin[3] and S. O. Sogunro[4]

[1]*Mathematics Department, University of Ilorin, Ilorin, Nigeria.*
[2]*Computer Science Department, University of Ilorin, Ilorin, Nigeria.*
[3]*Computer Science Department, Lagos State University, Lagos, Nigeria.*
[4]*Mathematics and Statistics Department, Lagos State Polytechnics, Lagos, Nigeria.*

Research Article

Received 3rd May 2011
Accepted 17th May 2011
Online Ready 2nd June 2011

Abstract

Reaction Diffusion Control problem is a class of optimization problem in the form
Minimize $\int_0^T \{av^2(t) + bu^2(t)\}dt$
Subject to $\dot{v}(t) - \dot{u}(t) = cv(t) + du(t)$.
It has been proved that a unique solution exists. In this work, the solution provided by ECGM and Euler-Lagrange algorithm, are admissible and favourably comparable.
Keywords: ECGM; Euler-Lagrange; Diffusion; Control; Convergence rate;

2010 Mathematics Subject Classification: 78M50, 93B40, 93CXX & G 1.7

1 Introduction

We begin by considering descent with a functional f on a Hilbert space H in which f is a Taylor series expansion truncated after the second order terms, namely:

* *Corresponding author: Email: raufkml@gmail.com;*

$$f(x) = f_0 + < a, x >_H + \frac{1}{2} < x, Ax >_H \quad (1)$$

Where A is an nxn symmetric positive definite matrix operator on the Hilbert space H. α is a vector in H and f_0 is a constant term.

Let us also consider what is termed conjugate descent with f. With conjugate descent, it is assumed that a sequence $\{p_i\} = p_0, p_1, \ldots, p_k, \ldots$ is available with the members of the sequence conjugate with respect to the positive definite linear operator A.

By conjugate with respect to A we mean that

$$< p_i, Ap_i >_H = \begin{cases} \neq 0, & if\ i \neq j \\ = 0, & if\ i = j \end{cases}$$

In the case here, A is assumed positive definite so $< p_i, Ap_i >_H > 0$.

2 Steps involved in conventional conjugate gradient method algorithm (CGM)

With conjugate gradient descent, as with any descent method, the first **Step** simply involves guessing the first sequence x_0. The remaining members of the sequence are then calculated as follows:

Step 2

$p_0 = -g_0 = -(a + Ax_0)$

(p_0 is the descent direction and g_0 is the gradient of $F(x)$ when $x = x_0$)

Step3

$x_{i+1} = x_i + \alpha_i p_i,\ \alpha_i = < g_i, g_i >_H / < p_i, Ap_i >_H$

$g_{i+1} = g_i + a_i Ap_i;$

α is the step length

$p_{i+1} = -g_{i+1} + \beta_i p_i;\quad \beta_i = < g_{i+1}, g_{i+1} >_H / < g_i, g_i >_H$

Step 4

If $g_{i=0}$ for some i terminate the sequence else, set $i = i + 1$ and go to step 3.

If $H = \mathbb{R}^n$, operator A turns out to be a positive definite symmetric matrix operator and for this case can easily be computed. The CGM algorithm has a well worked out theory with an elegant convergence profile Bertsekas (1973). It has been proved that the algorithm converges, at most, in n iterations in a well posed problem and the convergence rate is given as:

$$E(x_n) = \left(\frac{1 - \frac{m}{M}}{1 + \frac{m}{M}} \right)^{2n} E(x_0)$$

Where m and M are smallest and spectrums of matrix A, respectively.

That is, for an n dimensional problem, the algorithm will converge in at most n iterations. This Conventional CGM algorithm, due to Hestene and Stiefel, was originally designed for the minimization of a quadratic objective functional. CGM appears to be the most popular among the descent iterative methods because of its simplicity, elegance and the convergence property in handling quadratic functional. With the above qualities in mind, Ibiejugba, Onumanyi (1984) worked on the possibility of applying CGM algorithm to continuous optimal cost functional. Their own version of CGM is called the Extended Conjugate Gradients Method (Hestenes, 1969; Omolehin, 1985).

