Journal of the Nigerian Association of Mathematical Physics,

Volume 8 (*November 2004*).

A discretized algorithm for the solution of a constrained, continuous quadratic control problem.

S. A. Olorunsola

Department of Mathematical Sciences

University of Ado-Ekiti,

Ado-Ekiti, Ekiti State, Nigeria.

and

O. Olotu

Department of Industrial Mathematics

The Federal University of Technology,

Akure, Ondo State, Nigeria.

Abstract

Numerical solution techniques such as Function space algorithm (FSA), Extended conjugate gradient method (ECGM) and Imbedding extended conjugate gradient method (MECGM) are common techniques for solving optimal control problems. However, these techniques are computationally expensive and iteratively time consuming. In this paper, a Discretized constrained algorithm (DCA) with an associated operator which replaces the integral features of these techniques by a series of summation is developed. Illustrative examples are presented. The results obtained show that the Discretized constrained algorithm (DCA) is much more accurate and more efficient than some of these techniques, particularly the FSA.

pp 295 - 300

1.0 **Introduction**

Some schemes, Function space algorithm (FSA), Extended conjugate gradient method (ECGM) and Multipliers extended conjugate gradient method (MECGM) developed by Ibiejugba and Onumanyi [4] though based on Fletcher and Reeves [1] ideas have been used to solve quadratic control problems constrained by ordinary differential equation of the linear type or evolution equation of the retarded type. This new scheme, discretized constrained algorithm (DCA) has reduced the computational rigour characterized of the old schemes by discretizing, thus replacing the integral features of the old schemes and constructing an associated operator for the discretized problem. The objective of this paper is to solve two problems with penalty constant μ assuming values 0.5(2.5)0.5 for each cycle of iteration terminated by the stopping rule of the conjugate gradient method and using qbasic-programming language to evaluate the efficiency of this new scheme compared to the old schemes

2.0 **Generalized problem**

$Min\int_0^T \left(ax^2(t)+bu^2(t)\right)dt$ subject to $\dot{x}(t)=cx(t)+du(t)$ $0 \le t \le T$

$$X(0) = X_0 = 0 \qquad a, b, c, d \text{ are in } R \qquad (2.1)$$

The constrained problem (1) can be turned into unconstrained problem via the penalty method (2.1) The problem may be put in the following equivalence form;

$\langle Z, AZ\rangle_H = Min_{(x,U)}\int_0^T \left\{ax^2(t)+bu^2(t)+\mu\left\|x\&(t)-cx(t)-dux(t)\right\|^2\right\}$, $\mu \ge 0$ the penalty (2.2)

constant *t*.

2.1 **Discretization**

By discretizing (2), subdivide [0,T] into *n* equal intervals at mesh points $x_0 \;\pi\; x_1, \Lambda\;, x_{n-1}, x_n$ where *n* is the number of partition points chosen arbitrarily, thus having (*n* + 1) partition points, with $x_j = j{*}\Delta_j = 0,1,2\ldots n$, and $\Delta j = \Delta_k$ is the fixed length of each subinterval for $j = k$ or not. By $j{*}\Delta_{j,}$ it means j multiplied by Δ_j. Let $t_0 = 0$ and $t_k = \sum_{j=1}^{k} \Delta j \; t_{n-1} = T$, $k = 1,2,3,\Lambda\;, n$,

$$x(k) = x_k(t_{k,}), \quad u(k) = u_k(t_k), \quad k = 0,1,2,\Lambda\;, n$$

By Euler's scheme or finite difference method,

$$\dot{X}(k) = \frac{(X(k+1)-X(k))}{\Delta_k}, \; k = 0,1,\Lambda\;, N-1$$

$$\dot{X}(t) = cx(t) + du(t)$$

$$\frac{(x(k+1)-x(k))}{\Delta_k} = cx_k(t_k) + du_k(t_k) \qquad (2.3)$$

$$X(0) = 0$$

We then have the discretized generalized problem in the form;

$$\min J = \sum_{k=0}^{n} \Delta_k \left(a x_k^{\,2}(t_k) + b u_k^{\,2}(t_k)\right)$$

