Journal of Mathematics and Statistics 3 (4): 237-242, 2007
ISSN 1549-3644


# On the Discretized Algorithm for Optimal Problems Constrained by Differential Equation with Real Coefficients

O. Olotu
Department of Mathematical Sciences, The Federal University of Technology,
P.M.B. 704, Akure, Ondo State, Nigeria

**Abstract:** A discretized scheme, Discretized Continuous Algorithm (DCA), for solving constrained quadratic optimal control problems was developed to ease the computational cumbersomeness inherent in some existing algorithms, particularly, the Function Space A lgorithm (FSA) by replacing the integral by a series of summation. In order to accomplish this numerical scheme, we resort to a finite approximation of it by discretizing its time interval and using finite difference method for its differential constraint. Using the penalty function method, an unconstrained formulation of the problem was obtained. With the bilinear form expression of the problem, an associated operator was constructed which aided the scheme for the solution of such class of problems. A sample problem was examined to test the effectiveness of the scheme as to convergence with relation to other existing schemes such as Extended Conjugate Gradient Method (ECGM), Multiplier Imbedding Extended Conjugate Gradient Method (MECGM) and Function Space Algorithm (FSA) for solving penalized functional of optimal control problem characterized by non-linear integral quadratic nature.

**Key words:** Discretized, quadratic, differential constraint, conjugate gradient and operator

## INTRODUCTION

The discretized scheme, DCA, with less computational rigour was proposed and compared to some existing algorithms, particularly the FSA which circumvented the use of operator, for solving a class of quadratic optimal control problems. ECGM and MECGM based on[1] on function minimization reviewed by[4] were ingredients to the development of the discretized scheme. Here, a generalized constrained formulation of the problem is given below for the discretization exercise of the scheme.

## MATERIALS AND METHODS

Generalized problem (P1)

$$\text{Min} \int_0^T (ax^2(t) + bu^2(t))dt$$

Subject to

$\dot{x}(t) = cx(t) + dU(t) \qquad 0 \le t \le T$

$X(0)=X_0=0$ a ,b,c , d are in R (1)

The constrained problem can be turned into unconstrained problem via the penalty method[2].

The problem may be put in the following equivalent form;

$$\langle Z, AZ\rangle_H = \text{Min}_{(x,U)} \int_0^T \{ax^2(t) + bu^2(t) + \mu\left\|\dot{x}(t) - cx(t) - dux(t)\right\|^2\}dt \tag{2}$$

$\mu \ge 0$ is the penalty constant.

**Discretization:** By[3], discretizing (2), subdivide [0,T] into n equal intervals at meshpoints

$$x_0 < x_1 < x_2 < x_3 < \ldots < x_{n-1} < x_n$$

with

$$nx_j = j\Delta_j,\ j = 0,1,2,\ldots,n,$$

where $\Delta_j$ is the fixed length of each subinterval.

$$\text{Let} \quad t_0 = 0 \ \text{and} \quad t_k = \sum_{j=1}^{k} \Delta j, \quad t_{n-1} = T, \quad k = 1,2,3,\cdots,n$$

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

By finite difference method ,

$$\dot{X}(k) = (X(k+1) - X(k))/\Delta_k,\ k = 0,1,2,\ldots N\text{-}1$$

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

$$\left(x(k+1) - x(k)\right)/\Delta_k = cx_k(t_k) + du_k(t_k) \qquad (3)$$

X(0) = 0

We then have the discretized function in the form;

$$\min J = \sum_{k=0}^{n} \Delta_k \left(ax_k^2(t_k) + bu_k^2(t_k)\right)$$
$$\text{subject to} \quad \left(x(k+1) - x(k)\right)/\Delta_k = cx_k(t_k) + du_k(t_k) \qquad (4)$$
$$x(0) = 0$$