We know that if $H = \mathbb{R}^n$ the operator A turns out to be a positive definite symmetric constant matrix and A can easily be computed in quadratic functional so that CGM Algorithm will be appropriate for the solution but if $H \neq \mathbb{R}^n$ the situation becomes very difficult (Yosida, 1984). This is what motivated Ibiejugba (1980), knowing explicitly the control operator Λ so as to compute AP_i needed for the step length α_i. Ibiejugba (1980) successfully constructed this control operator A. He then used the constructed operator to formulate his own method called the Extended Conjugate Gradient Method (ECGM). The formalism of CGM was adopted in the construction of the Extended Conjugate Gradient Method (EGCM) algorithm. It was originally formulated to solve problems in the following class:

Minimize $$\int_0^{\delta}\{x^T(t)px(t) + u^T(t)qu(t)\}dt \tag{2}$$

Subject to

$$\dot{x}(t) = cx(t) + du(t)$$
$$0 \leq t \leq \delta \quad (\delta;\text{given});$$

$x^T(t)$ denotes the transpose of $x(t)$,

$\dot{x}(t)$ stands for the first derivative of $x(t)$ with respect to t.

$x(t)$ is the $nx1$ – state vector,

$u(t)$ is the $qx1$ – control vector applied to the system at time t.

c and c are nxn, nxq constant matrices respectively,

while p and q are symmetric , positive definite , constant square matrices of dimensions n and q respectively.

The control operator A associated with Bertsekas (1973) satisfies equation (3).

$$\langle Z,Z\rangle_K = J(x,u,\mu) = \int_o^{\delta}\{x^T(t)Px(t) + u^T(t)qu(t)dt\} + \mu\int_o^{\delta}\{\|\dot{x}(t) - cx(t) - du(t)\|^2\}dt \tag{3}$$

By transforming (2) into an unconstrained optimal control problem, μ is a penalty constant greater than zero, K is defined by $K = H_1[0,\delta] x L_2^q[0,\delta]$ Where $H_1[0,\delta]$ denotes Sobolev space of the absolutely continuous functions $x(\cdot)$ square integrable over the closed interval $[0,\delta]$. $L_2^q[0,\delta]$ stands for the Hilbert space consisting of the equivalence classes of square integrable functions from $[0,\delta]$ into $\mathbb{R}^2$, with norm denoted by $\|\cdot\|_E$ defined by $\|u\| = \left\{\int_0^\delta \|u(t)\|^2 dt\right\}^{\frac{1}{2}}$ and with scalar product conventionally denoted by $\langle . \,, . \rangle$ and defined by $\langle u_1, u_2 \rangle = \int_0^\delta \langle u_1(t), u_2(t) \rangle_E dt$. Furthermore, $\|\cdot\|_E$ and $\langle . \,, \; . \rangle_E$ denote the norm and scalar product in Euclidean q-dimensional space. The result of the constructed operator A is as follows:

$$AZ(t) = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} = \begin{bmatrix} A_{11}x(t) & A_{12}u(t) \\ A_{21}x(t) & A_{22}x(t) \end{bmatrix}, \qquad (4)$$

Where A_{11} is such that

$$\begin{aligned}(A_{11}x)(t) = {} & -\mu(\dot{x}(0) - cx(0))sinht + \mu \int_0^t (\dot{x} - cx)\,(s)cosh(t-s)ds \\ & - \int_0^t [(q + \mu c^T c)x - \mu cx](s)\,sinh(t-s)ds \\ & + \left\{ -\mu sinhT(x10) + cx(0)] + \mu \int_0^T (x \dot{-} cx)\,(s)cosh(T-s)ds \right. \\ & \left. - \int_0^T [(q + \mu c^T c)x - \mu c\dot{x}]\,(s)sinh(T-s)ds \right\} exp(T), \qquad 0 \le t \le T\end{aligned}$$

$$(A_{12}u)(t) = cu(t) + \mu d^T du(t), \qquad 0 \le t \le T$$

$$(A_{21}x)(t) - \mu c^T dx(t) + \mu du(t), \qquad 0 \le t \le T$$

$$\begin{aligned}(A_{22}d)(t) = {} & \mu du(s)Sinht - \mu \int_0^t du(s)Cosh(t-s)ds + \int_0^T c^T dSinh(t-s)ds \\ & + \{\mu du(0)Sinht \\ & \int_0^T \mu du(S)Cosh(t \;\; s)ds \;|\; \mu \int_0^T du(S)Sinh(t \;\; s)ds\}exp(T) \quad 0 \le t \le T\end{aligned}$$

The desired AP_i can now be applied when calculating $g_{i+1} = g_i + \alpha_i AP_i$ in the CGM algorithm which will now allow us to exploit the simplicity of the CGM algorithm. Therefore the ECGM algorithm is given as follows:

Step 1

It involves guessing the first sequence x_0. The remaining members of the sequence are then calculated as follows:

Step 2

$$p_0 = -g_0$$

(p_0 is the descent direction and g_0 is the gradient of the cost functional when $x = x_0$).

