Block Method for Solving State-Space Equations of Linear Continuous-Time Control Systems

Raghad Kadhim Salih* **Shymaa Hussain Salih***

Atheer Jawad Kadhim*

Date of acceptance 19/3/2007

Abstract

This paper presents a newly developed method with new algorithms to find the numerical solution of nth-order state-space equations (SSE) of linear continuous-time control system by using block method. The algorithms have been written in Matlab language. The state-space equation is the modern representation to the analysis of continuous-time system. It was treated numerically to the single-input-single-output (SISO) systems as well as multiple-input-multiple-output (MIMO) systems by using fourth-order-six-steps block method. We show that it is possible to find the output values of the state-space method using block method. Comparison between the numerical and exact results has been given for some numerical examples for solving different types of state-space equations using block method for conciliated the accuracy of the results of this method.

Key words: State-space equation, Block method, Control system and Algorithms.

1. Introduction

Control systems are playing vital role in our life for instance: thermostat, automatic control of airplane, etc. The system is a combination of component that act together and perform a certain objective [1,2].

In recent years, automatic control systems have assumed an increasingly important role in the development and advancement of modern civilization and technology. They are employed in numerous applications, such as quality control of manufactured products and machine tooling. The basic control system problem may be described by the simple block diagram shown in figure (1) [1,3,4].

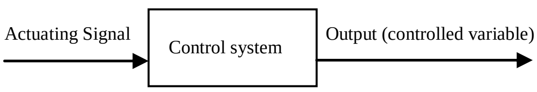

Fig.(1) The basic control system.

Modern control theory adopts what known as state-space equations (SSE) for mathematical representation of systems. Among its different advantages it makes possible to deal with [4,5]:

- Time variant systems.
- Nonlinear systems.
- Multiple-input-multiple-output system.

The linear *state-space equation* is given by:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

where $x(t) \in lR^n$ is the state vector, $u(t) \in lR^m$ is the control input of the system, $y(t) \in lR^p$ is the output of the system, A is the system matrix, B is the control input matrix, C is the output or measurement matrix and D is the direct feed matrix. This description is said to be

*Department of Applied Sciences University of Technology

time-invariant if A,B,C and D are constant matrices. The definitions of state, state variable, state vector and state-space are [1,4] :-

State: The state of a dynamic system is the smallest set of variables, called state variables, such that the knowledge of these variables at time $t=t_0$, together with the input $u(t)$ for $t \geq t_0$, completely determines the behavior of the system for any time $t \geq t_0$.

Thus, the state of a dynamic system at time t is uniquely determined by the state at time t_0 and the input for $t \geq t_0$, and it is independent of the state and input before t_0. Note that, in dealing with linear time-invariant systems, we usually choose the reference time t_0 to be zero.

State variables: The state variables of a dynamic system are the smallest set of variables which determine the state of the dynamic system. If at least n variables $x_1(t), x_2(t), ..., x_n(t)$ are needed to completely describe the behavior of a dynamic system, then such n variables $x_1(t), x_2(t), ..., x_n(t)$ are a set of state variables.

State vector: If n state variables are needed to completely describe the behavior of a given system, then these n state variables can be considered to be the n components of a vector $x(t)$. Such a vector is called a state vector.

State-space: The n-dimensional space whose coordinate axes consist of the x_1 axis, x_2 axis,…, x_n axis is called a state-space. Any state can be represented by a point in the state space.

State-space representation is helpful to represent a complex system by a simple first-order vector-matrix differential equations [2]. State-space method of continuous-time system (i.e. system that can be described by differential equation), was solved by several methods as Laplace transformation and matrix exponential [1]. In this work different types of state-space representation are solved numerically by using fourth order *block method*. The modeling of linear continuous-time systems by using state space method with their solutions have been presented in the following section.

2. State - Space Method :

State-pace method has emerged in the last fifty years, where toward the end of 1950’s, the concept of representing a continuous-time system by a set of first order differential equations has become a standard tool in control theory [2]. State space representation has become popular in the early 1970’s with high-speed digital computers which become more readily available, since this technique uses vector and matrices for system representation, it permits a simple notation that is easily accepted and processed by digital computer [6]. State space method is ideally suited for the analysis of multiple-input-multiple-output systems as well as single-input-single-output systems.

State space method describes the state of the system, where the “state” of a system refers to the past, present and future of the system. In particular, we see how to represent an nth order linear differential equation by a first-order linear, vector-matrix differential equation describing the evolution of an n-dimensional state vector and an equation relating the output to the present state and input. We call these linear equations the state equations and output equations, or a state space representation for the system of differential equations [7,8]. On the other word, if n elements of the vector are a set of state variables, then the vector-matrix differential equation is called a *state equation*.