$$\text{subject to } \frac{(x(k+1) - x(k))}{\Delta_k} = c x_k(t_k) + d u_k(t_k) \qquad (2.4)$$

$$x(0) = 0$$

2.2 Discretized, unconstrained generalized problem

$$\text{Min } J(x,u,\mu) = \sum_{k=0}^{n} \left\{ \Delta_k \left(a x_k^{\,2}(t_k) + b u_k^{\,2}(t_k)\right) + \mu [\, x_{K+1}(t_{k+1}) - x_k(t_k) - \Delta_k c x_k(t_k) - d\Delta_k u_k(t_k)]^2 \right\}.$$

$$= \sum_{k=0}^{nn} \{ x_k^{\,2}(t_k)[a\Delta_k + \mu + \mu\Delta_k^{\,2} c^2 + \mu 2c\Delta_k\,] + u_k^{\,2}[b\Delta_k + \mu d^2 \Delta_k^{\,2}\,] + \mu x_{k+1}^{\,2}(t_k) + x_k(t_k) u_k(t_k)[2d\Delta_k \mu + 2cd\Delta_k^{\,2}\mu] \\ + x_{k+1}(t_k) x_k(t_k)[-2\mu - 2\mu c\Delta_k\,] + x_{k+1}(t_k) u_k(t_k)[-2\mu d\Delta_{k]}\,] \} \qquad (2.5)$$

Let $Z_k = \begin{pmatrix} x_k(t_k) \\ u_k(t_k) \end{pmatrix}$, and $y_k(t_k) = x_{k+1}(t_k)$. Let $\alpha_k = a\Delta_k + \mu + \mu\Delta_k^{\,2} c^2 + \mu 2c\Delta_k . \beta_k = b\Delta_k + \mu d^2 \Delta_k^{\,2}$,

$\lambda_k = 2\mu d\Delta_k$

$+ 2\mu c d\Delta_k^{\,2}$, $\delta_k = -2\mu(1 + c\Delta_k)$, $\rho_k = -2\mu d\Delta_k$ (2.5) becomes

$$\sum_{k=0}^{n} \left\{ \alpha_k x_k^{\,2}(t_k) + \beta_k u_k^{\,2}(t_k) + y_k^{\,2}(t_k)\mu + x_k(t_k) u_k(t_k)\lambda_k + y_k(t_k) x_k(t_k)\delta_k + y_k(t_k) u_k(t_k)\rho_k \right\} \qquad (2.6)$$

2.3 Construction of operator A

$$\langle Z_{k1}(t_k), AZ_{k2}(t_k) \rangle = \sum_{k=0}^{n} \{ \alpha_k x_{k1}(t_k) x_{k2}(t_k) + \beta_k u_{k1}(t_k) u_{k2}(t_k) + y_{k1}(t_k) y_{k2}(t_k)\varpi \\ + \lambda_k x_{k1}(t_k) u_{k2}(t_k) + \delta_k u_{k1}(t_k) x_{k2}(t_k) + \delta_k y_{k1}(t_k) x_{k2}(t_k) + \delta_k y_{k2}(t_k) x_{k1}(t_k) \\ + \rho_k y_{k1}(t_k) u_{k2}(t_k) + \rho_{kk} y_{k2}(t_k) u_{k1}(t_k) \qquad (2.7)$$

where $AZ_{K2}(t_k) = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_{k2} \\ u_{k2} \end{pmatrix} = \begin{pmatrix} A_{11} x_{k2} + A_{12} u_{k2} \\ A_{21} x_{k2} + A_{22} u_{K2} \end{pmatrix}$. Further simplifying (2.7), we have