**Equivalent unconstrained and discretized formulation of the problem:** (4) is turned into an unconstrained problem

$$\mathrm{MinJ}(x,u,\mu) = \sum_{k=0}^{n} \left\{ \begin{array}{l} \Delta_k\left(ax_k^2(t_k) + bu_k^2(t_k)\right) + \mu[x_{K+1} \\ (k+1) - x_k(t_k) - \Delta_k cx_k(t_k) \\ -d\Delta_k u_k(t_k)]^2 \end{array} \right\}.$$

$$= \sum_{k=0}^{n} \left\{ \begin{array}{l} \Delta_k\left(ax_k^2(t_k) + bu_k^2(t_k)\right) + \mu[x_{k+1}^2(t_k) \\ +x_k^2(t_k) + \Delta_k^2 c^2 x_k^2(t_k) + d^2\Delta_k^2 u_k^2(t_k) \\ +2c\Delta_k x_k^2(t_k) + 2d\Delta_k x_k(t_k) u_k(t_k) \\ +2cd\Delta_k^2 x_k(t_k) u_k(t\ ) - 2x_{k+1}(t_k) x_k(t_k) \\ -2c\Delta_k x_{k+1} x_k(t_k) - 2d\Delta_k x_{k+1}(t_k) u_k(t_k)] \end{array} \right\} \qquad (5)$$

Simplifying (5), we have

$$\sum_{k=0}^{nn} \left\{ \begin{array}{l} x_k^2(t_k)[a\Delta_k + \mu + \mu\Delta_k^2 c^2 + \mu 2c\Delta_k] + u_k^2(t_k) \\ [b\Delta_k + \mu d^2 \Delta_k^2] + \mu x_{k+1}^2(t_k) + x_k(t_k) u_k(t_k) \\ [2d\Delta_k \mu + 2cd\Delta_k^2 \mu] + x_{k+1}(t_k) x_k(t_k)[-2\mu - 2\mu c\Delta_k] \\ +x_{k+1}(t_k) u_k(t_k)[-2\mu d\Delta_{k]}]\vdots \end{array} \right\} \qquad (6a)$$

$$\text{Let} \quad Z_k = \begin{pmatrix} x_k(t_k) \\ u_k(t_k) \end{pmatrix} \text{and} \quad y_k(t_k) = x_{k+1}(t_k)$$

$$\begin{array}{l} \text{Let} \quad \alpha_k = a\Delta_k + \mu + \mu\Delta_k^2 c^2 + \mu 2c\Delta_k. \\ \beta_k = b\Delta_k + \mu d^2 \Delta_k^2 \\ \lambda_k = 2\mu d\Delta_k + 2\mu cd\Delta_k^2 \\ \delta_k = -2\mu(1 + c\Delta_k), \cdots \rho_k = -2\mu d\Delta_k \end{array} \qquad (6b)$$

(6a) becomes

$$\sum_{k=0}^{n} \left\{ \begin{array}{l} \alpha_k x_k^2(t_k) + \beta_k u_k^2(t_k) + y_k^2(t_k)\mu + x_k(t_k) u_k(t_k) \\ \lambda_k + y_k(t_k) x_k(t_k) \delta_k + y_k(t_k) u_k(t_k) \rho_k \end{array} \right\} \qquad (7)$$

**Construction of operator a**:
Now,

$$\begin{array}{l} \langle Z_{K1}(t_k), AZ_{K2}(t_k)\rangle_H = \sum_{k=0}^{n} \{\alpha_k x_{K1}(t_k) x_{K2}(t_k) + \beta_k u_{K1} \\ (t_k) u_{K2}(t_k) + y_{k1}(t_k) y_{k2}(t_k)\mu \\ +\lambda_K x_{K1}(t_k) u_{K2}(t_k) + \lambda_K u_{K1}(t_k) \\ x_{K2}(t_k) + \delta_K y_{k1}(t_k) x_{k2}(t_k) \\ +\partial_k y_{k2}(t_k) x_{k1}(t_k) + \rho_k y_{k1}(t_k) \\ u_{k2}(t_k) + \rho_k u_{k1}(t_k) y_{k2}(t_k)\} \end{array} \qquad (8)$$

$$AZ_{K2}(t_k) = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} x_{k2} \\ u_{k2} \end{pmatrix} = \begin{pmatrix} A_{11}x_{k2} + A_{12}u_{k2} \\ A_{21}x_{k2} + A_{22}u_{K2} \end{pmatrix} \text{by}^{[5]} \qquad (9)$$