J. John [9] used state-space representation for solving the pitch controller problem and Dk. James [10] used state space equations to solve the cruise control problem.

In this section, we shall present methods for obtaining state-space representation of linear continuous-time systems.

2.1 State-Space Representation of n^th Order Continuous-Time Systems of Linear Differential Equations In Which The Forcing Function Dose Not Involve Derivative Terms :

A continuous system may be defined as a mathematical abstraction which utilizes three types of variables to represent or model the dynamics of a continuous-time process. The three variables are called the input (i.e. forcing function), the output and the state variables [1,4].

Consider the following *n*th-order dynamic system :

$$y^{(n)}(t)+a_1y^{(n-1)}(t)+...+a_{n-1}\dot{y}(t)+a_ny(t)=u(t) \quad \dots(1)$$

Noting that the knowledge of $y(0),\dot{y}(0),...,y^{(n-1)}(0)$, together with the input or forcing function $u(t)$ for $t\geq 0$, determines completely the future behavior of the system, we may take $y(t),\dot{y}(t),...,y^{(n-1)}(t)$ as a set of n state variables.

Let us define the following state variable:

$$\left.\begin{aligned} x_1(t)&=y(t)\\ x_2(t)&=\dot{y}(t)\\ &\vdots\\ x_n(t)&=y^{(n-1)}(t)\end{aligned}\right\} \quad \dots(2)$$

Then Eq.(1) can be written as

$$\left.\begin{aligned} \dot{x}_1(t)&=x_2(t)\\ \dot{x}_2(t)&=x_3(t)\\ &\vdots\\ \dot{x}_{n-1}(t)&=x_n(t)\\ \dot{x}_n(t)&=-a_nx_1(t)-\cdots-a_1x_n(t)+u(t)\end{aligned}\right\} \dots(3)$$

Or $\quad \dot{x}(t)=Ax(t)+Bu(t) \quad \dots(4)$

where $x\ (t)$ is the $n\times 1$ state vector as given by :

$$x(t)=\begin{bmatrix} x_1(t)\\ x_2(t)\\ \vdots\\ x_n(t)\end{bmatrix},$$

A is the square $(n\times n)$ time-invariant system matrix (i.e. constant system matrix does not depend on the time t), defined by:

$$A=\begin{bmatrix} 0 & 1 & 0 & \dots & 0\\ 0 & 0 & 1 & \dots & 0\\ \vdots & \vdots & \vdots & & \vdots\\ 0 & 0 & 0 & \dots & 1\\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1\end{bmatrix}_{n\times n}$$

and B is the $(n\times 1)$ time-invariant input matrix (i.e. constant input matrix) defined by :

$$B=\begin{bmatrix} 0\\ 0\\ \vdots\\ 0\\ 1\end{bmatrix}_{n\times 1}.$$

The output equation becomes

$$y(t)=\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}\begin{bmatrix} x_1(t)\\ x_2(t)\\ \vdots\\ x_n(t)\end{bmatrix}$$

Or $\quad y(t)=Cx(t) \quad \dots(5)$

where $\quad C=\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}$.

The first-order differential equation, Eq.(4), is the state equation, and the algebraic equation, Eq.(5), is the output equation.

2.2 State-Space Representation of n^th Order Continuous-Time Systems of Linear Differential Equations With (m) Forcing Functions :

Consider the multiple-input-multiple-output (MIMO) linear continuous system shown in figure (2). In this system, $x_1(t),x_2(t),...,x_n(t)$ represent the state variables; $u_1(t),u_2(t),...,u_m(t)$ denote the input variables and $y_1(t),y_2(t),...,y_p(t)$ are the output variables. From Fig.(2) we obtain the system equations as follows [2,7,4]:

$$\left.\begin{aligned}\dot{x}_1(t)&=a_{11}x_1(t)+a_{12}x_2(t)+\cdots+a_{1n}x_n(t)+b_{11}u_1(t)+b_{12}u_2(t)+\cdots+b_{1m}u_m(t)\\\dot{x}_2(t)&=a_{21}x_1(t)+a_{22}x_2(t)+\cdots+a_{2n}x_n(t)+b_{21}u_1(t)+b_{22}u_2(t)+\cdots+b_{2m}u_m(t)\\&\vdots\\\dot{x}_n(t)&=a_{n1}x_1(t)+a_{n2}x_2(t)+\cdots+a_{nn}x_n(t)+b_{n1}u_1(t)+b_{n2}u_2(t)+\cdots+b_{nm}u_m(t)\end{aligned}\right\}\text{...(6)}$$