Step3

$$x_{i+1} = x_i + \alpha_i \, P_{x,i}$$
$$u_{i+1} = u_i + \alpha_i \, P_{u,i}$$
$$g_{x,i+1} = g_{x,i} + \alpha_i \, AP_{x,i}$$
$$g_{u,i+1} = g_{u,i} + \alpha_i \, AP_{u,i}$$
$$P_{x,i+1} = -g_{x,i+1} + \beta_i \, P_{x,i}$$
$$P_{u,i+1} = -g_{u,i+1} + \beta_i \, P_{u,i}, \qquad \text{where}$$
$$\alpha_i = \frac{\langle g_i, g_i \rangle}{\langle P_i, AP_i \rangle} \text{ and } \beta_i = \frac{\langle g_{i+1}, g_{i+1} \rangle}{\langle g_i, g_i \rangle}$$

$$\begin{aligned} AP_i = \Bigg(& -\mu \sinh t (J_{x,0} - cp_{x,0}) + \mu \int_0^t (cj_i - cp_{x,i}) \cosh(t-s)\,ds - \int_0^t [q + \mu c^T c] P_{x,i} \\ & - \mu c J_{xi} \sinh(t-s)\,ds \\ & + \Big\{ \mu \sinh\sigma [-j_{x,0} - cp_{x,0}] \\ & + \mu \int_0^T (J - cp_{x,i}) \cosh(\sigma - s)\,ds \\ & - \int_0^t [q + \mu c^T c] p_{x,i} - \mu c j_{x,i}] \sinh(\delta - s)\,ds \Big\} \exp(\sigma) + cu_i(t) \\ & + \mu d^T du(t);\ \mu [c^T dp_{x,i} - dJ_{x,i}] \\ & + \mu \Big\{ Du(0) \operatorname{Sinh} t - \int_0^t Du_i(s) \cosh(\delta - s)\,ds \\ & + \mu \int_0^\delta c^T Du_i(s) \operatorname{Sinh}(\sigma - s)\,ds \Big\} \exp(\delta) \Bigg), \end{aligned}$$

and where we have used the following notations:

$J_i = J(x_i, u_i, \mu)$
$J_{x,i} = J_x(x_i, u_i, \mu)$
$J_{u,i} - J_u(x_i, u_i, \mu)$
$p_{x,i} = p_x(x_i, u_i, \mu)$
$p_{u,i} = p_u(x_i, u_i, \mu)$
$p_{u,i} = p_u(x_i, u_i, \mu)$

$$p_x(x_i, u_i, \mu) = \int_o^t J_x(x_i(s), u_i(s), \mu)ds$$

and

$$p_u(x_i, u_i, \mu) = \int_o^t J_u(x_i(s), u_i(s), \mu)ds$$

Step 4

If g_i satisfies the tolerance, for some i terminate the sequence else, set $i = i + 1$ and go to step 3.

Basically, the ECGM algorithm was formulated by Ibiejugba and Onumanyi (1984) to solve problems in quadratic cost functions of the type

Minimize $\int_0^{\delta}\{ax^2(t) + bu^2(t)\}dt$ (5)

Subject to
$\dot{x}(t) = cx(t) + du(t)$

However, the algorithm is extended to the solution of Reaction Diffusion problem.

3 Euler – Lagrange Approach to Reaction Diffusion Control Problem

Let us consider the Reaction Diffusion Control problem:

Minimize $J(v, u) = \int_0^T\{av^2(t) + bu^2(t)\}dt$ (6)

Subject to

$$\dot{v}(t) - \dot{u}(t) = cv(t) + du(t) \quad (7)$$

Consider the Euler – Lagrange equivalent to equations (6) and (7):

$$J(u,v,\lambda) = \int_0^T F(u,v,\lambda)dt \quad (8)$$

Where λ is the augmented Lagrangian and

$$F(u,v,\lambda) = av^2(t) + bu^2(t) + \lambda(\dot{u}(t) - \dot{v}(t) - cv(t) - du(t) \quad (9)$$

is the Euler-Lagrange equation for equation (8), then

$$\frac{\delta F}{\delta v} - \frac{d}{dt}\ \frac{\delta F}{\delta \dot{v}} = 0 \qquad \text{i.e} \quad 2av(t) - \lambda c + \dot{\lambda} = 0 \quad (10)$$

$$\frac{\delta F}{\delta u} - \frac{d}{dt}\ \frac{\delta F}{\delta \dot{u}} = 0 \qquad \text{i.e} \quad 2bu(t) - \lambda c - \dot{\lambda} = 0 \quad (11)$$

