Electronic Journal of Mathematical Analysis and Applications
Vol. 11(2) July 2023, No. 11
ISSN: 2090-729X (online)
http://ejmaa.journals.ekb.eg/

NUMERICAL SOLUTION OF OPTIMAL CONTROL PROBLEMS USING BLOCK METHOD

S. ADAMU

Abstract. Forward-backward sweep approach is used to solve optimal control problems utilizing a collocation hybrid second derivative block method using polynomial approximate solution via pontryagin's principle. The block method is formulated from the discrete linear multistep methods. Also, the forward algorithms, backward algorithm written. The stability properties of the block method are analyzed and proved to be stable, convergent and of order 6. The algorithm is implemented with a written MATLAB code, and three optimal control problems are solved to test the accuracy of the approach, which the numerical examples show that, forward-backward sweep methods together with block method via Pontryagin's principle are more accurate than when solving optimal control problems with the traditional classical Runge-Kutta method. This research work therefore established that block method can be combined with forward backward sweep method using Pontryagin's principle to solve optimal control problems and produce more accurate result than using the traditional classical Runge-Kutta method.

1. Introduction

Most mathematical models used in natural sciences and engineering are based on Ordinary Differential Equations (ODEs). Traditionally, solutions to these differential equations can be obtained using analytical methods, however, solutions to certain differential equations are very difficult except the approximate solution by the application of numerical methods [1].

Optimal control problems can be used to model many classes of phenomena, such as population dynamics, continuum mechanics of materials with memory, economic problems, the spread of epidemics, non-local problems of diffusion and heat conduction problem [2,3].

2020 *Mathematics Subject Classification.* 65L05, 65L06, 49N05.
Key words and phrases. Runge-Kutta method, hybrid second derivative block method, optimal control problems, forward-backward sweep method, pontryagin's principle.
Submitted Jun. 2023.

In this paper, we consider Optimal Control (OC) problem that optimizes the performance index

$$\min_{u(\cdot)} \; J\left[x(\cdot), u(\cdot)\right] = \int_{t_0}^{t_f} f(x, x(t), u(t))dt, \tag{1.1}$$

subject to dynamic system

$$\dot{x} = g(t, x(t), u(t)), \;\; x \in [t_0, t_f], \tag{1.2}$$

$$x(t_0) = x_0, \qquad x(t_f) \text{ is free and unrestricted.}$$

J is the value of the functional to be optimized, $x(t)$ and $u(t)$ are real valued functions, $f(t, x, u)$ and $g(t, x, u)$ are continuously differentiable functions.

OC are applied in fields of ordinary differential equations, partial differential equations, discrete equations, stochastic differential equations, integro difference equations, combination of discrete and continuous systems, to solve problems of physical systems, aerospace, economics and management, biology and medicine [4]. Thus, OC with ODEs have wider applications in sciences and engineering, hence the methods for the solution of optimal control problems are important. Numerical methods for solving equations $(1.1) - (1.2)$ have enable the simulation of highly complex real world scenarios [4, 5].

Forward Backward Sweep (FBS) is an iterative method named based on how the algorithm solves the problem's state and adjoint ODE's. Given an approximation of the control function, FBS first solves the state 'forward' in time, then solves the adjoint backward. Once it has found the state and adjoint functions, the control is updated based on the Hamiltonian (H) and then the state, control, and adjoint are tested for convergence against a user provided tolerance and depending on that, the algorithm either starts the process over using the updated control or the algorithm terminates with the final approximations for the state, adjoint, and control functions considered as the solution to the optimal control problem [6].

Forward-Backward Sweep methods are been used to take advantage of certain characteristics of the optimality system. First, given an initial condition for the state x but a final time condition for the adjoint λ. Second, g is a function of t, x, and u only. Values for λ are not needed to solve the differential equation for x using a standard ODE solver. Taking this into account, the method presented here is very intuitive, it is generally referred to as the Forward-Backward Sweep methods [3].

Pontryagin's maximum (or minimum) principle (PMP) is used in optimal control theory to find the best possible control for taking a dynamical system from one state to another, especially in the presence of constraints for the state or input controls [7].

This paper adopt forward-backword sweep methods. The process begins with an initial guess on the control variable. Then, simultaneously, the state equations are solved forward in time and the adjoint equations are solved backward in time. The control is updated by inserting the new values of states and adjoints into its characterization, and the process is repeated until convergence occurs [4].

Considering related works, [3], [4] and [6] applied forward-backward sweep methods using classical Runge–Kutta to solve optimal control problems with initial value problems via Pontryagin's principle. [8] developed higher derivative methods for the solution of stiff initial value problem of ODEs, second and higher derivative methods are reported to be more effective and economical to implement.
In this research, we improve on the method of [6] by developing Hybrid Second Derivative Block Method (HSDBM) for the solution of optimal control problems in combination with forward backword sweep methods.