where the a's and b's are constants. Expression (6) is a set of first order differential equations which may be put into the convenient matrix form:

$$\dot{x}(t)=Ax(t)+Bu(t) \qquad \text{...(7)}$$

where $x(t)$ is the $n\times 1$ state vector as given by :

$$x(t)=\begin{bmatrix}x_1(t)\\x_2(t)\\\vdots\\x_n(t)\end{bmatrix},$$

while $u(t)$ is the $m\times 1$ input vector as given by :

$$u(t)=\begin{bmatrix}u_1(t)\\u_2(t)\\\vdots\\u_m(t)\end{bmatrix}$$

A is the square $(n\times n)$ time-invariant system matrix defined by:

$$A=\begin{bmatrix}a_{11}&a_{12}&\dots&a_{1n}\\a_{21}&a_{22}&\dots&a_{2n}\\\vdots&&&\\a_{n1}&a_{n2}&\dots&a_{nn}\end{bmatrix}_{n\times n}$$

and B is the $(n\times m)$ time-invariant input matrix defined by :

$$B=\begin{bmatrix}b_{11}&b_{12}&\dots&b_{1m}\\b_{21}&b_{22}&\dots&b_{2m}\\\vdots&&&\\b_{n1}&b_{n2}&\dots&b_{nm}\end{bmatrix}_{n\times m}$$

Eq.(7) is the state equation for the system.

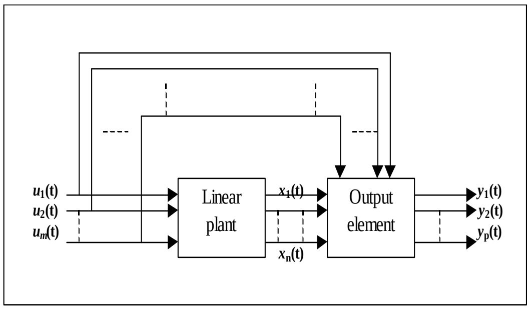

Fig.(2) Multiple-input-multiple-output linear continuous system.

Similarly, the output variables at continuous time t are linear combinations of the values of the input and state variables. That is :

$$\left.\begin{aligned}y_1(t)&=c_{11}x_1(t)+c_{12}x_2(t)+\cdots+c_{1n}x_n(t)+d_{11}u_1(t)+d_{12}u_2(t)+\cdots+d_{1m}u_m(t)\\y_2(t)&=c_{21}x_1(t)+c_{22}x_2(t)+\cdots+c_{2n}x_n(t)+d_{21}u_1(t)+d_{22}u_2(t)+\cdots+d_{2m}u_m(t)\\&\vdots\\y_p(t)&=c_{p1}x_1(t)+c_{p2}x_2(t)+\cdots+c_{pn}x_n(t)+d_{p1}u_1(t)+d_{p2}u_2(t)+\cdots+d_{pm}u_m(t)\end{aligned}\right\}\text{..(8)}$$

This set of equations may be put into the matrix form :

$$y(t)=Cx(t)+Du(t) \qquad \text{...(9)}$$

where $y(t)$ is the $p\times 1$ output vector as given by expression

$$y(t)=\begin{bmatrix}y_1(t)\\y_2(t)\\\vdots\\y_p(t)\end{bmatrix},$$

C is the $p\times n$ time-invariant output matrix defined by :

$$C=\begin{bmatrix}c_{11}&c_{12}&\dots&c_{1n}\\c_{21}&c_{22}&\dots&c_{2n}\\\vdots&&&\\c_{p1}&c_{p2}&\dots&c_{pn}\end{bmatrix}_{p\times n},$$

and D is the $p\times m$ time-invariant transmission matrix defined by :

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m} \\ d_{21} & d_{22} & \dots & d_{2m} \\ \vdots & & & \\ d_{p1} & d_{p2} & \dots & d_{pm} \end{bmatrix}_{p\times m} .$$

Eq.(9) is the output equation for the system. The matrices A, B, C and D completely characterize the system dynamics.

Eq.(7) and Eq.(9) are the state-space equation of the continuous system. Note that, when the technique :

$$\left. \begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \right\} \quad \dots(10)$$

1- Has one input (m = 1) and one output (p = 1), then the system is called system with single-input-single-output (SISO).

2- Has one input (m = 1) and (p) outputs, then the system is called system with single-input-multiple-output (SIMO).

3- Has (m) inputs and one output (p = 1), then the system is called system with multiple-input-single-output (MISO).