Further simplifying (8) and using[6,7], we have

$$\begin{array}{l} \langle Z_{K1}(t_k), AZ_{K2}(t_k)\rangle_H = \sum_{k=0}^{n} \{\alpha_k x_{K1}(t_k) x_{K2}(t_k) + \beta_k u_{K1} \\ (t_k) u_{K2}(t_k) + \mu[(\Delta_K \dot{x}_{K1} + x_{K1}) \\ (\Delta_K \dot{x}_{K21} + x_{K2})] + \lambda_K x_{K1} u_{K2} + \\ \lambda_K u_{K1} x_{K2} + \delta_K(\Delta_K \dot{x}_{K1} + x_{K1}) \\ x_{K2} + \delta_K x_{K1}(\Delta_K \dot{x}_{K2} + x_{K2}) \\ +\rho_K(\Delta_K \dot{x}_{K1} + x_{K1}) u_{K2} + \rho_K u_{K1} \\ (\Delta_K \dot{x}_{K2} + x_{K2})\} \end{array} \qquad (10)$$

$$\begin{array}{l} = \sum_{k=0}^{n} \{\alpha_k x_{K1}(t_k) x_{K2}(t_k) + \beta_k u_{K1}(t_k) u_{K2}(t_k) + \mu\Delta_K^2 \dot{x}_{K1} \\ (t_k) \dot{x}_{K2}(t_k) + \mu\Delta_K \dot{x}_{K1}(t_k) x_{K2}(t_k) + \mu\Delta_K x_{K1}(t_k) \\ \dot{x}_{K2}(t_k) \dot{x}_{k2}(t_k) + \mu x_{K1}(t_k) x_{K2}(t_k) + \lambda_K x_{K1}(t_k) \\ u_{K2}(t_k) + \lambda_K u_{K1}(t_k) x_{K2}(t_k) + \delta_K \Delta_K \dot{x}_{K1}(t_k) x_{K2} \\ (t_k) + \delta_K x_{K1}(t_k) x_{K2}(t_k) + \delta_K \Delta_K x_{K1}(t_k) \dot{x}_{K2}(t_k) \\ +\delta_K x_{K1}(t_k) x_{K2}(t_k) + \rho_K \Delta_K u_{K2}(t_k) \dot{x}_{K1}(t_k) \\ +\rho_K u_{K2}(t_k) x_{K1}(t_k)\} \end{array} \qquad (11)$$

Setting $\mu_{K2}(T_k) = 0$, in (11) and by[5], we have

$$\begin{pmatrix} A_{11}x_{K2} \\ A_{21}x_{K2} \end{pmatrix} = \begin{pmatrix} V_{11} \\ V_{21} \end{pmatrix} \qquad (12)$$

and

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

$$= \sum_{k=0}^{n}\{x_{K1}(t_k)[\alpha_k x_{k2}(t_k) + \mu\Delta_K\dot{x}_{K2}(t_k) + \mu x_{K2}(t_K) + \delta_K x_{K2}(t_k) + \delta_K\Delta_K\dot{x}_{K2}(t_k) + \delta_K x_{K2}(t_k)] + \dot{x}_{K1}(t_k) [\mu\Delta_k{}^2\dot{x}_{K2}(t_k) + \mu\Delta_K x_{K2}(t_k) + \delta_K\Delta_K x_{K2}(t_k)] + u_{K1}(t_k)[\lambda_K x_{k2}(t_k)]\} \quad (14)$$

$$= \sum_{k=0}^{n}\{x_{K1}(t_k)V_{11}(t_k) + \dot{x}_{K1}(t_k)\dot{V}_{11}(t_k) + u_{K1}(t_k)V_{21}\} \quad (15)$$