1.1. **Necessary condition for the solution of optimal control problem.** [6]

To solve basic optimal control problem, a set of necessary conditions must be satisfied. These conditions are derived from Hamiltonian, H, given as

$$H(t,x,u,\lambda) = f(t,x,u) + \lambda g(t,x,u), \tag{1.3}$$

where λ denotes the adjoint and is dependent on t, x and u.These conditions are;

$$\frac{\partial H}{\partial u} = 0 \text{ at } u^* \implies f_u + \lambda g_u = 0, \qquad \text{optimality condition} \tag{1.4}$$

$$\lambda' = \frac{\partial H}{\partial x} \implies \lambda' = h(t,x,\lambda,u) - (f_x + \lambda g_x), \qquad \text{adjoint equation} \tag{1.5}$$

$$\lambda(t_1) = 0. \qquad \text{Transversality condition} \tag{1.6}$$

$$\begin{cases} x' = g(t_1,x,u), \\ x(t_0) = x_0. \end{cases} \qquad \text{Dynamics of the state equation} \tag{1.7}$$

With these conditions, there is now a process on how to solve the standard problems.

2. Methodology

2.1. **Development of Hybrid Second Derivative Block Method (HSDBM).**

We considered a polynomial approximate solution of the form

$$y\left(x\right) = \sum_{n=0}^{k} a_n x^n, \tag{2.1}$$

with first and second derivative given as

$$y'(x) = \sum_{n=1}^{k} n a_n x^{n-1}, \tag{2.2}$$

$$y''(x) = \sum_{n=2}^{k} n(n-1) a_n x^{n-2}, \tag{2.3}$$

Evaluate equation (2.1) at point x_n, equation (2.2) at $\left[x_n, x_{n+\frac{1}{3}}, x_{n+\frac{2}{3}}, x_{n+1}\right]$ and equation (2.3) at points $[x_{n+\frac{2}{3}}, x_{n+1}]$ to give

$$\begin{bmatrix} 1 & x_n & x_n^2 & x_n^3 & x_n^4 & x_n^5 & x_n^6 \\ 0 & 1 & 2x_n & 3x_n^2 & 4x_n^3 & 5x_n^4 & 6x_n^5 \\ 0 & 1 & 2x_{n+\frac{1}{3}} & 3x_{n+\frac{1}{3}}^2 & 4x_{n+\frac{1}{3}}^3 & 5x_{n+\frac{1}{3}}^4 & 6x_{n+\frac{1}{3}}^5 \\ 0 & 1 & 2x_{n+\frac{2}{3}} & 3x_{n+\frac{2}{3}}^2 & 4x_{n+\frac{2}{3}}^3 & 5x_{n+\frac{2}{3}}^4 & 6x_{n+\frac{2}{3}}^5 \\ 0 & 1 & 2x_{n+1} & 3x_{n+1}^2 & 4x_{n+1}^3 & 5x_{n+1}^4 & 6x_{n+1}^5 \\ 0 & 0 & 2 & 6x_{n+\frac{2}{3}} & 12x_{n+\frac{2}{3}}^2 & 20x_{n+\frac{2}{3}}^3 & 30x_{n+\frac{2}{3}}^4 \\ 0 & 0 & 2 & 6x_{n+1} & 12x_{n+1}^2 & 20x_{n+1}^3 & 30x_{n+1}^4 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{bmatrix} = \begin{bmatrix} y_n \\ f_n \\ f_{n+\frac{1}{3}} \\ f_{n+\frac{2}{3}} \\ f_{n+1} \\ g_{n+\frac{2}{3}} \\ g_{n+1} \end{bmatrix}.$$

2.1.1. *Continuous scheme.*

$$y_{n+t} = \alpha_0(t)y_n+\beta_0(t)f_n+\beta_{\frac{1}{3}}(t)f_{n+\frac{1}{3}}+\beta_{\frac{2}{3}}(t)f_{n+\frac{2}{3}}+\beta_1(t)f_{n+1}+\gamma_{\frac{2}{3}}(t)g_{n+\frac{2}{3}}+\gamma_1(t)g_{n+1}, \tag{2.4}$$

where,

$\alpha_0 = 1,$

$\beta_0\left(t\right) = -\dfrac{1}{240}t\left(960t - 1940t^2 + 2115t^3 - 1188t^4 + 270t^5 - 240\right),$

$\beta_{\frac{1}{3}}\left(t\right) = \dfrac{9}{16}t^2\left(-80t + 111t^2 - 72t^3 + 18t^4 + 24\right),$

$\beta_{\frac{2}{3}}\left(t\right) = \dfrac{27}{80}t^3\left(75t - 84t^2 + 30t^3 - 20\right),$