4- Has (m) inputs and (p) outputs, then the system is called system with multiple-input-multiple-output (MIMO).

A block diagram representation of the system defined by Eq.(10) is shown in figure (3). Double lines are used in the diagram to indicate vector quantities [1,4].

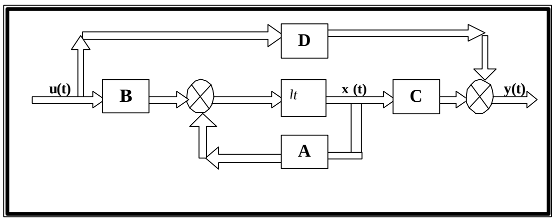

Fig.(3) Block diagram of the continuous-time system described by state-space technique in Eq.(7) and Eq.(9).

2.3 State-Space Representation of nth Order Continuous-Time Systems of Linear Differential Equations In Which The Forcing Function Involves Derivative Terms :

If the differential equation of the system involves derivatives of the forcing function, such as

$$y^{(n)}(t) + a_1 y^{(n-1)}(t) + \dots + a_{n-1}\dot{y}(t) + a_n y(t) = b_0 u^{(n)}(t) + b_1 u^{(n-1)}(t) + \cdots + b_{n-1}\dot{u}(t) + b_n u(t) \quad \dots(11)$$

then, we define the following n variables as a set of n state variables [1,2,5]:

$$\left. \begin{aligned} x_1(t) &= y(t) - \beta_0 u(t) \\ x_2(t) &= \dot{y}(t) - \beta_0 \dot{u}(t) - \beta_1 u(t) = \dot{x}_1(t) - \beta_1 u(t) \\ x_3(t) &= \ddot{y}(t) - \beta_0 \ddot{u}(t) - \beta_1 \dot{u}(t) - \beta_2 u(t) = \dot{x}_2(t) - \beta_2 u(t) \\ &\vdots \\ x_n(t) &= \overset{(n-1)}{y}(t) - \beta_0 \overset{(n-1)}{u}(t) - \beta_1 \overset{(n-2)}{u}(t) - \cdots - \beta_{n-2}\dot{u}(t) - \beta_{n-1}u(t) = \dot{x}_{n-1}(t) - \beta_{n-1}u(t) \end{aligned} \right\} \quad \dots(12)$$

where $\beta_0, \beta_1, \beta_2, \dots, \beta_n$ are determined from

$$\left. \begin{aligned} \beta_0 &= b_0 \\ \beta_1 &= b_1 - a_1\beta_0 \\ \beta_2 &= b_2 - a_1\beta_1 - a_2\beta_0 \\ \beta_3 &= b_3 - a_1\beta_2 - a_2\beta_1 - a_3\beta_0 \\ &\vdots \\ \beta_n &= b_n - a_1\beta_{n-1} - \cdots - a_{n-1}\beta_1 - a_n\beta_0 \end{aligned} \right\} \dots(13)$$

Hence, the state equation and the output equation of state-space method are:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_{n-1}(t) \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} + \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix} [u(t)]$$

$$y(t) = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix} + \beta_0 u(t)$$

Or

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad ...(14)$$

$$y(t) = Cx(t) + Du(t) \qquad ...(15)$$

Where

$$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_{n-1}(t) \\ x_n(t) \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix},$$

$$B = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_{n-1} \\ \beta_n \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix} \quad \textit{and} \quad D = \beta_0 = b_0$$

The initial condition $x(0)$ may be determined by using Eq.(12).

3. Block Method:

Block method provides easy and efficient mean for the solution of the many problems. The concept of block method is essentially an extrapolation procedure and has the advantage of being self-starting. Block method was described for differential equation by Milne and for integral equations was given by Young [11,12].

In this research block method was employed for finding the numerical solution for different types of nth-order state-space equations (SSE) of linear continuous-time control system.

Consider the following first order differential equation :

$y' = f(t, y(t))$ with initial condition

$$y(t_0) = y_0 \qquad \ldots(16)$$

A block method up to the fourth-order for Eq.(16) is computed by:

$$y_{n+1} = y_n + 2hy'_n \qquad \textit{order } 2$$
$$y_{n+2} = y_n + 2hy'_{n+1} \qquad \textit{order } 2$$
$$y_{n+1} = y_n + (h/2)[y'_n + y'_{n+1}] \qquad \textit{order } 3$$
$$y_{n+2} = y_n + (h/2)[y'_n + y'_{n+2}] \qquad \textit{order } 3$$
$$y_{n+1} = y_n + (h/12)[5y'_n + 8y'_{n+1} - y'_{n+2}] \qquad \textit{order } 4$$
$$y_{n+2} = y_n + (h/3)[y'_n + 4y'_{n+1} + y'_{n+2}] \qquad \textit{order } 4$$

Block method of second and third order have been little used for ordinary differential equations, in general, and delay differential equations in particular because they required more evaluation of the function f.