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

$$f(t_k) = \mu\Delta_K{}^2\dot{x}_{K2}(t_k) + (\mu\Delta_K + \delta_K)x_{K2}(t_k)$$

$$A_{21}u_{K1}(t_k) = V_{21}(t_k) = \lambda_K x_{K2}(t_k). \quad (15a)$$

To obtain the component $A_{11}x_{K1}(t_k)=V_{11}(tk)$, where

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

are both continuous functions on [0,T] and choosing

$x_{K1}(\bullet)\in D[0,T] \ni x_{K1}(0) = x_{K1}(T) = 0.$

We then have

$$\int_0^T\{x_{K1}(t_k)[\Omega(t_k) - V_{11}(t_k)] + \dot{x}_{K1}(t_k)[f(t_k) - \dot{V}_{11}(t_k)]\} dt_k = 0, \quad (16)$$

by[5]

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

$$\frac{d}{dt}[f(t_k) - \dot{V}_{11}(t_k)] = \Omega(t_k) - V_{11}(t_k) \quad (17)$$

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

$$\ddot{V}_{11}(t_k) - V_{11}(t_k) = f^\bullet(t_k) - \Omega(t_k)$$

$$\ddot{V}_{11}(t_k) - V_{11}(t_k) = q(t_k) = \dot{f}(t_k) - \Omega(t_k) \quad (18)$$

with the initial conditions $V_{11}(0) = p_0$ and $V_{11}(0) = r_0$ Solving (18) by Laplace method and letting

$$L\{V_{11}(t_k)\} = \hat{V}_{11}(s),\cdots L\{q(t_k)\} = Q(s)$$

We have

$$S^2\hat{V}(s)\text{-}p_0 s\text{-}r_0\text{-}\hat{V}_{11}(s)=Q(s)$$

$$\hat{V}_{11}(s)=\frac{Q(s)}{s^2\text{-}1}+\frac{p_0 s}{s^2}+\frac{r_0}{s^2\text{-}1}$$

and taking Laplace inverse, we have But,

$$V_{11}(t_k) = \int_0^T q(s_k)\sinh(t_k - s_k)ds_k + p_0\cosh(t_k) + r_0\sinh(t_k) \quad (19)$$

$$\Omega(T) - V_{11}(T) = 0 \quad (20)$$

$$\Omega(0) - V_{11}(0) = 0 \quad (21)$$

$$\Omega(0) = \rho_0$$

$$\Omega(0)=(\alpha_k+\mu+2\delta_K)x_{K2}(0)+(\mu\Delta_K+\delta_K\Delta_K)\dot{x}_{K2}(0)=\rho_0$$

From (20) $\Omega(T) = V_{11}(T)\cdots$

$$V_{11}(t_k) = \int_0^T q(s_k)\sinh(T - s_k)ds_k + [(\alpha_k + \mu + 2\delta_k)x_{K2}(0) + (\mu\Delta_K + \Delta_K\delta_K)\dot{x}_{K2}(0)]\cosh(T) + \tau_0\sinh(T)$$
$$= [(\alpha_K + \mu + 2\delta_K)\dot{x}_{K2}(T) + (\mu\Delta_K + \delta_K\Delta_K)\dot{x}_{K2}(T)]$$

Therefore,

$$\tau_0 = \frac{1}{\sinh(T)}\langle -\int_0^T q(s_k)\sinh(T - s_k)ds_k - [(\alpha_k + \mu + 2\delta_k) x_{K2}(0) + (\mu\Delta_K + \Delta_K\delta_K)\dot{x}_{K2}(0)]\cosh(T) + [(\alpha_K + \mu + 2\delta_K)\dot{x}_{K2}(T) + (\mu\Delta_K + \delta_K\Delta_K)\dot{x}_{K2}(T)]\rangle \quad (22)$$