$\beta_1\left(t\right) = -\dfrac{1}{240}t^2\left(-10\,480t + 18\,945t^2 - 15\,336t^3 + 4590t^4 + 2280\right),$

$\gamma_{\frac{2}{3}}\left(t\right) = \dfrac{3}{40}t^2\left(-260t + 435t^2 - 324t^3 + 90t^4 + 60\right),$

$\gamma_1\left(t\right) = \dfrac{1}{120}t^2\left(-560t + 1035t^2 - 864t^3 + 270t^4 + 120\right).$

2.1.2. *Discrete schemes.*

When $t = \frac{1}{3}$,

$$y_{n+\frac{1}{3}} = y_n+\frac{637}{6480}hf_n+\frac{65}{144}hf_{n+\frac{1}{3}}-\frac{29}{720}hf_{n+\frac{2}{3}}-\frac{1141}{6480}hf_{n+1}+\frac{97}{1080}h^2g_{n+\frac{2}{3}}+\frac{59}{3240}h^2g_{n+1}. \tag{2.5}$$

When $t = \frac{2}{3}$,

$$y_{n+\frac{2}{3}} = y_n + \frac{13}{135}hf_n + \frac{5}{9}hf_{n+\frac{1}{3}} + \frac{7}{45}hf_{n+\frac{2}{3}} - \frac{19}{135}hf_{n+1} + \frac{8}{135}h^2g_{n+\frac{2}{3}} + \frac{2}{135}h^2g_{n+1}. \tag{2.6}$$

When $t = 1$,

$$y_{n+1} = y_n + \frac{23}{240}hf_n + \frac{9}{16}hf_{n+\frac{1}{3}} + \frac{27}{80}hf_{n+\frac{2}{3}} + \frac{1}{240}hf_{n+1} + \frac{3}{40}h^2g_{n+\frac{2}{3}} + \frac{1}{120}h^2g_{n+1}. \tag{2.7}$$

Writing (2.5),(2.6) and (2.7) in block, we have

$$A^{(1)}Y_{m+1} = A^{(0)}Y_m + hB^{(0)}F_m + hB^{(1)}F_{m+1} + h^2\gamma^{(1)}G_{m+1}, \tag{2.8}$$

where,

$$Y_{m+1}=\left[y_{n+\frac{1}{3}} \quad y_{n+\frac{2}{3}} \quad y_{n+1}\right]^T, Y_m=\left[y_{n-1} \quad y_{n-2} \quad y_n\right]^T, F_m=\left[f_{n-1} \quad f_{n-2} \quad f_n\right]^T,$$

$$F_{m+1}=\left[f_{n+\frac{1}{3}} \ f_{n+\frac{2}{3}} \ f_{n+1}\right]^T, \quad G_{m+1}=\left[g_{n-1} \ g_{n+\frac{2}{3}} \ g_{n+1}\right]^T,$$

$$A^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ A^{(0)} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, B^{(0)} = \begin{bmatrix} 0 & 0 & \frac{637}{6480} \\ 0 & 0 & \frac{13}{135} \\ 0 & 0 & \frac{23}{240} \end{bmatrix},$$

$$B^{(1)} = \begin{bmatrix} \frac{65}{144} & -\frac{29}{720} & -\frac{1141}{6480} \\ \frac{5}{9} & \frac{7}{45} & -\frac{19}{135} \\ \frac{9}{16} & \frac{27}{80} & \frac{1}{240} \end{bmatrix} \text{ and } \gamma^{(1)} = \begin{bmatrix} 0 & \frac{97}{1080} & \frac{59}{3240} \\ 0 & \frac{8}{135} & \frac{2}{135} \\ 0 & \frac{3}{40} & \frac{1}{120} \end{bmatrix}.$$

Representing(2.8) in table to give

$$\left[\begin{array}{c|cccc|cccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{637}{6480} & \frac{65}{144} & -\frac{29}{720} & -\frac{1141}{6480} & 0 & 0 & \frac{97}{1080} & \frac{59}{3240} \\ \frac{2}{3} & \frac{13}{135} & \frac{5}{9} & \frac{7}{45} & -\frac{19}{135} & 0 & 0 & \frac{8}{135} & \frac{2}{135} \\ 1 & \frac{23}{240} & \frac{9}{16} & \frac{27}{80} & \frac{1}{240} & 0 & 0 & \frac{3}{40} & \frac{1}{120} \\ \hline & \frac{23}{240} & \frac{9}{16} & \frac{27}{80} & \frac{1}{240} & & & & \end{array}\right].$$