However, the following fourth order block method which is most popular and more efficient for dealing with differential equations.

Let

$$\left.\begin{aligned} B_1 &= f(t_n, y(t_n)) \\ B_2 &= f(t_n + h, y(t_n) + hB_1) \\ B_3 &= f\left(t_n + h, y(t_n) + \frac{h}{2}B_1 + \frac{h}{2}B_2\right) \\ B_4 &= f(t_n + 2h, y(t_n) + 2hB_3) \\ B_5 &= f\left(t_n + h, y(t_n) + \frac{h}{12}(5B_1 + 8B_3 - B_4)\right) \\ B_6 &= f\left(t_n + 2h, y(t_n) + \frac{h}{3}(B_1 + B_4 + 4B_5)\right) \end{aligned}\right\} \ldots(17)$$

Then the fourth order-six steps block method may be written in the form :

$$y_{n+1} = y_n + \frac{h}{12}(5B_1 + 8B_3 - B_4) \ldots(18)$$

$$y_{n+2} = y_n + \frac{h}{3}(B_1 + 4B_5 + B_6) \quad \ldots(19)$$

4. Numerical Solution of State-Space Equations (SSE) of Linear Continuous-Time Systems Using Block Method :

In this section different types of linear state-space equations have been solved using block method.

4.1 The Solution of n^th Order SSE In Which The Forcing Function Dose Not Involve Derivative Terms :

In this subsection the block method including fourth order is candidate to find the numerical solution for the following SSE:

Recall eq.(3) in section (2.1), eq.(3) can be written as:

$$\frac{dx_i(t)}{dt} = f_i(x_1(t), x_2(t), ..., x_n(t), u(t)) \quad ...(20)$$

where $x_1(t), x_2(t), ..., x_n(t)$ are the state variables, $u(t)$ is the input of the system and f_i, i=1,2,…,n denotes the ith linear functional relationship.

The output of the system is obtained from eq.(5) as:

$$y(t) = x_1(t) \quad ...(21)$$

The numerical solution of SSE in eq.(20) and eq.(21) can be found using fourth order block method as follows :

Consider the state equation in eq.(20). By applying block method for eq.(20) by using equations (17), (18) and (19), one gets the following formula :

$$x_i(t_{j+1}) = x_i(t_j) + \frac{h}{12}(5B_{1i} + 8B_{3i} - B_{4i}) \quad ...(22)$$

$$x_i(t_{j+2}) = x_i(t_j) + \frac{h}{3}(B_{1i} + 4B_{5i} + B_{6i}) \quad ...(23)$$

where i=1,2,…,n , j=0,1,…,k and

$$\left.\begin{aligned}
B_{1i} &= f_i(x_1(t_j), x_2(t_j), ..., x_n(t_j), u(t_j)) \\
B_{2i} &= f_i(x_1(t_j) + hB_{11}, x_2(t_j) + hB_{12}, ..., x_n(t_j) + hB_{1n}, u(t_j + h)) \\
B_{3i} &= f_i\left(x_1(t_j) + \frac{h}{2}B_{11} + \frac{h}{2}B_{21}, ..., x_n(t_j) + \frac{h}{2}B_{1n} + \frac{h}{2}B_{2n}, u(t_j + h)\right) \\
B_{4i} &= f_i(x_1(t_j) + 2hB_{31}, ..., x_n(t_j) + 2hB_{3n}, u(t_j + 2h)) \\
B_{5i} &= f_i\left(x_1(t_j) + \frac{h}{12}(5B_{11} + 8B_{31} - B_{41}), ..., x_n(t_j) + \frac{h}{12}(5B_{1n} + 8B_{3n} - B_{4n}), u(t_j + h)\right) \\
B_{6i} &= f_i\left(x_1(t_j) + \frac{h}{3}(B_{11} + B_{41} + 4B_{51}), ..., x_n(t_j) + \frac{h}{3}(B_{1n} + B_{4n} + 4B_{5n}), u(t_j + 2h)\right)
\end{aligned}\right\} \quad ...(24)$$

for each i=1,2,…,n and j=0,1,…, k.

The output values of SSE in eq.(21) can be computed using block method as :

$$y(t_j) = x_1(t_j)$$

where j=0,1,…, k.

The following (SS-SSEB) algorithm summarizes the steps for finding the numerical solution for the **S**I**S**O-**SSE** in eq.(3) using **b**lock method.