But,

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

$$\int_0^T f(\dot{s}_K)\sinh(t_K-s_k)ds_k=-\sinh(T) \quad f(0)+\int_0^T f(\dot{s}_K)\cosh(t_K-s_k)ds_k \tag{23}$$

$$\begin{aligned}&\int_0^T q(s_k)\sinh(T-s_k)ds_k=- \\ &\sinh T\{\mu\Delta_K^{2}\dot{x}_{K2}(0)+\Delta_K(\mu+\delta_K)x_{K2}(0)\} \\ &+\int_0^T\{\mu\Delta_K^2\dot{x}_{K2}(t_k)+\Delta_K(\mu+\delta_K)x_{K2_{K2}}(s_K)\} \\ &\cosh(T-s_K)ds_K-\int_{00}^T\{(\alpha_K+\mu+2\delta_K)x_{K2}(s_k) \\ &+\Delta_K(\mu+\delta_K)\dot{x}_{K2}(s_k)\}\sinh(T-s_k)ds_k\end{aligned} \tag{24}$$

$$\begin{aligned}\tau_0=&\frac{1}{\sinh T}\{[(\alpha_K+\mu+2\delta_K)x_{K2}(T)+\Delta_K(\mu+\delta_K) \\ &\dot{x}_{K2}(T)]-[(\alpha_K+\mu+2\delta_K)x_{K2}(0)+\Delta_K(\mu+\delta_K) \\ &\dot{x}_{K2}(0)]\cosh T\}-\frac{1}{\sinh T}\{-\sinh T[\left(\mu\Delta_K^2\dot{x}_{K2}(0)\right) \\ &+\Delta_K(\mu+\delta_K)x_{K2}(0)]+\int_0^T\{\left(\mu\Delta_K^2\dot{x}_{K2}(s_k)\right)+\Delta_K \\ &(\mu+\delta_K)x_{K2}(s_k)\}\cosh(T-s_k)ds_k-\int_0^T\{(\alpha_K+\mu+2\delta_K) \\ &x_{K2}(s_k)+\Delta_K(\mu+\delta_K)\dot{x}_{K2}(s_k)\}\sinh(T-s_k)ds_k\}\end{aligned} \tag{25}$$

$$\begin{aligned}V_{11}(t_k)=&A_{11}(t_k)=\tau_0\sinh(t_k)+[(\alpha_k+\mu+2\delta_{kk})x_{K2}(0) \\ &+(\mu+\delta_k)\Delta_K\dot{x}_{K2}(0)]\cosh t_k-\sinh T\{\mu\Delta_k^2\dot{k}_{k2}(0) \\ &+\Delta_K(\mu+\delta_K)x_{K2}(0)\}+\int_0^T\{\left(\mu\Delta_K^2\dot{x}_{K2}(s_k)\right) \\ &+\Delta_K(\mu+\delta_K)x_{K2}(s_k)\}\cosh(t_k-s_k)ds_k- \\ &-\int_0^T\{(\alpha_K+\mu+2\delta_K)x_{K2}(s_k)+\Delta_K(\mu+\delta_K) \\ &\dot{x}_{K2}(s_k)\}\sinh(t_{kk}-s_k)ds_k\end{aligned} \tag{26}$$