2.2. **Algorithm for FBS implementation of HSDM.**

2.2.1. *Forward algorithm.*

for $i = 1 : N$

$$f_n = f\left(t_i, x_i, u_i\right)$$
$$f_{n+\frac{1}{3}} = f\left(t_i + \frac{1}{3}h, x_i + \frac{1}{3}h, u_i + \frac{u_{i+1} - u_i}{3}\right)$$
$$f_{n+\frac{2}{3}} = f\left(t_i + \frac{2}{3}h, x_i + \frac{2}{3}h, u_i + \frac{2}{3}\left(u_{i+1} - u_i\right)\right)$$
$$f_{n+1} = f\left(t_i + h, x_i + h, u_{i+1}\right)$$

$$g_{n+\frac{2}{3}} = g\left(t_i + \frac{2}{3}h, x_i + \frac{2}{3}h, u_i + \frac{2}{3}\left(u_{i+1} - u_i\right)\right)$$
$$g_{n+1} = g\left(t_i + h, x_i + h, u_{i+1}\right)$$
$$X_{\frac{1}{3}} = X_i + h\left(\frac{637}{6480}f_n + \frac{65}{144}f_{n+\frac{1}{3}} - \frac{29}{720}f_{n+\frac{2}{3}} - \frac{1141}{6480}f_{n+1}\right) + h^2\left(\frac{97}{1080}g_{n+\frac{2}{3}} + \frac{59}{3240}g_{n+1}\right)$$
$$X_{\frac{2}{3}} = X_i + h\left(\frac{13}{135}f_n + \frac{5}{9}f_{n+\frac{1}{3}} + \frac{7}{45}f_{n+\frac{2}{3}} - \frac{19}{135}f_{n+1}\right) + h^2\left(\frac{8}{135}g_{n+\frac{2}{3}} + \frac{2}{135}g_{n+1}\right)$$
$$X_1 = X_i + h\left(\frac{23}{240}f_n + \frac{9}{16}f_{n+\frac{1}{3}} + \frac{27}{80}f_{n+\frac{2}{3}} + \frac{1}{240}f_{n+1}\right) + h^2\left(\frac{3}{40}g_{n+\frac{2}{3}} + \frac{1}{120}g_{n+1}\right)$$
$$K_1 = f\left(t_i, x_i, u_i\right),\ K_2 = f\left(t_i + \frac{1}{3}h, X_{\frac{1}{3}}, u_i + \frac{1}{3}\left(u_{i+1} - u_i\right)\right)$$
$$K_3 = f\left(t_i + \frac{2}{3}h, X_{\frac{2}{3}}, u_i + \frac{2}{3}\left(u_{i+1} - u_i\right)\right)$$
$$K_4 = f\left(t_i + h, X_1, u_{i+1}\right)$$
$$X(i+1) = X(i) + h\left(\frac{23}{240}K_1 + \frac{9}{16}K_2 + \frac{27}{80}K_3 + \frac{1}{240}K_4\right)$$

2.2.2. *Backward algorithm.*

for $i = 1 : N$
$i = N + 2 - j$

$$f_n = f\left(t_i, x_i, \lambda_i, u_i\right)$$
$$f_{n+\frac{1}{3}} = f\left(t_i - \frac{1}{3}h, x_i + \frac{1}{3}\left(x_{i+1} - x_i\right), \lambda_i - \frac{1}{3}h, u_i + \frac{1}{3}\left(u_{i+1} - u_i\right)\right)$$
$$f_{n+\frac{2}{3}} = f\left(t_i - \frac{2}{3}h, x_i + \frac{2}{3}\left(x_{i+1} - x_i\right), \lambda_i - \frac{2}{3}h, u_i + \frac{2}{3}\left(u_{i+1} - u_i\right)\right)$$
$$f_{n+1} = f\left(t_i - h, x_{i-1}, \lambda_i - h, u_i - h\right)$$
$$g_{n+\frac{2}{3}} = g\left(t_i - \frac{2}{3}h, x_i + \frac{2}{3}\left(x_{i+1} - x_i\right), \lambda_i - \frac{2}{3}h, u_i + \frac{2}{3}\left(u_{i+1} - u_i\right)\right)$$
$$g_{n+1} = g\left(t_i - h, x_{i-1}, \lambda_i - h, u_i - h\right)$$
$$\lambda_{\frac{1}{3}} = \lambda_i - h\left(\frac{637}{6480}f_n + \frac{65}{144}f_{n+\frac{1}{3}} - \frac{29}{720}f_{n+\frac{2}{3}} - \frac{1141}{6480}f_{n+1}\right) - h^2\left(\frac{97}{1080}g_{n+\frac{2}{3}} + \frac{59}{3240}g_{n+1}\right)$$
$$\lambda_{\frac{2}{3}} = \lambda_i - h\left(\frac{13}{135}f_n + \frac{5}{9}f_{n+\frac{1}{3}} + \frac{7}{45}f_{n+\frac{2}{3}} - \frac{19}{135}f_{n+1}\right) - h^2\left(\frac{8}{135}g_{n+\frac{2}{3}} + \frac{2}{135}g_{n+1}\right)$$
$$\lambda_1 = \lambda_i - h\left(\frac{23}{240}f_n + \frac{9}{16}f_{n+\frac{1}{3}} + \frac{27}{80}f_{n+\frac{2}{3}} + \frac{1}{240}f_{n+1}\right) - h^2\left(\frac{3}{40}g_{n+\frac{2}{3}} + \frac{1}{120}g_{n+1}\right)$$
$$K_1 = f\left(t_i, x_i, \lambda_i, u_i\right)$$
$$K_2 = f\left(t_i - \frac{1}{3}h, x_i + \frac{1}{3}\left(x_{i+1} - x_i\right), \lambda_{\frac{1}{3}}, u_i + \frac{1}{3}\left(x_{i+1} - x_i\right)\right)$$
$$K_3 = f\left(t_i - \frac{2}{3}h, x_i + \frac{2}{3}\left(x_{i+1} - x_i\right), \lambda_{\frac{2}{3}}, u_i + \frac{2}{3}\left(x_{i+1} - x_i\right)\right)$$
$$K_4 = f\left(t_i - h, x_{i-1}, \lambda_1, u_{i-1}\right)$$
$$\lambda(i-1) = \lambda(i) - h\left(\frac{23}{240}K_1 + \frac{9}{16}K_2 + \frac{27}{80}K_3 + \frac{1}{240}K_4\right)$$