SS-SSEB Algorithm :

Step 1: Set $h = \frac{t_k - t_0}{k}$ where (k + 1) is the number of the points $(t_0, t_1, ..., t_k)$ and t_0 is the initial state.

Step 2: Define the state equation in eq.(3).

Step 3: Set j=0

Step 4: For each i=1,2,…,n compute:

$$B_{1i} = f_i(x_1(t_j), x_2(t_j), ..., x_n(t_j), u(t_j))$$

Step 5: $\forall\, i = 1,2,...,n$ compute :

$$B_{2i} = f_i(x_1(t_j) + hB_{11}, x_2(t_j) + hB_{12}, ..., x_n(t_j) + hB_{1n}, u(t_j + h))$$

Step 6: $\forall\, i = 1,2,...,n$ compute :

$$B_{3i} = f_i\left(x_1(t_j) + \frac{h}{2}B_{11} + \frac{h}{2}B_{21}, ..., x_n(t_j) + \frac{h}{2}B_{1n} + \frac{h}{2}B_{2n}, u(t_j + h)\right)$$

Step 7: $\forall\, i = 1,2,...,n$ compute :

$$B_{4i} = f_i(x_1(t_j) + 2hB_{31}, ..., x_n(t_j) + 2hB_{3n}, u(t_j + 2h))$$

Step 8: $\forall\, i = 1,2,...,n$ compute:

$$B_{5i} = f_i\left(x_1(t_j) + \frac{h}{12}(5B_{11} + 8B_{31} - B_{41}), ..., x_n(t_j) + \frac{h}{12}(5B_{1n} + 8B_{3n} - B_{4n}), u(t_j + h)\right)$$

Step 9: $\forall\, i = 1,2,...,n$ compute :

$$B_{6i} = f_i\left(x_1(t_j) + \frac{h}{3}(B_{11} + B_{41} + 4B_{51}), ..., x_n(t_j) + \frac{h}{3}(B_{1n} + B_{4n} + 4B_{5n}), u(t_j + 2h)\right)$$

Step 10: $\forall\, i = 1,2,...,n$ compute :

$$x_i(t_{j+1}) = x_i(t_j) + \frac{h}{12}(5B_{1i} + 8B_{3i} - B_{4i})$$

$$x_i(t_{j+2}) = x_i(t_j) + \frac{h}{3}(B_{1i} + 4B_{5i} + B_{6i})$$

Step 11: Put $j = j+1$

Step 12: If $j = k$ then go to (step 13). Else go to (step 4)

Step 13: For j=0,1,…, k compute the output values of SSE :

$$y(t_j) = x_1(t_j) \quad .$$

4.2 The Solution of n^th Order SSE With (m) Forcing Functions :

The fourth order block method has been used to find the numerical solution for the following MIMO-SSE:

Recall eq.(6) in section (2.2), eq.(6) can be written as:

$$\frac{dx_i(t)}{dt} = f_i(x_1(t), x_2(t),..., x_n(t), u_1(t), u_2(t),..., u_m(t)) \quad \dots(25)$$

where $x_1(t), x_2(t),..., x_n(t)$ are the state variables, $u_1(t), u_2(t),..., u_m(t)$ are the input variables of the system and f_i , i=1,2,…,n denotes the ith linear functional relationship.

The outputs ($y_q(t)$, q =1,2,…,p) of the system in eq.(8) are related to the state variables and the input through the following expression:

$$y_q(t) = g_q(x_1(t), x_2(t),..., x_n(t), u_1(t), u_2(t),..., u_m(t)) \quad \dots(26)$$

where $x_1(t), x_2(t),..., x_n(t)$ are the state variables, $u_1(t), u_2(t),..., u_m(t)$ are the input variables of the system and g_q , q=1,2,…,p denotes the qth linear functional relationship.

The numerical solution of SSE in eq.(25) and eq.(26) can be found using fourth order block method as follows :