Adding equations (10) and (11), we obtain

$$\lambda = \frac{2av(t) + 2bu(t)}{c+d} \quad (12)$$

substitute (11) into (9), we get

$$2av(t) - (c/(c+d))\big(2av(t) + 2bu(t)\big) + (2a\dot{v}(t) + 2b\dot{u}(t))/(c - d) = 0$$

i.e.,

$2a\dot{v}(t) + 2b\dot{u}(t) = 2bcu(t) - 2adv(t)$, i.e.,

$$\dot{v}(t) + \frac{b}{a}\dot{u}(t) = 2bcu(t) - 2adv(t) \quad (13)$$

Adding equations (7) and (13), we obtain

$$v(t) = [(a + b\dot{u}(t) - (ad + bc)u(t)]/[ac - ad] \quad (14)$$

Substitute (14) into (13) and simplify, we get

$$\ddot{u}(t) - \frac{(bc^2 + ad^2)}{(a+b)}u(t) = 0 \quad (15)$$

Let

$$k = \frac{bc^2 + ad^2}{(a+b)}$$

Equation (15) becomes

$$\ddot{u}(t) - ku(t) = 0 \quad (16)$$

Observe that equation (16) ordinary equation and $a, b > 0$ (from our hypothesis) $c^2, d^2 > 0$ therefore constant k is positive.

The general solution to equation (16) is

$$u(t) = Ae^{\sqrt{k}t} + B\bar{e}^{\sqrt{k}t}$$

or $u(t) = \alpha \cosh(\sqrt{k}t) + \beta \sinh(\sqrt{k}t),$ (17)

Where α and β are arbitrary constants to be determined using the initial condition (5) in equation (17) to obtain

$$u_o = \alpha \quad (18)$$

Substitute the value of α in (17) to obtain

$$u(t) = u_0 \cosh(\sqrt{k}t) + \beta \sinh(\sqrt{k}t)$$

To determine the value of β appearing in the last equation we make use of free-end condition (9).

From (9) and (12), we obtain

$$\lambda = \frac{\delta F}{\delta \dot{u}} \Big|_{t=T} = 0,$$

i.e.,

$$2av(T) + 2bu(T) = 0$$

$$av(T) + bu(T) = 0$$

but $u(T) - v(T) - 0,$

using (18) in (14), we obtain

$$0 = (a + b)\ddot{u}(T) - (ad + bc)u(T) \quad (19)$$

Substitute the value of $\dot{u}(T)$ and $u(T)$ into equation (19) and simplify to obtain

$$\beta(a + b)\sqrt{k}\cosh\sqrt{k}T - (ad + bc)\sinh\sqrt{k}T\} =$$

$$(ad + bc)u_o \cosh\sqrt{k}T - (a + b)u_o \sqrt{k}\sinh\sqrt{k}t. \quad (20)$$

From (20), we get

$$\beta = \frac{(ad+bc)u_0\cosh\sqrt{k}T - (a+b)u_0\sinh\sqrt{k}}{(a+b)\sqrt{k}\cosh\sqrt{k}T - (ad+bc)\sinh\sqrt{k}T} \quad (21)$$

and substituting for the value of β given by (21) into (18), we obtain

$$u(t) = u_o \cosh kt + \left\{\frac{(ad+bcu_o\cosh\sqrt{k}t-(a+b)u_o\sqrt{k}T}{(a+b)\sqrt{k}\cosh\sqrt{k}T-(ad+bc)\sinh\sqrt{k}T}\right\} \sinh\sqrt{kt} \quad (22)$$

Differentiate (22) to obtain $\dot{u}(t)$, also substitute for the values of $u(t)$ and $\dot{u}(t)$ in equation (14) we obtain

$$v(t)\frac{1}{a(c-d)}(a+b)[u_o k\sinh\ kt\left\{\frac{(ad+bu)u_o\cosh\sqrt{kT}-(a+b)u_o\sqrt{k\sinh\sqrt{kT}}}{(a+b)\sqrt{k\cosh\sqrt{kT}}-(ad+bc)\sinh\sqrt{kT}}\right\}\times\{\sqrt{k}\cosh\sqrt{k}T]-(ad+bc)[u_o\cosh\sqrt{k}T+g(T)\sinh\sqrt{k}t]\} \quad (23)$$

where

$$g(T) = \left\{\frac{(ad+bu)u_o\cosh\sqrt{k}T-(a+b)u_o\sqrt{k}\sinh\sqrt{k}T}{(a+b)\sqrt{k}\cosh\sqrt{k}T-(ad+bc)\sinh\sqrt{k}T}\right\}$$

Therefore equations (22) and (23) give the solution of the objective functional.