$$\langle Z_{K1}(t_k), AZ_{K2}(t_k) \rangle_H = \sum_{k=0}^{n} \{ \alpha_k x_{K1}(t_k) x_{K2}(t_k) + \beta_k u_{K1}(t_k) u_{K2}(t_k) \\ + \mu [\, (\Delta_K \dot{x}_{K1} + x_{K1})(\Delta_K \dot{x}_{K2} + x_{K2})] + \lambda_K x_{K1} u_{K2} + \lambda_K u_{K1} x_{K2} + \delta_K (\Delta_K \dot{x}_{K1} + x_{K1}) x_{K2} \\ + \delta_K x_{K1} (\Delta_K \dot{x}_{K2} + x_{K2}) + \rho_K (\Delta_K \dot{x}_{K1} + x_{K1}) u_{K2} + \rho_K u_{K1} (\Delta_K \dot{x}_{K2} + x_{K2}) \} \qquad (2.8)$$

$$\langle Z_{K1}(t_k), AZ_{K2}(t_k) \rangle_H = \sum_{k=0}^{n} \{ \alpha_k x_{K1}(t_k) x_{K2}(t_k) + \beta_k u_{K1}(t_k) u_{K2}(t_k) + \mu\Delta_K^{\,2} \dot{x}_{K1}(t_k) \dot{x}_{K2}(t_k) + \mu\Delta_K \dot{x}_{K1}(t_k) x_{K2}(t_k) \\ + \mu\Delta_K x_{K1}(t_k) \dot{x}_{K2}(t_k) + \mu x_{K1}(t_k) x_{K2}(t_k) + \lambda_K x_{K1}(t_k) u_{K2}(t_k) + \lambda_K u_{K1}(t_k) x_{K2}(t_k) + \delta_K \Delta_K \dot{x}_{K1}(t_k) x_{K2}(t_k) \\ + \delta_K x_{K1}(t_k) x_{K2}(t_k) + \delta_K \Delta_K x_{K1}(t_k) \dot{x}_{K2}(t_k) + \delta_K x_{K1}(t_k) x_{K2}(t_k) + \rho_K \Delta_K u_{K2}(t_k) \dot{x}_{K1}(t_k) + \rho_K u_{K2}(t_k) x_{K1}(t_k) \} \qquad (2.9)$$

Setting $u_{K2}(t_k) = 0$, in (2.9) we have

$$\begin{pmatrix} A_{11} x_{K2} \\ A_{21} x_{K2} \end{pmatrix} = \begin{pmatrix} V_{11} \\ V_{21} \end{pmatrix}$$

$$\langle Z_{K1}(t_k), AZ_{K2}(t_k) \rangle_H = \sum_{k=0}^{n} \{ \alpha_k x_{K1}(t_k) x_{K2}(t_k) + \mu\Delta_K^{\,2} \dot{x}_{K1}(t_k) \dot{x}_{K2}(t_k) + \mu\Delta_K \dot{x}_{K1}(t_k) x_{K2}(t_k) + \mu\Delta_K x_{K1}(t_k) \dot{x}_{K2}(t_k) \\ + \mu x_{K1}(t_k) x_{K2}(t_k) + \lambda_K u_{K1}(t_k) x_{K2}(t_k) + \delta_K \Delta_K \dot{x}_{K1}(t_k) x_{K2}(t_k) + \delta_K x_{K1}(t_k) x_{K2}(t_k) + \delta_K \Delta_K x_{K1}(t_k) \dot{x}_{K2}(t_k) \delta_K x_{K1}(t_k) x_{K2}(t_k) \qquad (2.10)$$

$$\sum_{k=0}^{n} \{ x_{K1}(t_k)[\, \alpha_k x_{K2}(t_k) + \mu\Delta_K \dot{x}_{K2}(t_k) + \mu x_{K2}(t_K) + \delta_K x_{K2}(t_k) + \delta_K \Delta_K \dot{x}_{K2}(t_k) + \delta_K x_{K2}(t_k)] \\ + x_{K1}(t_k)[\mu\Delta_k^{\,2} \dot{x}_{K2}(t_k) + \mu\Delta_K x_{K2}(t_k) + \delta_K \Delta_K x_{K2}(t_k)] + u_{K1}(t_k)[\, \lambda_K x_{K1}(t_k)]] \} \qquad (2.11)$$

$$= \sum_{k=0}^{n} \{ x_{K1}(t_k) V_{11}(t_k) + \dot{x}_{K1}(t_k) \dot{V}_{11}(t_k) + u_{K1}(t_k) V_{21} \} \qquad (2.12)$$