3. Stability Properties of the Method

3.1. **Order of the block method.** Evaluating equations (2.5) , (2.6) and (2.7) in a Taylor series about x_n gives

$$L\left[y\left(x\right);h\right]=y_{n+t}-\alpha_0(t)y_n-\beta_0(t)f_n-\beta_{\frac{1}{3}}(t)f_{n+\frac{1}{3}}-\beta_{\frac{2}{3}}(t)f_{n+\frac{2}{3}}-\beta_1(t)_{n+1}-\gamma_{\frac{2}{3}}(t)g_{n+\frac{2}{3}}-\gamma_1(t)g_{n+1}=0,$$

where,

$$h^{p+1}\neq 0 \text{ and } p+1=7.$$

Therefore, the order of the HSDBM is $p=[6,6,6]^T$ with error constant

$$\text{Error Constant}=\left[-\frac{1067}{661\,348\,800}h^7,-\frac{59}{41\,334\,300}h^7,-\frac{11}{8164\,800}h^7\right]^T.$$

3.2. **Zero-stability.** Since the roots $z_s, s=1,2,3,...n$ of the first characteristics polynomial $\rho\left(z\right)$ of HSDBM, defined by

$$\rho\left(\lambda\right)=\det\left[A^{(1)}\lambda-A^{(0)}\right]=0,$$

are $\lambda=\left[0,0,1\right]^T$ respectively. Hence the HSDBM is zero stable.

3.3. **Consistency of the block method.** Since the block method (2.8) has order $p=6>1$, therefore, the block method is consistent.

3.4. **Convergence of the block method.** The block method (2.8) is convergent since it consistent and zero-stable.

3.5. **Region of absolute stability of the block method.** The region of absolute stability of the block method is shown in Figure (1) below.

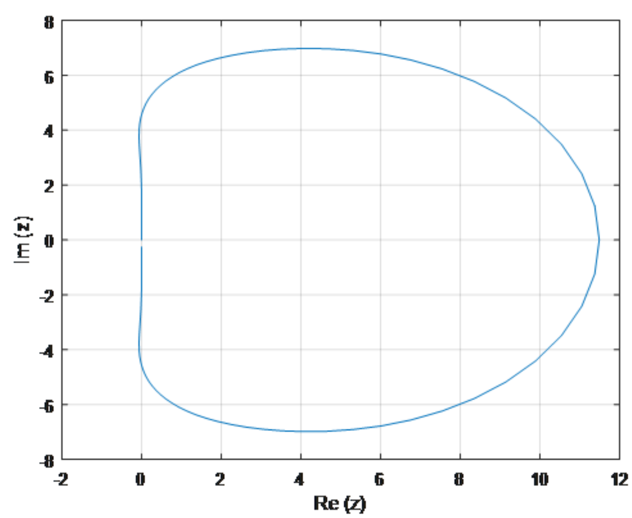


Figure 1. Region of Absolute Stability of HSDBM

4. Numerical Experiment

We considered the following problems to test the efficiency of the developed methods. It is assumed that $T = 1$.