4 ECGM approach to Reaction Diffusion Problem

Recall that our reaction Diffusion control problem

Minimize

$$\int_o^T\{v_1^2(t)+v_2^2(t)+\cdots+v_N^2(t)+u_1^2(t)+u_2^2(t)+\cdots+u_N(t)\}dt \quad (24)$$

Subject to

$$\dot{u}_i(t)-\dot{v}_i(t)=\bar{c}v_i(t)+\bar{D}u_i(t)$$

Where,

$$\bar{C} - D_2\pi^2 i^2 + d + b, \qquad i - 1,2,\ldots\ldots\ldots\ldots, N$$
$$\bar{D} = a - D_1\pi^2 i^2 - C, \qquad i = 1,2,\ldots\ldots\ldots\ldots, N$$

One dimensional control problem (Gland, 1976) is considered in the following form:

Minimize $\int_o^T\{v^2(t)+u^2(t)\}dt$

Subject to

$$\dot{u}(t) - \dot{v}(t) - \bar{C}u(t) + \bar{D}v(t). \quad (25)$$

Example [1]: In this numerical example, values are assigned to the variables of the control problem in the following form:

Let

$$T = 1, \quad D_1 = 1, D_2 = 1, \quad d = 1, \quad b = 1, \quad a = 3 \quad \text{and } c = 1$$

Substituting these last values in equation (25) we obtain

Minimize

$$\int_o^1 \{v^2(t) + u^2(t)\}dt \qquad (26)$$

Subject to $\dot{u}(t) - \dot{v}(t) = (2 + \pi^2)v + (2 - \pi^2)u(t)$

The last equation is now transformed to an unconstrained penalized functional with the introduction of a penalty parameter μ in the following form:

Minimize

$$\int_o^1 \{v^2(t) + u^2(t) + \mu \parallel \dot{u}(t) - \dot{v}(t) - (2 + \pi^2)v(t) - (2 - \pi^2)u(t) \parallel^2\} \, dt \qquad (27)$$

ECGM is now used for the minimization of problem (7). The numerical result is displayed in Table [1]. It also displays the performance of ECGM with the analytical solution to our control problem through Euler-Lagrange $(E - L)$ approach. The computer listing is given in Appendix.

ANALYSIS OF TABLE [1]

Since this is the first attempt to transform reaction diffusion problem into optimization, we went further to find the analytical solution of the resulting reaction diffusion control problem via Euler-Lagrange method. The analytical solution is composed favorably with the performance of the ECGM algorithm when used as a method of solution to the control problem (Di Pillo and Grippo, 1972; Powell, 1969).

For example, at $\Delta t = 0.4$, we have at the second iteration,

$$\text{OBJF/ECGM} = 4.60542\text{E} - 12$$
$$\text{OBJF/E} - \text{L} = 2.12620\text{E} - 13$$

These two values are favorably comparable, research will still continue in this area. Appendix is the programme listing for the Euler-Lagrange method.

Table 1. The numerical solution to the reaction diffusion control problem with ECGM algorithm and Euler-Lagrange approach

N=1	$\Delta t = 0.1$ $\mu * (t) = 0.330685606$		$\Delta t = 0.2$ $\mu * (t) = 0.388755981$		$\Delta t = 0.3$ $\mu * (t) = 0.446797229$		$\Delta t = 0.4$ $\mu * (t) = 0.562968249$		$\Delta t = 0.5$ $\mu * (t) = 0.755581395$	
ITRN	OBJF/ECGM	0BJF/E-L	OBJF/ECGM	0BJF/E-L	OBJF/ECGM	0BJF/E-L	OBJF/ECGM	0BJF/E-L	OBJF/ECGM	0BJF/E-L
1	2E – 11	3.20982E-17	4E-1	8.47424E-16	6E-1	1.43336E-14	8E-1	2.12620E-13	1E-10	2.95325E-12
2	7.2909E-13		1.64402E-12		2.85961E-12		4.60542E-12		7.40762E-12	
3	3.322575E-13		1.07363E-12		5.76730E-12		5.76730E-12		1.94914E-11	
4	3.04532E-13		1.21488E-12		8.10635E-12		8.10635E-12		3.09329E-11	
5	2.64007E-13		1.87394E-12		1.71462E-11		1.71462E-11		6.48852E-11	

Table 1. (Continue)