Define $\Omega(t_k) = (\alpha_K + \mu + 2\delta_K)x_{K2}(t_k) + (\mu\Delta_K + \delta_K\Delta_K)\dot{x}_{K2}(t_k)$ and $f(t_k) = \mu\Delta_K{}^2\dot{x}_{K2}(t_k) + (\mu\Delta_K + \delta_K\Delta_k)x_{K2}(t_k)$ $A_{21}u_{K1}(t_k) = V_{21}(t_k) = \lambda_K x_{K2}(t_k)$. To obtain the component A_{11} x $K_1(t_k) = V_{11}(t_k)$, $\Omega(t_k) - V_{11}(t_k)$ and $f(t_k)$ - V_{11} (t_k) are both continuous functions on [0,T] , i.e.
$\Omega(t_k)$ is a function of $x_{k2}(t_k)$ and $\dot{x}_{k2}(t_k)$, which are both continuous. So also, $f(t_k)$ is a function of $\dot{x}_{k2}(t_k)$ and $x_{k2}(t_k)$. Hence, the difference of two continuous functions is continuous.

And choosing $x_{K1}(\bullet) \in D[0,T] \ni x_{K1}(0) = x_{K1}(T) = 0$, we then have

$$\int_0^T \{x_{K1}(t_k)[\Omega(t_k) - V_{11}(t_k)] + \dot{x}_{K1}(t_k)[f(t_k) - \dot{V}_{11}(t_k)]\}dt_k = 0 \qquad (2.13)$$

$f(t_k) - \dot{V}_{11}(t_k)$ is continuously differentiable on [0, T] with

$$\frac{d}{dt}[f(t_k) - \dot{V}_{11}(t_k)] = \Omega(t_k) - V_{11}(t_k) \qquad (2.14)$$

$$\dot{f}(t_k) - \ddot{V}_{11}(t_k) = \Omega(t_k) - V_{11}(t_k) \text{ or } \ddot{V}_{11}(t_k) - V_{11}(t_k) = \dot{f}(t_k) - \Omega(t_k)$$

$$\ddot{V}_{11}(t_k) - V_{11}(t_k) = q(t_k) = \dot{f}(t_k) - \Omega(t_k) \qquad (2.15)$$

with the initial conditions $V_{11}(0) = p_0$ and $\ddot{V}_{11}(0) = r_0$. Solving (2.15) by Laplace transform and letting $L\{V_{11}(t_k)\} = \hat{V}_{11}(s), L\{q(t_k)\} = Q(s)$, we have $S^2\hat{V}(s) - p_0 s - r_0 - \hat{V}_{11}(s) = Q(s)$, $\hat{V}_{11}(s) = \frac{Q(s)}{s^2 - 1} + \frac{p_0 s}{s^2} + \frac{r_0}{s^2 - 1}$ and taking the inverse of Laplace transform, we have

$$V_{11}(t_k) = \int_0^T q(s_k)\sinh(t_k - s_k)ds_k + p_0\cosh(t_k) + r_0\sinh(t_k) \qquad (2.16)$$

But
$$\Omega(T) - V_{11}(T) = 0 \qquad (2.17)$$

$$\Omega(0) - V_{11}(0) = 0 \qquad (2.18)$$

$\Omega(0) = \rho_0$. $\Omega(0) = (\alpha_k + \mu 2\delta_{kk})x_{k2}(0) + (\mu\Delta_k + \delta_k\Delta_k)\dot{x}_{k2}(0) = \rho_0$. From (2.17) $\Omega(T) = V_{11}(T)$,
$V_{11}(t_k) = \int_0^T q(s_k)\sinh(T - s_k)ds_k + [(\alpha_k + \mu + 2\delta_k)x_{K2}(0) + (\mu\Delta_K + \Delta_K\delta_K)\dot{x}_{K2}(0)]\cosh(T) + \tau_0\sinh(T)$