NOTATIONS

The following notations are used in table (1) to (3)

x	Point of Evaluation
HSDBM	Hybrid Second Derivative Block Method
CRKM	Classical Runge Kutta Method in [6] and [3]
Err	Absolute Error

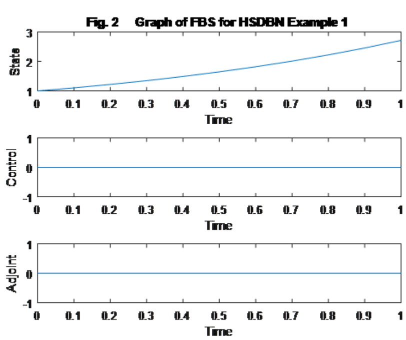
Example 4.1. *[3]*

$$\min_{u} \int_0^1 u(t)^2 dt,$$

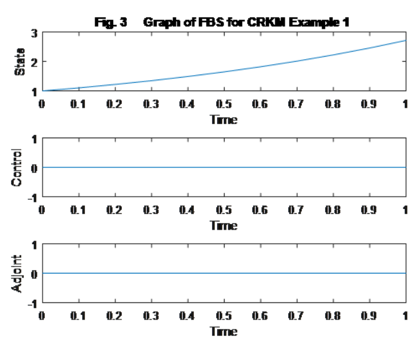
$$\textit{subject to} \begin{cases} x'(t) = x(t) + u(t), \\ x(0) = 1, \quad x(1) \textit{ free}, \end{cases}$$

with the optimal solution

$$\begin{aligned} x^*(t) &\equiv e^t, \\ u^*(t) &\equiv 0. \end{aligned}$$



(A) Graph of FBS for HSDBM Example 1



(B) Graph of FBS for CRKM Example 1

Figure 2

Solution 1.

Table 1: Result of State for Example 1

x	*Exact Sol.*	*CRKM*	*HSDBM*	*Err in CRKM*	*Err in HSDBM*	*Err in [9]*
0	1.0000e+00	1.0000e+00	1.0000e+00	0.0000	0.0000	0.0000
1	1.1052e+00	1.1052e+00	1.1052e+00	8.4742e-08	1.5743e-13	3.9618e-11
2	1.2214e+00	1.2214e+00	1.2214e+00	1.8731e-07	3.4817e-13	8.7570e-11
3	1.3499e+00	1.3499e+00	1.3499e+00	3.1051e-07	5.7709e-13	1.4517e-10
4	1.4918e+00	1.4918e+00	1.4918e+00	4.5756e-07	8.5043e-13	2.1392e-10
5	1.6487e+00	1.6487e+00	1.6487e+00	6.3210e-07	1.1748e-12	2.9552e-10
6	1.8221e+00	1.8221e+00	1.8221e+00	8.3830e-07	1.5583e-12	3.9191e-10
7	2.0138e+00	2.0138e+00	2.0138e+00	1.0809e-06	2.0091e-12	5.0532e-10
8	2.2255e+00	2.2255e+00	2.2255e+00	1.3652e-06	2.5375e-12	6.3825e-10
9	2.4596e+00	2.4596e+00	2.4596e+00	1.6974e-06	3.1553e-12	7.9354e-10
10	2.7183e+00	2.7183e+00	2.7183e+00	2.0843e-06	3.8747e-12	9.7445e-10

It can be observed from Table 1 that, the new method competes effectively with the existing classical Runge-Kutta method and [9] in example 1, after 8 iterations. From the absolute error, it shows that HSDBM give better approximation than the existing CRKM and [9]. The results of absolute error of control for example 1 in the existing classical Runge-Kutta method and the new method were all zero in example 1 since the exact solution of the control in example 1 is 0.

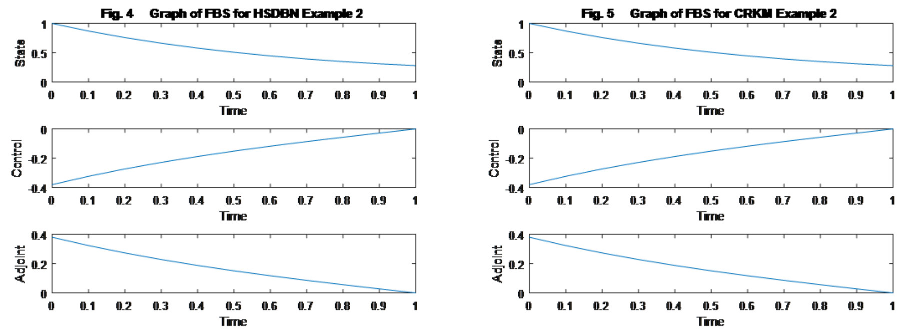
Example 4.2. *[6]*

$$\min_u \frac{1}{2}\int_0^1 x(t)^2 + u(t)^2 dt,$$

$$\textit{subject to} \quad \begin{cases} x'(t) = -x(t) + u(t), \\ x(0) = 1, \end{cases}$$

with the optimal solution

$$\begin{aligned} x^*(t) &= \frac{\sqrt{2}\cosh\left(\sqrt{2}(t-1)\right) - \sinh\left(\sqrt{2}(t-1)\right)}{\sqrt{2}\cosh\left(\sqrt{2}\right) + \sinh\left(\sqrt{2}\right)}, \\ u^*(t) &= -\frac{\sinh\left(\sqrt{2}(t-1)\right)}{\sqrt{2}\cosh\left(\sqrt{2}\right) + \sinh\left(\sqrt{2}\right)}. \end{aligned}$$



(A) Graph of FBS for HSDBM Example 2 (B) Graph of FBS for CRKM Example 2

FIGURE 3

Solution 2.