N=1	$\Delta t = 0.6$		$\Delta t = 0.7$		$\Delta t = 0.8$		$\Delta t = 0.9$	
ITRN	OBJF/ECGM	0BJF/E-L	OBJF/ECGM	0BJF/E-L	OBJF/ECGM	0BJF/E-L	OBJF/ECGM	0BJF/E-L
1	1.2E-10	3.93748E-11	1.4E-1	5.10386E-10	1.6E-10	1.48073E-09	1.8E-10	8.10046E-08
2	7.40762E-12		2.64391E-11		9.38138E-11		2.58062E-09	
3	1.94914E-11		1.14150E-09		6.66133E-09		3.74576E-06	
4	3.09329E-11		1.101902E-09		6.681346-09		3.76303E-06	
5	6.48851E-11		1.08674E-09		6.70421E-09		3.77983E-06	

OBJF/ECGM = the value of the objective functional through the Extended Conjugate Gradient Method at time t,
OBJF/E-L = the value of the objective functional via Euler-Lagrange approach,
ITRN = the iteration number,
$u*(t)$ *= the optimal penalty parameter,*
Δt *= the time descritization*

5 Conclusion

Since it has been proved that the class of control problem in equation (24) admits unique solution, therefore the ECGM and Euler- Lagrange methods provide suitable trial solutions. To the best knowledge of the authors, this is the first time of using ECGM to obtain the solution of reaction diffusion control problem.

References

Bertsekas, D. P. (1973). Convergence rate of the penalty and Multipliers Methods. Proceedings of the IEEE Conference on Decision and Control, San Diego, Califonia, 260-264.

Di Pillo, G., Grippo, L. (1972). A Computing Algorithm for the Epsilon Method to Identification and Optimal Control Problems. Ricerche di Automatica, 3, 54-77.

Gland, T. (1976). Constrained Optimization using Multiplier Methods with Applications to Control problems. University of Lund, Lund, Sweden, report No. 7603.

Hestenes, M. R. (1969). Multipliers and Gradient Methods. Journal of Optimization Theory and Application, 4, 303-320.

Ibiejugba, M. A. (1980). Computing Method in optimal control. University of Leeds, Leeds, England PhD Thesis.

Ibiejugba, M.A., Onumanyi, P. (1984). On a Control Operator and some of its Applications. Journal of Mathematical Analysis and Applications 103, 31-47.

Omolehin, J.O. (1985). Numerical Experiments with the Extended Conjugate Gradient Method Algorithm. University of Ilorin, Ilorin, Nigeria, Unpublished M.Sc. Thesis.

Powell, M. J. D. (1969). A method for Nonlinear constraints in Minimization problems. Optimization Edited by Fletcher, Academic Press, NewYork.

Yosida, K. (1984). Functional Analysis. Springer, International Student Edition, Springer-Verlag Berlin.