In equation (11), setting

$$x_{K2}(t_k)=0 ® \dot{x}_{K2}(t_k)=0$$

We have

$$\begin{aligned}\langle Z_{K1},AZ_{K2}(t_k)\rangle_H=&\sum_{k=0}^{n}\{\beta_k u_{K1}(t_k)u_{K2}(t_k)+\lambda_k x_{K1}(t_k) \\ &u_{K2}(t_k)+\rho_k\Delta_K\dot{x}_{K1}u_{K2(t_k)}+\rho_K x_{K1}(t_k)u_{K2}(t_k)\} \\ =&\sum_{k=0}^{n}\{x_{K1}(t_k)[\lambda_k u_{K2}(t_k)+\rho_k u_{K2}(t_k)]+\dot{x}_{K1} \\ &[\rho_k\Delta_k u_{K2}(t_k)]+u_{K1}(t_k)\beta_K u_{K2}(t_k)\} \\ =&\sum_{k=0}^{n}\{x_{K1}(t_k)V_{12}(t_k)+\dot{x}_{K1}\dot{V}_{12}(t_k)+u_{K1} \\ &(t_k)V_{22}(t_k)\}\end{aligned} \tag{27}$$

$$V_{22}(t_k)=A_{22}u_{K2}(t_k)=\beta_K u_{K2}(t_k) \tag{27a}$$

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 functions on $[0,T]$

As before

$$V_{12}(t_k)=\int^T q_1(t_k)\sinh(t_k-s_k)ds_k+e_0\cosh t_k+l_0\sinh t_k \tag{28}$$

Where,

$$\begin{aligned}e_0&=g(0)=(\lambda_k+\rho_k)u_{K2}(0) \\ l_0&=[g(T)-\int_0^T q_1(s_k)\sinh(T-s_k)ds_k-g(0) \\ &\cosh T]/\sinh T \\ &=[(\lambda_K+\rho_K)u_{K2}(T)-\int_0^T q_1(s_k)\sinh(T-s_k) \\ &ds_k-(\lambda_k+\rho_K)u_{K2}(0)\mathrm{csh}T]/\sinh T\end{aligned} \tag{29}$$

$$\begin{aligned}V_{12}(t_k)=&(\rho_k\Delta_k)u_{K2}(0)\sinh(t_k)-\int_0^{t_k}(\rho_k\Delta_k)u_{K2}\cosh \\ &(t_k-s_k)ds_k-\int_0^{t_k}(\lambda_k+\rho_k)u_{K2}(s_K)\sinh(t_k-s_k)ds_k+ \\ &(\lambda_k+\rho_k)u_{K2}(0)\cosh t_k+\frac{\sinh t}{\sinh T}\{(\lambda_k+\rho_k)u_{K2}(T)- \\ &(\lambda_k+\rho_k)u_{K2}(0)\cosh T-(\rho_k\Delta_k)u_{K2}(0)\sinh(T) \\ &+\int_0^{T_k}(\rho_K\Delta_K)u_{K2}(s_K)\cosh(T-s_k)ds_k \\ &+\int_0^T(\lambda_K+\rho_K)u_{K2}\sinh(T-s_k)ds_k\}\end{aligned} \tag{30}$$

Having constructed operator A, written as

$$A=\begin{pmatrix} v_{11} & v_{12} \\ v_{21} & v_{22} \end{pmatrix}$$

Where,

$$v_{11} \text{ is } (26), v_{12} \text{ is } (30),$$
$$v_{21} \text{ is } (15a), \quad v_{22} \text{ is } (27a)$$

the discretized algorithm is now applied to the following hypothetical problems P1 and P2 stated thus;

## EXAMPLES

Example problem P1

$$\text{Min}\int_0^1 \left(x^2(t)+u^2(t)\right)dt$$

such that

$$\dot{x}(t)=2.095x(t)+1.904u(t), \quad 0£t£1$$

The exact analytical solution is 1.0647 given by[7]. The numerical solution to this problem is obtained by assuming the following initial values and parameters; x0 = 1, u0 =.5 and.5 $\le \mu \le$ 2.5

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

Where,

$\Delta_k$ is the stepsize
$\mu$ the penalty constant
$a = 1, b = 1, c = 2.095$ and $d = 1.904$