Table 2: Result of State for Example 2

x	Exact Sol.	CRKM	HSDBM	Err in CRKM	Err in HSDBM	Err in [9]
0	1.0000e+00	1.0000e+00	1.0000e+00	0.0000	0.0000	0.0000
1	8.7097e-01	8.6964e-01	8.7100e-01	1.3285e-03	3.0592e-05	1.2648e-04
2	7.5939e-01	7.5730e-01	7.5941e-01	2.0903e-03	1.4040e-05	1.9834e-04
3	6.6303e-01	6.6067e-01	6.6299e-01	2.3579e-03	3.7085e-05	2.2094e-04
4	5.7994e-01	5.7776e-01	5.7983e-01	2.1850e-03	1.1344e-04	1.9942e-04
5	5.0848e-01	5.0687e-01	5.0827e-01	1.6074e-03	2.0817e-04	1.3859e-04
6	4.4720e-01	4.4656e-01	4.4688e-01	6.4495e-04	3.1633e-04	4.2818e-05
7	3.9488e-01	3.9558e-01	3.9445e-01	6.9805e-04	4.3438e-04	8.3998e-05
8	3.5047e-01	3.5291e-01	3.4991e-01	2.4326e-03	5.5987e-04	2.3841e-04
9	3.1308e-01	3.1767e-01	3.1239e-01	4.5849e-03	6.9111e-04	4.1741e-04
10	2.8197e-01	2.8744e-01	2.8114e-01	5.4701e-03	8.2697e-04	6.1829e-04

Table 3: Result of Control for Example 2

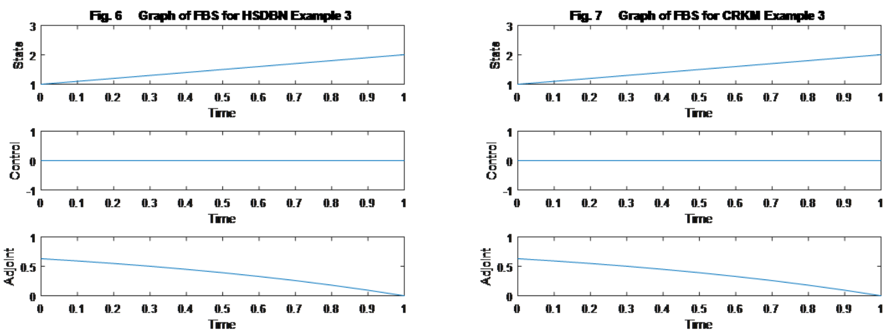
x	Exact Sol.	CRKM	HSDBM	Err in CRKM	Err in HSDBM	Err in [9]
0	3.8582e-01	4.0195e-01	3.8020e-01	7.8777e-01	7.6602e-01	7.7892e-01
1	3.2806e-01	3.3938e-01	3.2347e-01	6.6744e-01	6.5153e-01	6.6222e-01
2	2.7687e-01	2.8368e-01	2.7316e-01	5.6055e-01	5.5003e-01	5.5880e-01
3	2.3123e-01	2.3368e-01	2.2826e-01	4.6492e-01	4.5949e-01	4.6659e-01
4	1.9023e-01	1.8834e-01	1.8789e-01	3.7856e-01	3.7812e-01	3.8375e-01
5	1.5303e-01	1.4669e-01	1.5124e-01	2.9972e-01	3.0428e-01	3.0862e-01
6	1.1890e-01	1.0785e-01	1.1759e-01	2.2675e-01	2.3649e-01	2.3971e-01
7	8.7152e-02	7.0989e-02	8.6248e-02	1.5814e-01	1.7340e-01	1.7564e-01
8	5.7149e-02	3.5300e-02	5.6597e-02	9.2449e-02	1.1375e-02	1.1513e-01
9	2.8291e-02	0.0000	2.8039e-02	2.8291e-02	5.6330e-02	5.6968e-02
10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Again, it can be observed from Table 2 that, the new method competes effectively with the existing classical Runge-Kutta method and [9] in example 2 after 3 iterations. Table 3 show the results of absolute error of control for example 2 in the existing classical Runge-Kutta method, [9] and the new method, it was also observed that HSDBM perform better.