APPENDIX

```
20 REM OPTIMAL PENALTY PARAMETER (μ) FOR R/D SYSTEM IS CONSIDERED
30 REM ECGM WAS USED FOR R/D SYSTEM
50 REM N IS THE DIMENSION OF THE STATE AND THE CONTROL VECTORS
60 N=1; K=N
70 DIM U(K), X(K), D(K), C(K), PX(K), PU(K), JK(K),JU(K), EPI(K), EP2(K), PXO(K)
80 DIM PUO(K), GX(K), GU(K),ALPX(K), ALPU(K),BITAX(K),BITAU(K),GXO(K)
90 DIM GUO(K),DK(K),UPX(K),UPU(K),UPPX(K),UPPU(K),UPGX(K),UPGU(K)
95 DIM JXO(K), JUO(K)
100 REM THE CONSTANTS IN THE CONSTRAINT ARE RE-DEFINE AS FOLLOWS
110 REM a=A1, b=B1, c=C1, d=DS, A AND B ARE THE COEEFFICIENTS IN THE
115 REM COST FUNCTIONAL
120 READ A, B, A1, B1, DS, C1, U0
130 REM THE DATATO READ ARE AS FOLLOWS: 1,1,3,1,1,1,0.001
140 REM U0 IS THE ARBITRARY CONSTANTS IN THE ANALYTICAL SOLUTION
150 K1 = 0.001:K2 = K1: ITERA = 0: T = 0:OB2 = 0:CS1 = 0
160 FOR I = 1 TO N
170 X(I) = 0:U(I) = 0:D(I) = 0:C(I) = 0: PX(I) = 0:PU(I) = 0: JX(I) = 0: JU(I) = 0:EP1(I) = 0
180 EP2(I) = 0:GX(I) = 0;GU(I) = OALPHAX(I) = 0:ALPHAU(I) = 0:BITAX(I) = 0
190 BITAU(I) = 0:UPX(I) = 0:UPU(I) = 0:UPPU(I):UPPX(I) = 0;UPGX(I) = 0 UPGU(I) = 0
200 JXO(I) = 0:JUO(I) = 0:DK(I) = 0:PXO(I) = 0:PUO(I) = 0
210 NEXT I
220 T = T + 0.0: ITERA = 0
230 PRINT
240 PRINT,"T = ",T
250 FOR I =I TO N
260 X(I) = K1:U(I) = K2
270 NEXT I
280 IF T > 1 GOTO 970
290 FOR I = 1 TO N
300 C(I) = ( ( 3.142 *I)^2) + 4:D(I)  = - ( (3.142*I)^2):DK(I) =(C(I)^2 +D(I)^2)/2
310 SK = SQR(DK(I) ):SKT = SK*T:ST = 0.5*(EXP(SK)-EXP(-SK))
320 CT = .5*(EXP(SK)-EXP(-SK) ):B1 = 1/(B(I)-C(I) ):B2 = 2:B3 = D(I) +C(I):A1 = UO*CT
330 STT = .5*(EXP(SKT)-EXP(-SKT) ):CTT = .5 * (EXP(SKT) + EXP(-SKT)
340 A2 = (B3*UO*CT-2*UO*SK*ST)/(2*SK*CT-B3*ST):A3 = U0*SK*ST
350 UDT = A1+A2*SK*CTT:UD = A1+A2*STT
360 XD = B1*(B2*( A3+A2*DK(I)*STT)-B3*(A1+A2*SK*CTT) ) )
370 CEX(I) = C(I) * (B1*(B2* (A3+A2*SK*CTT) ) )
380 UDD =  D(I) *(A1*A2*STT)
390 DN = (B3*(UDT-XD-CEX-UDD)
395 REM THE CALCULATION OF THE OPTIMAL PARAMETER(AM) NOW FOLLOWS
400 AM = (B1*(B2*(A3+A2*SK*CTT)-B3*(A1+A2*STT) ) ) +(A1+A2*STT) )/DN
410 JX(I) = 2*X(I)-2*AM*C(I)*(-C(I)*X(I)-D(I)*U(I) )
420 JU(I) = 2*U(I)-2*AM*D(I)*(-C(I)*X(I)-D(I)*U(I) )
430 PX(I) = 2*X(I)-2*T+2*AM*C(I)*(C(I)*X(I)+D(I)*U(I) ) + T
440 PU(I) = 2*U(I)-2*T+2*AM*D(I)*(C(I)*X(I)+D(I)*U(I) ) + T
```

```
450 GX(I) = JX(I):GU(I)=JU(I)
460 NEXT I
470 ITRE = ITERA+1:OB1 = 0:OB2 = 0:CS1 = 0:CS = 0
480 IF ITERA > 10 GOTO 220
490 PRINT:"ITERA=ITERA";ITERA
500 FOR I = 1 TO N
510 C(I) = (3.142*I)^2+4:D(I) = -(3.142*I)^2:DK(I) = (C(I)^2+D(I)^2)/2
520 REM SK = SQR(DK(I)):SKT = SK*T
530 SK = SQR(DK(I) ):SKT = SK*T:ST =.5*(EXP(SK)-EXP(-SK))
535 CT =.5 *(EXP(SK)+EXP(-SK))
540 B1 =1/(C(I)-D(I)):B2 = 2:B3 = D(I)+C(I):A1 = UO*CT
550 STT =.5*(EXP(SKT)-EXP(-SKT)):CTT = .5*(EXP(SKT)+EXP(-SKT)