$= [(\alpha_K + \mu + 2\delta_K)\dot{x}_{K2}(T) + (\mu\Delta_K + \delta_K\Delta_K)\dot{x}_{K2}(T)]$. Therefore

$$\tau_0 = \frac{1}{\sinh(T)}\langle -\int_0^T q(s_k)\sinh(T - s_k)ds_k - [(\alpha_k + \mu + 2\delta_k)x_{K2}(0) + (\mu\Delta_K + \Delta_K\delta_K)\dot{x}_{K2}(0)]\cosh(T) + [(\alpha_K + \mu + 2\delta_K)\dot{x}_{K2}(T) + (\mu\Delta_K + \delta_K\Delta_K)\dot{x}_{K2}(T)]\rangle \qquad (2.19)$$

But $q(t_K) = \dot{f}(t_k) - \Omega_K(t_k)$,

$$\int_0^T f(\dot{s}_K)\sinh(t_K - s_k)ds_k = -\sinh(T)f(0) + \int_0^T f(\dot{s}_K)\cosh(t_K - s_k)ds_k \qquad (2.20)$$

$$\int_0^T q(s_k)\sinh(T - s_k)ds_k = -\sinh T\{\mu\Delta_k{}^2\dot{x}_{K2}(0) + \Delta_K(\mu + \delta_K)x_{K2}(0)\} + \int_0^T\{\mu\Delta_K{}^2\dot{x}_{K2}(t_k) + \Delta_K(\mu + \delta_K)x_{K2K2}(s_K)\}\cosh(T - s_K)ds_K - \int_0^T\{(\alpha_K + \mu + 2\delta_K)x_{K2}(s_k) + \Delta_K(\mu + \delta_K)\dot{x}_{K2}(s_k)\}\sinh(T - s_k)ds_k \qquad (2.21)$$

$$\begin{aligned}\tau_0 &= \frac{1}{sinh\,T}\{[(\alpha_K+\mu+2\delta_K)x_{K2}(T)+\Delta_K(\mu+\delta_K)\dot{x}_{K2}(T)]-[(\alpha_K+\mu+2\delta_K)x_{K2}(0)\\ &+\Delta_K(\mu+\delta_K)\dot{x}_{K2}(0)cshT\}-\frac{1}{sinh\,T}\{-sinh\,T[\left(\mu\Delta_K{}^2\,\dot{x}_{K2}(0)\right)+\Delta_K(\mu+\delta_K)x_{K2}(0)]\\ &+\int_0^T\{\left(\mu\Delta_K{}^2\,\dot{x}_{K2}(s_k)\right)+\Delta_K(\mu+\delta_K)x_{K2}(s_k)\}cosh(T-s_k)ds_k--\int_0^T\{(\alpha_K+\mu+2\delta_K)x_{K2}(s_k)\\ &\qquad+\Delta_K(\mu+\delta_K)\dot{x}_{K2}(s_k)\}sinh(T-s_k)ds_k\}\end{aligned} \tag{2.22}$$

$$\begin{aligned}V_{11}(t_k)&=A_{11}(t_k)=\tau_0\,sinh(t_k)+[(\alpha_k+\mu+2\delta_{kk})x_{K2}(0)+(\mu+\delta_k)\Delta_K\,\dot{x}_{K2}(0)]\,cosh\,t_k-sinh\,T\{\mu\Delta_k{}^{\Delta 2}\dot{x}_{K2}(0)+\\ &+\Delta_K(\mu+\delta_K)x_{K2}(0)\}+\int_0^T\{\left(\mu\Delta_K{}^2\,\dot{x}_{K2}(s_k)\right)+\Delta_K(\mu+\delta_K)x_{K2}(s_k)\}cosh(t_k-s_k)ds_k\\ &-\int_0^T\{(\alpha_K+\mu+2\delta_K)x_{K2}(s_k)+\Delta_K(\mu+\delta_K)\dot{x}_{K2}(s_k)\}sinh(t_{kk}-s_k)ds_k\end{aligned} \tag{2.23}$$