Example 4.3. *[6]*

$$\min_{u} \int_0^1 x(t) + u(t)dt,$$

$$\textit{subject to} \begin{cases} x'(t) = 1 - u(t), \\ x(0) = 1, \end{cases}$$



(A) Graph of FBS for HSDBM Example 3 (B) Graph of FBS for CRKM Example 3

Figure 4

Solution 3.

Table 4: Result of State for Example 3

x	CRKM	HSDBM	[9]
0	1.0000e+00	1.0000e+00	1.0000e+00
1	1.1000e+00	1.1008e+00	1.0990e+00
2	1.2000e+00	1.2017e+00	1.1980e+00
3	1.3000e+00	1.3025e+00	1.2970e+00
4	1.4000e+00	1.4033e+00	1.3960e+00
5	1.5000e+00	1.5042e+00	1.4950e+00
6	1.6000e+00	1.6050e+00	1.5940e+00
7	1.7000e+00	1.7058e+00	1.6930e+00
8	1.8000e+00	1.8067e+00	1.7920e+00
9	1.9000e+00	1.9075e+00	1.8910e+00
10	2.0000e+00	2.0083e+00	1.9900e+00

Problem 3 has no exact solution, so only the results are presented. It can also be observed from Table 4 that, the new method competes effectively with the existing CRKM in example 3 and [9] after 3 iterations. The results of absolute error of control for example 3 in the existing classical Runge-Kutta method, [9] and the new method were all zero in example 3. This example 3 fail to perform in CRKM using the code developed by [6], so our code for example 3 and the methods perform well when compared to [6].

5. Conclusion

The approximate optimal solution of both control and state functions are obtained by applying forward backward sweep method and solving the necessary conditions derived from Pontryagin's minimum principles. HSDBM for the solution of optimal control system with ODEs is developed using polynomial approximate solution and implemented with the aid of a written MATLAB code. This approach gave a more efficient method of higher order and with larger region of absolute stability as can be seen in Figure 1. The basic properties of the block method is investigated and found to be zero stable, consistent and convergent.

Finally, the effectiveness of the method is tested on some numerical examples and compare the results with the results of Runge Kutta method and [9] as shown in Table 1 to Table 4, and found to be more accurate. One of the advantage of optimal control technique is that, it optimized the given performance index and if there is change in the state, only the code needed to be adjusted in forward backward sweep. Also, the use of Runge Kutta method as in [6] and [3] increases function evaluation, hence block method give fewer function evaluation and also, higher order methods can be developed easily. Therefore, the method developed in this research is faster, more computationally stable, possess better rate of convergence and economical to implement.

References

[1] E. A. Areo, R. B. Adeniyi, A self-starting linear multistep method for direct solution of initial value problems of second order ordinary differential equations. International Journal of Pure and Applied Mathematics, 82 (3), 2013, 345-364.

[2] K. Maleknejad, A. Ebrahimzadeh, Optimal Control of Volterra Integro-Differential Systems Based On Legendre Wavelets and Collocation Method. World Academy of Science, Engineering and Technology International Journal of Mathematical and Computational Sciences, 8(7), 2014, 1040-1044.

[3] S. Lenhart, J. T. Workman, Optimal Control Applied to Biological Models. Chepman & Hall/CRC, Mathematical and Computational Biology Series, 2007.

[4] H. S. Rodrigues, M. T. T. Monteiro, D. F. M. Torres, Systems Theory: Perspectives, Applications and Developments. Nova Science Publishers, 2014.

[5] H. Sofia, F. Rodrigues, Optimal Control and Numerical Optimization Applied to Epidemiological Models. Universidade de Aveiro Departamento de Matemática, 2012.

[6] G. R. Rose, Numerical Methods for Solving Optimal Control Problems. Tennessee Research and Creative Exchange, University of Tennessee, Knoxville, 2015.

[7] V. G. Boltyanskii, R. V. Gamkrelidze, L. S. Pontryagin, Towards a Theory of Optimal Processes. Dokl. Akad. Nauk SSSR. 110 (1), 1956, 7-10.

[8] M. Kida, S. Adamu, O. O. Aduroja, T. P. Pantuvo, Numerical Solution of Stiff and Oscillatory Problems using Third Derivative Trigonometrically Fitted Block Method. Nigerian Society of Physical Sciences, 4(1), 2022, 34-48. DOI:10.46481/jnsps.2022.271.

[9] D. Umar, S. Adamu, A. M. Alkali, U. Sulaiman, One-Step Second Derivative Hybrid Method for Solving Optimal Control Problems using Forward-Backward Sweep Method. Bagale Journal of Physical and Life Sciences, 11(1), 2019, 36-45.

Samuel Adamu
Department of Mathematics, Nigerian Army University Biu, Borno State, Nigeria
Email address: samuel.adamu@naub.edu.ng, malgwisa@gmail.com