560 A2 = (B3*UO*CT-2*UO*SK*SK)/(2*SK*ST-B3*ST):A3=UO*SK*ST
570 UDT = A1+A2*SK*CTT:UD = A1+A2*STT
580 XD =B1*(B2*(A3+A2*DK(I)*STT)-B3*(A1+A2*SK*CTT) ) )
590 CEX = C(I)*(B1*(B2*(A3+A2*SK*CTT)-B3*(A1+A2*SK*CTT) ) )
600 UDD = D(I)*(A1*A2*STT)
610 DN = -(B3*(UDT-XD-CEX-UDD) )
620 AM =( (B1*(B2*(A3+A2*SK*CTT)-B3*(A1+A2*STT) )/DN
630 JXO(I)  = 2*K1+2*AM*C(I)*(C(I)*K1+D(I)*K2)
640 JUO(I) = 2*K1+2*AM*D(I)*(C(I)*K1+D(I)*K2)
650 GXO(I) =JXO(I):GUO(I) = JUO(I):PXO(I) = -GXO(I):PUO(I) = -GUO(I)
660 REM THE CONSTRUCTION OF EPiNOW FOLLLOWS
670 STI = 0.5*(EXP(1)-EXP(-1) ):ST2 = 0.5*(EXP(1)+EXP(-1) )
680 ST3 =0.5 *(EXP(T)-EXP(-T) ):ST30.5*(EXP(T)+EXP(-T) )
690 AMC = A+(AM*(C(I)^2) ):BMD = B+(AM*(D(I)^2) )
700 EP = -AM*ST3*(JXO(I)+(C(I)*PXO(I) ) ):EC = AM*(JX(I)+PX(I) ) * ST3
710 EP3 = (AMC*PX(I) + (AM*C(I0*JX(I) ) )* (-1+ ST4)
715 EP4  = AM*ST3*(JXO(I)+(C(I)*PXO(I) ) )
720 EP5 = -AM*(JX(I) + (C(I)*PX(I) ) )*ST1
725 EP6 = (AMC*PX(I) +  (AM*C(I)* JX(I) ) )*(-1+ST2)
730 EP7 = -AM*ST3*(D(I)*PUO(I)-JUO(I):EP8=-AM*(PU(I)-JU(I) ) *ST3
740 EP9 = -AM*C(I)*(D(I)*PU(I)-JU(I):EP10=AM*ST1*(D(I)*PUO(I)-JUO(I) )
750 EP11= - AM*(D(I)*PU(I)-JU(I) ) *ST1
755 EP12= AM*C(I)*(D(I)*PU(I)-JU(I) ) *(-1+ ST4)
760 EP13= AM*ST3*JXO(I):EP14=AM*JX(I)*ST3
770 EP15= AM*(D(I)*JX(I)-C(I)*PX(I)*(-1+ST4):EP16=ST1*AM*JXO(I)
780 EP17 = AM*JX(I)*ST1:EP18=AM*(D(I)*JX(I)-C(I)*PX(I)*(-1=ST2)
790 EP19=AM *ST3(JUO(I)-(D(I)*PU0(I) ) ):EP20=AM*(JU(I) – D(I) *PU(I) ) * ST3
800 EP21 = (BMD*/PU(I)-(AM*D(I)*JU(I) ) )*(-1+ST2)
805 EP22= AM*ST1*(JUO(I)-(D(I)*PUO(I) ) )
810 EP23 = -AM*(JU(I)-(D(I)*PU(I) ) )*ST1
815 EP24= (BMD*PU(I) – (AM*D(I)*JU(I) ) )*(-1 + SP2)
820 PK1 = EP+EC+(EP3 +EP4 + EP5+EP6)*EXP(1)
825 PK2 = (EP7 + EP8 + EP9) * EXP(1) + EP10 + EP11 + EP12
830 PK3 = EP13+EP14 + EP15 + EP16 + EP17 + EP18
835 PK4 = (EP19 + EP20 + EP21 + EP22 + EP23 + EP24)*EXP(1)
840 EP1(I) = =PK1 + PK2: EP(2) = PK3 + PK4
850 ALPX(1) = (6X(I) ^ 2) /(PX(I) *EP1(I) ):ALPU (I )=  (6U(I) ^ 2 )/(PU(I) + EP2(I) )
```

```
860 UP6X(I) =6X(I) + ALPX(I) *PX(I):UP6U(I)=6U(I) + ALPU(I) *PU(I)
870 UPPX(I) = UPGX(I) + BITAX(I) * PX(I) : UPPU(I) = -UPGU(I) + BITAU(I) * PU(I)
880 UPX(I) = X(I) + ALPX(I) * PX(I):UPU(I) = U(I) + ALPU(I) * PU(I)
890 PX(I) = UPPX(I):PU(I)=UPPU(I):6X(I) = UPGX(I) :GU(I) = UPGU(I)
900 X(I) = UPX(I): U(I) = UPU(I)
910 OB1= OB1+(X(I) ^ 2):OB2=OB2+(U(I) ^ 2)
915 CS1 = CS1 + AM * ( (C(I) *X(I) + D(I) * U(I) ) ^ 2 )
920 PRINT , “X(“ ; I ; ”)=” ; X(I) ,”U(“ ; I ; ”) =” ; U(I )
925 PRINT , “GX(“ ; I ; ”)=” ; GX(I) ,”GU(“ ; I ; ”) =” ; GU(I )
930 NEXT I
940 OB = (OB1+OB2)*T + CS1 * T: CS = CS * T
950 PRINT,”AM=”;AM,”CS=”;CS,”OB=”;OB
960 GOTO 470
965 DATA 1,1,3,1,1,0.001
970 END
```