The problem has been solved by other numerical methods such as Function Space Algorithm(FSA), Extended Conjugate Gradient Method (ECGM) and Multiplier Imbedding Extended Conjugate Gradient Method(MECGM)[ ] with results tabulated below. The concern here, in this paper, is solving the discretized problem numerically using penalty constant $\mu$, where $\mu$ =.5(2.5).5. The stepsize = .2 is chosen arbitrarily constant. Also, the efficiency of each sheme is determined by the value of the objective function value closiest to the exact solution with a level of tolerance $\mp$.3500 per iteration

**Example problem P2**

$$Min\int_0^1 \left(x^2(t)+u^2(t)\right)dt$$

subject to

$$\dot{x}(t) = u(t), \quad x(0) = 0 \quad 0 \le t \le 1$$

The solution to this problem is obtained by assuming the following initial values for the variables;

$$x_0 = 1, \quad u_0 = 1$$

The exact analytical solution is 0.7641 Applying the same algorithm to problem P2 and solving by qbasic programming language, we have the following table 1.2

## COMMENTS AND CONCLUSION

In Table 1, for parameters, $0.5 \le \mu \le 2.5$, we see that the absolute value difference between the exact solution 1.0647 and the numerical solution per iteration in column 5 for DCA and MECGM all lie within the interval $|x_0 - x|$ £ .3500, except for $\mu$ = 2.5 and $\mu$ = 0.5, respectively,

Where,
$x_0 = 1.0647$, x is the iterate
and $\pm$.3500 is the tolerance level.

All other algorithms ,namely FSA and ECGM have performed less effectively. Although, ECGM for $\mu$ = 0.5 has a numerical solution 1.0956 with a deviation of only 0.0309. This is the only point where its solution compares much more favourably to any other algorithm. For other values, ECGM becomes irrelevant as to convergence profile.

FSA, for all parameters, is not comparable. Hence it performs worst of all the algorithms. Conclusively, DCA and MECGM are ranked qualitatively equivalent but comparably better than either FSA or ECGM.

In Table 2, step 2, the superiority of the DCA and ECGM has been exhibited for all values of the parameters in $1.0 \le \mu \le 2.5$ in column 5, except for $\mu$ = 0.5, where it ranks second to ECGM with minimum

Table 1: Numerical solution of problem p1 compared to the exact solution (1.0647)

| Penalty constants | Algor | Stepsize | Iteration | Objective function | Constraint satisfaction | Penalized functional |
|---|---|---|---|---|---|---|
| μ = 0.5 | DCA | 0.2 | 22 | 1.205379 | 5.231375 | 3.821067 |
| μ = 0.5 | FSA | 0.2 | 50 | 1.6517 | 11.6227 | |
| λ=-2.88 | ECGM | 0.2 | 7 | 1.0956 | 0.4544 | |
| | MECGM | 0.2 | 10 | 1.0715 | 1.1249 | |
| μ = 0.1 | DCA | 0.2 | 7 | 1.255964 | 4.345529 | 5.601494 |
| μ = 0.1 | FSA | 0.2 | 50 | 1.6250 | 11.2990 | |
| λ=-6.00 | ECGM | 0.2 | 7 | 1.4834 | 0.13813 | |
| | MECGM | 0.2 | 4 | 0.7073 | 0.95018 | |
| μ = 1.5 | DCA | 0.2 | 25 | 0.760914 | 3.265003 | 5.658419 |
| μ = 1.5 | FSA | 0.2 | 50 | 1.60017 | 10.9884 | |
| λ =-9.11 | ECGM | 0.2 | 6 | 1.5557 | 0.08652 | |
| | MECGM | 0.2 | 3 | 0.8686 | 1.1616 | |
| μ = 2 | DCA | 0.2 | 1 | 1.330563 | 7.641839 | 16.61424 |
| μ = 2 | ESA | 0.2 | 50 | 1.57684 | 10.6902 | |
| λ = -10.57 | ECGM | 0.2 | 7 | 1.4686 | 0.03531 | |
| | MECGM | 0.2 | 3 | 0.9386 | 1.0477 | |
| μ = 2.5 | DCA | 0.2 | 12 | 1.428915 | 3.642317 | 10.53471 |
| μ = 2.5 | FSA | 0.2 | 50 | 1.55497 | 10.402 | |
| λ = -10.37 | ECGM | 0.2 | 6 | 1.58206 | 8.1262*10-3 | |
| | MECGM | 0.2 | 2 | 1.0178 | 1.6313 | |