In equation (2.11), setting $x_{K2}(t_k)=0 \rightarrow \dot{x}_{K2}(t_k)=0$. We have

$$\begin{aligned}\langle Z_{K1},AZ_{K2}(t_k)\rangle_H &= \sum_{k=0}^{n}\{\beta_k u_{K1}(t_k)u_{K2}(t_k)+\lambda_k x_{K1}(t_k)u_{K2}(t_k)+\rho_k\Delta_K\,\dot{x}_{K1}\,u_{K2(t_k)}\\ &+\rho_K x_{K1}(t_k)u_{K2}(t_k)\}=\sum_{k=0}^{n}\{x_{K1}(t_k)[\lambda_k u_{K2}(t_k)+\rho_k u_{K2}(t_k)]+\dot{x}_{K1}[\rho_k\Delta_k u_{K2}(t_k)]\\ &+u_{K1}(t_k)\beta_K u_{K2}(t_k)\}=\sum_{k=0}^{n}\{x_{K1}(t_k)V_{12}(t_k)+\dot{x}_{K1}\,\dot{V}_{12}(t_k)+u_{K1}(t_k)V_{22}(t_k)\}\end{aligned} \tag{2.24}$$

$$V_{22}(t_k)=A_{22}u_{K2}(t_k)=\beta_K u_{K2}(t_k)$$

Again define $g(t_k)=(\lambda_k+\rho_k)u_{K2}(t_k)$ and $h(t_k)=\rho_k\Delta_k u_{K2}(t_k)$, $g(t_k)-V_{12}(t_k)$ *and* $h(t_k)-\dot{V}_{12}(t_k)$ are continuous function on $[0,T]$. As before

$$V_{12}(t_k)=\int^T q_1(t_k)\,sinh(t_k-s_k)ds_k+e_0 csht_k+l_0\,sinh\,t_k \tag{2.25}$$

$$e_0=g(0)=(\lambda_k+\rho_k)u_{K2}(0)$$

$$\begin{aligned}l_0&=\frac{[\,g(T)-\int_0^T q_1(s_k)\,sinh(T-s_k)ds_k-g(0)cosh\,T\,]}{sin\,h\,T}\\ &=\frac{[(\lambda_K+\rho_K)u_{K2}(T)-\int_0^T q_1(s_k)\,sinh(T-s_k)ds_k-(\lambda_k+\rho_K)u_{K2}(0)cshT\,]}{sin\,h\,T}\end{aligned}$$

$$\begin{aligned}V_{12}(t_k)&=(\rho_k\Delta_k)u_{K2}(0)\,sinh(t_k)-\int_0^{t_k}(\rho_k\Delta_k)u_{K2}\,cosh(t_k-s_k)ds_k\\ &-\int_0^{t_k}(\lambda_k+\rho_k)u_{K2}(s_K)\,sinh(t_k-s_k)ds_k+(\lambda_k+\rho_k)u_{K2}(0)cosh\,t_k+\frac{sinh\,t}{sinh\,T}\{(\lambda_k+\rho_k)u_{K2}(T)\\ &-(\lambda_k+\rho_k)u_{K2}(0)cosh\,T-(\rho_k\Delta_k)u_{K2}(0)\,sinh(T)+\int_0^{T_k}(\rho_K\Delta_K)u_{K2}(s_K)cosh(T-s_k)ds_k\\ &+\int_0^T(\lambda_K+\rho_K)u_{K2}\,sinh(T-s_k)ds_k\,\}\end{aligned} \tag{2.26}$$

2.4 Data and analysis

Having constructed operator A, written as $A = \begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix}$. where V_{11} is (2.16), V_{12} is (2.26), V_{21} is (2.12b), V_{22} is (2.24b). The discretized generalized problem (P1) is now applied to the following problems P1 and P2 stated thus,