Consider the state equation in eq.(25). By applying block method for eq.(25) by using equations (17), (18) and (19), one gets the following formula :

$$x_i(t_{j+1}) = x_i(t_j) + \frac{h}{12}(5B_{1i} + 8B_{3i} - B_{4i}) \quad \dots(27)$$

$$x_i(t_{j+2}) = x_i(t_j) + \frac{h}{3}(B_{1i} + 4B_{5i} + B_{6i}) \quad \dots(28)$$

where i=1,2,…,n , j=0,1,…, k and

$$\left.\begin{aligned}
B_{1i} &= f_i(x_1(t_j), x_2(t_j),..., x_n(t_j), u_1(t_j), u_2(t_j),..., u_m(t_j)) \\
B_{2i} &= f_i(x_1(t_j) + hB_{11}, x_2(t_j) + hB_{12},..., x_n(t_j) + hB_{1n}, u_1(t_j + h),..., u_m(t_j + h)) \\
B_{3i} &= f_i\left(x_1(t_j) + \frac{h}{2}B_{11} + \frac{h}{2}B_{21},..., x_n(t_j) + \frac{h}{2}B_{1n} + \frac{h}{2}B_{2n}, u_1(t_j + h),...u_m(t_j + h)\right) \\
B_{4i} &= f_i(x_1(t_j) + 2hB_{31},..., x_n(t_j) + 2hB_{3n}, u_1(t_j + 2h),..., u_m(t_j + 2h)) \\
B_{5i} &= f_i\left(x_1(t_j) + \frac{h}{12}(5B_{11} + 8B_{31} - B_{41}),..., x_n(t_j) + \frac{h}{12}(5B_{1n} + 8B_{3n} - B_{4n}), u_1(t_j + h),..., u_m(t_j + h)\right) \\
B_{6i} &= f_i\left(x_1(t_j) + \frac{h}{3}(B_{11} + B_{41} + 4B_{51}),..., x_n(t_j) + \frac{h}{3}(B_{1n} + B_{4n} + 4B_{5n}), u_1(t_j + 2h),..., u_m(t_j + 2h)\right)
\end{aligned}\right\} \quad \dots(29)$$

for each i=1,2,…,n and j=0,1,…, k .

The outputs of SSE in eq.(26) can be computed using block method as :

$$y_q(t_j) = g_q(x_1(t_j), x_2(t_j),..., x_n(t_j), u_1(t_j), u_2(t_j),..., u_m(t_j)) \quad \dots(30)$$

where q=1,2,…,p and j=0,1,…, k .

The following (MM-SSEB) algorithm summarizes the steps for finding the numerical solution for the <u>**M**</u>I<u>**M**</u>O-<u>**SSE**</u> in eq.(6) and eq.(8) using <u>**b**</u>lock method.

<u>MM-SSEB Algorithm:</u>

Step 1: Set $h = \frac{t_k - t_0}{k}$ where (k + 1) is the number of the points $(t_0, t_1,..., t_k)$ and t_0 is the initial state.

Step 2: Find the state equation as eq.(6) and the output equation as eq.(8).
Step 3: Set j=0
Step 4: For each i=1,2,…,n compute:

$$B_{1i} = f_i(x_1(t_j), x_2(t_j),..., x_n(t_j), u_1(t_j),..., u_m(t_j))$$

Step 5: $\forall\ i = 1,2,...,n$ compute :

$$B_{2i} = f_i(x_1(t_j) + hB_{11}, x_2(t_j) + hB_{12},..., x_n(t_j) + hB_{1n}, u_1(t_j + h),...u_m(t_j + h))$$

Step 6: $\forall\ i = 1,2,...,n$ compute :

$$B_{3i} = f_i\left(x_1(t_j) + \frac{h}{2}B_{11} + \frac{h}{2}B_{21},..., x_n(t_j) + \frac{h}{2}B_{1n} + \frac{h}{2}B_{2n}, u_1(t_j + h),..., u_m(t_j + h)\right)$$

Step 7: $\forall\ i = 1,2,...,n$ compute :

$$B_{4i} = f_i(x_1(t_j) + 2hB_{31},..., x_n(t_j) + 2hB_{3n}, u_1(t_j + 2h),..., u_m(t_j + 2h))$$

Step 8: $\forall\ i = 1,2,...,n$ compute :

$$B_{5i} = f_i\left(x_1(t_j) + \frac{h}{12}(5B_{11} + 8B_{31} - B_{41}),..., x_n(t_j) + \frac{h}{12}(5B_{1n} + 8B_{3n} - B_{4n}), u_1(t_j + h),..., u_m(t_j + h)\right)$$

Step 9: $\forall\ i = 1,2,...,n$ compute :

$$B_{6i} = f_i\left(x_1(t_j) + \frac{h}{3}(B_{11} + B_{41} + 4B_{51}),..., x_n(t_j) + \frac{h}{3}(B_{1n} + B_{4n} + 4B_{5n}), u_1(t_j + 2h),..., u_m(t_j + 2h)\right)$$

Step 10: $\forall\ i = 1,2,...,n$ compute :

$$x_i(t_{j+1}) = x_i(t_j) + \frac{h}{12}(5B_{1i} + 8B_{3i} - B_{4i})$$

$$x_i(t_{j+2}) = x_i(t_j) + \frac{h}{3}(B_{1i} + 4B_{5i} + B_{6i})$$

Step 11: $\forall\ q = 1,2,..., p$ compute the output values of MIMO-SSE :

$$y_q(t_j) = g_q(x_1(t_j), x_2(t_j),..., x_n(t_j), u_1(t_j), u_2(t_j),..., u_m(t_j))$$

.