Table 2: Numerical results for hypothetical problem p2 compared to the exact solution, 0.7641

| Penalty constants | Algor | Stepsize | Iteration | Objective function | Constraint satisfaction | Penalized functional |
|---|---|---|---|---|---|---|
| μ = 0.5 | DCA | 0.2 | 1 | 1.6 | 0.8 | 3.821067 |
| μ = 0.5 | FSA | 0.2 | 50 | 1.9777 | 0.9789 | |
| λ=-2.88 | ECGM | 0.2 | 3 | 0.79989 | 0.01313 | |
| μ = 0.1 | DCA | 0.2 | 3 | 0.8303 | 4.345529 | 5.601494 |
| μ = 0.1 | FSA | 0.2 | 50 | 1.9742 | 11.2990 | |
| λ=-6.00 | ECGM | 0.2 | 4 | 0.72768 | 0.13813 | |
| μ = 1.5 | DCA | 0.2 | 7 | 0.8676 | 3.265003 | 5.658419 |
| μ = 1.5 | FSA | 0.2 | 50 | 1.971011 | 10.9884 | |
| λ =-9.11 | ECGM | 0.2 | 4 | 0.97258 | 0.08652 | |
| μ = 2 | DCA | 0.2 | 11 | 0.8769 | 7.641839 | 16.61424 |
| μ = 2 | ESA | 0.2 | 50 | 1.9677 | 10.6902 | |
| λ = -10.57 | ECGM | 0.2 | 7 | 1.98866 | 0.03531 | |
| μ = 2.5 | DCA | 0.2 | 15 | 0.8747 | 3.642317 | 10.53471 |
| μ = 2.5 | FSA | 0.2 | 50 | 1.9645 | 10.402 | |
| λ = -10.37 | ECGM | 0.2 | 6 | 1.92047 | 8.1262*10-3 | |

at 0.79989. In fact, for $\mu = 1.00$, the optimum 0.8303 is attained for DCA comparable to the exact solution 0.7641.

Also, DCA's solution deviation from the exact solution has been within a tolerance level of $\pm 0.1186$ as seen in column 5 from its iterates, while other algorithms's iterates fall outside this tolerance level for some parameters in $1.0 \le \mu \le 2.5$. However, FSA did worst of all the algorithms.

Conclusively, DCA, with its less computational rigour, has performed better absolutely than either FSA or ECGM. So, it is an additional algorithm for the solution of such class of problems under consideration

## REFERENCES


1. Fletcher, R. and C.M. Reeves, 1964. Function minimization by conjugate gradients. Computer J., 7: 149-154.
2. Gianni, D. and Laura, Palogi, 1998. Non-linear programming. Introduction to unconstrained and constrained optimization; Technical report.
3. Gianni, D. and L. Grippo, 1972. A computing algorithm for the epsilon method to identification and Optimal Control Problems, 3: 54-77.
4. Ibiejugba, M.A., 1984. A control operator and some of its applications. J. Optimization, pp:31-47.
5. Olorunsola, S.A., 1995. On two updating methods of the multipliers in the Multipliers imbedding algorithm. SAMS, 21: 51-73.
6. Omolehin, J.O. *et al*., 2006. Jacobian approach to optimal determination of perturbation parameter for gradient method. J. Nig. Assoc. Math. Physics, 10: 363-370.
7. Gutknecht, M.H. and M. Rozloznik, 2002. A framework for generalized conjugate methods-with special emphasis on contributions by Rudiger Weiss. Applied Numeri. Anal., 41: 7-22.