Problem P1

Min $\int_0^1 \left(x^2(t)+u^2(t)\right)dt$ such that $\dot{x} = 2.095x(t)+1.904u(t)$, the solution to this problem is obtained by assuming the following initial values; $x_0 = t, u_0 = 0.5$ and $0.5 \le \mu \le 2.5$. The exact analytical solution is 1.0647 given by [7]. Assume the followings arising from the discretization with meshpoints in the interval [0,1];

$$\alpha_k = a\Delta_k + \mu + \mu\Delta_k{}^2c^2 + \mu 2c\Delta_k . \beta_k = b\Delta_k + \mu d^2\Delta_k{}^2,\ \lambda_k = 2\mu d\Delta_k + 2\mu cd\Delta_k{}^2,\ \delta_k = -2\mu(1+c\Delta_k),\quad \rho_k = -2\mu d\Delta_k$$

where Δ_k is the step size, μ the penalty constant, $a = 1, b = 1, c = 2.095$ and $d = 1.904$

The problem has been solved analytically and by other numerical methods such as function space algorithm (FSA), Extended conjugate gradient method (ECGM) and multiplier imbedding extended conjugate gradient method (MECGM)[8]. The concern here, in this paper, is solving the discretized constrained algorithm (DCA) problem numerically using penalty constant μ, where μ=. 5(2.5). 5, i.e. μ assumes initial value .5 with increment = .5 and terminal value 2.5. These penalty constants are chosen small, since bigger penalty constants, say 10, 20, 40, 50, 60, 80 and 100 tend to violate constraints satisfaction [8]. The step size=.2 is chosen arbitrarily constant. Also, the number of iterations is determined by the value of the gradient within some specified interval, say [.0025, +.0025], in the conjugate gradient algorithm , otherwise allowing the gradient , g_0 =0 may result in an infinite loop. A program written in q-basic gave the following tabulated results:

Table 1.1: Numerical Solution of Problem P1

Penalty Constants	Alg	Stepsize	Iteration	Objective Function	Constraint Satisfaction	Penalized Functional
μ = .5	DCA	.2	5	1.9859	2.6864	3.32917
μ = .5 λ =-2.88	FSA ECGM MECGM	.2 .2 .2	50 7 10	1.6517 1.0956 1.0715	11.6227 0.4544 1.1249	
μ = .1	DCA	.2	5	1.982307	8.557396	10.4697
μ = .1 λ = -6.00	FSA ECGM MECGM	.2 .2 .2	50 7 4	1.6250 1.4834 0.7073	11.2990 0.13813 0.95018	
μ = 1.5	DCA	.2	5	1.352	9.61526	15.76779
μ = 1.5 λ = -9.11	FSA ECGM MECGM	.2 .2 .2	50 6 3	1.60017 1.5557 0.8686	10.9884 0.08652 1.1616	
μ = 2	DCA	.2	5	1.352	9.61526	20.51305
μ = 2 λ =-10.57	ESA ECGM MECGM	.2 .2 .2	50 7 3	1.57684 1.4686 0.9386	10.6902 0.03531 1.0477	
μ =2.5	DCA	.2	5	1.352	9.61526	25.37831
μ =2.5 λ =-10.37	FSA ECGM	.2 .2	50 6	1.55497 1.58206	10.402 $8.1262*10^{-3}$	

	MECGM	.2	2	1.0178	1.6313	

Problem P 2

$\text{Min}\int_0^1 \left(x^2(t)+u^2(t)\right)dt$ subject to $\dot{x}(t) = \underset{0\le t\le T}{U},\ x(0)=1.$ The solution to this problem is obtained by assuming the following initial values for the variables; $x_0 = 1,\ u_0 = 1$. The exact analytical solution is 0.7641. Applying the same algorithm to Problem P2 and solving by gbasic programming language, we have the following Table 1.2:

Table 1.2: numerical solution of P2

Penalty Constant	Algo	Stepsize	Iteration	Objective	Constrained Satisfaction
μ = .5	DCA	.2	1	1.6	.8
μ = .5 λ = -0.74	FSA ECGM MECGM	.2 .2 .2	50 3 3	1.9777 0.79989 0.1781	0.9789 0.01313 0.9169
μ = 1	DCA	.2	3	0.8303	0.06353
μ = 1.0 λ = -1.55	FSA ECGM MECGM	.2 .2 .2	50 4 2	1.9742 0.72768 0.6051	0.9648 0.0206 0.4063
μ = 1.5	DCA	.2	7	0.8676	0.065391
μ = 1.5 λ = -3.49	FSA ECGM MECGM	50 4 2	1.971011 0.97256 0.7647	0.9514 0.2232 0.2875	
μ = 2.0	DCA	.2	11	0.8769	0.04895
μ = 2.0 λ = -4.86	FSA ECGM MECGM	.2	50 7 3	1.9677 0.98866 0.7013	0.9379 0.01256 0.1942
μ=2.5	DCA	.2	15	0.8747	0.038922
μ = 2.5 λ = -8.49	FSA ECGM MECGM	.2	50 6 4	1.9645 0.92047 0.6894	0.9247 0.02595 0.13710

3.0 Summary and recommendation

From the above Table 1.1, we see that for parameter constant.5 ≤ μ ≤ 1.5 the result of DCA trails behind other methods with step length .2. But for parameter μ greater than 1.0 the result is better than either the FSA or the ECGM but trails behind the MECGM. And for parameter greater than 1.5 the results for the objectives and the constrained are repeated.

The penalized functional values in the penalized functional column are also included in this table. These values reflect what are expected, since the penalty constants are also increasing. For Table 1.2 with time step (.2), the FSA trails behind every other algorithm in term of convergence. In fact, FSA maintains a largest constant number of iterations per circle for every μ, since its decreasing sequence of solutions; 1.9777, 1.9742, 1.971011, 1.9677, 1.9645,is obviously diverging Therefore attention for comparison is focussed between either the MECGM or ECGM and the DCA.

On one hand, the DCA with an optimum at .8303 trails behind the MECGM wiith an optimum at .7647 yet its trend in terms of iteration is increasing as its objective functional values appreciate to the analytic Optimum .7641, for 1.0 ≤ μ ≤ 2.5 while the MECGM's iteration, though lowest, can be likened to a discrete sinusoidal graph or valley for all μ such that .5 ≤ μ ≤ 1.0.

On the other hand, the ECGM comes closest to the optimum .7276 for $.5 \leq \mu \leq 1.0$ but suddenly deteriorates to trail behind the DCA for $1.5 \leq \mu \leq 2.5$ with unstable pattern number of iterations.

Conclusively, DCA performs better than either FSA or ECGM and trails behind MECGM but iteratively predictable than MECGM.

References

[1] Fletcher,R. and Reeves,C.M(1964),Function minimization by conjugate gradients, Computer Journal electronic computing laboratory. The university of Leeds. Volume 7, pg. 149-154.

[2] Gianni, D. and Laura,Palogi(1998) ,Non-linear programming; Introdution to unconstrained and constrained optimization;Technical report, New York, U.S.A

[3] Gianni, D. and L.Grippo(1972), A computing algorithm for the epsilon method to identification and optimal control problems, journal of Ricerche Di Automata,Italy, Volume 1, pg. 54-77.

[4] Ibiejugba, M.A.and Inumanyi,P(1984), A control operator and some of its applications , Journal of mathematical analysis and applications, New York,. Volume 103, pg. 31-47.

[5] Loostman,A. (1972), A derivation of conjugate gradients numerical methods for non-linear optimization , London Academic press Company, London.

[6] Morton,K.W. and Mayers, D.F,(1996), Numerical solution of Partial Differential Equations 40 West 20th Street, New York, NY 100011-4211, U..S.A.

[7] Olorunsola, S.A.(1991) On a multiplier imbedding extended conjugate gradient method , PHD. Thesis.Mathematics Department, University of Ilorin, Ilorin , Kwara State, Nigeria

[8] Olorunsola,S.A. (1995), On two updating methods of the multipliers in the multipliers imbedding algorithm, Journal of Systems analysis modeling simulation, Netherland, volume 25, pg. 51-73.