Step 12: Put $j = j+1$
Step 12: If $j = k$ then stop.
Else go to (step 4)

4.3 The Solution of nth Order SSE In Which The Forcing Function Involves Derivative Terms :

The fourth order block method has been used to find the numerical solution for the following SSE:

Recall eq.(14) in section (2.3), eq.(14) can be written as:

$$\frac{dx_i(t)}{dt} = f_i(x_1(t), x_2(t),..., x_n(t), \beta_i\, u(t)) \quad \text{…(31)}$$

where $x_1(t), x_2(t),..., x_n(t)$ are the state variables, $u(t)$ are the input variable of the system, β_i in eq.(13) and f_i , i=1,2,…,n denotes the ith linear functional relationship.

The output of the system is obtained from eq.(15) as:

$$y(t) = x_1(t) + \beta_0\, u(t) \quad \text{…(32)}$$

The numerical solution of SSE in eq.(31) and eq.(32) can be found using fourth order block method as prescribed in section (4.1).

5. Numerical Examples:

The previous methods in section (4) are illustrated in the following examples:-

Example (1) :

In the Cruise Control Problem [10], the state-space model was derived as:

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & -0.05 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0.001 \end{bmatrix} u(t)$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + [0]u(t)$$

where the initial state is: $x(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and the forcing function $u(t) = e^t$, $t \geq 0$.

The exact solution of the above SISO state-space model is:

$$x(t) = \begin{bmatrix} exact_1 \\ exact_2 \end{bmatrix} = \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} =$$

$$\begin{bmatrix} \frac{-2098}{105} e^{-\frac{1}{20}t} + \frac{999}{50} + \frac{1}{1050} e^t \\ \frac{1049}{1050} e^{-\frac{1}{20}t} + \frac{1}{1050} e^t \end{bmatrix}$$

When the algorithm (SS-SSEB) is applied, table (1) presents the comparison between the exact and numerical solution using block method for k=10, h=0.1 and $t_i = ih$, $i = 0,1,..., k$ depending on least square error (L.S.E.). The output variables y(t) of state space model by applying (SS-SSEB) algorithm is also tabulated.

Table (1) The solution $x(t)$ and the output variables$y$(t) of state space model for Ex.(1).

t	$Exact_1$	Block x_1(t)	$Exact_2$	Block x_2(t)	output y(t)	Block y(t)
0	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000
0.1	0.0998	0.0998	0.9951	0.9951	0.0998	0.0998
0.2	0.1990	0.1990	0.9903	0.9903	0.1990	0.1990
0.3	0.2978	0.2978	0.9855	0.9855	0.2978	0.2978
0.4	0.3961	0.3961	0.9807	0.9807	0.3961	0.3961
0.5	0.4939	0.4939	0.9760	0.9760	0.4939	0.4939
0.6	0.5913	0.5913	0.9713	0.9713	0.5913	0.5913
0.7	0.6882	0.6882	0.9666	0.9666	0.6882	0.6882
0.8	0.7846	0.7846	0.9620	0.9620	0.7846	0.7846
0.9	0.8806	0.8806	0.9574	0.9574	0.8806	0.8806
1	0.9761	0.9761	0.9529	0.9529	0.9761	0.9761
L.S.E.		0.145e-13	L.S.E.	0.158e-13	L.S.E.	0.145e-13

Example (2) :

Consider the MIMO control system shown in figure (4) :

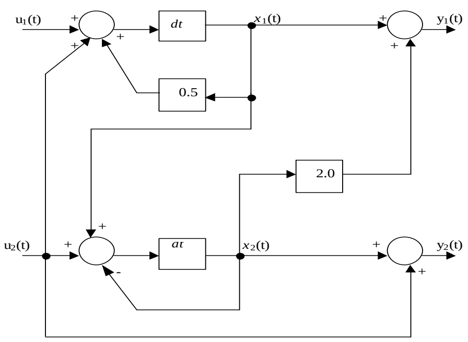

Fig.(4) Simulation diagram for a multivariable system.

The MIMO state-space model was derived from fig.(4) as follows :

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

The initial state of the MIMO state-space model is: $x(0) = \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and the forcing function $u(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} = \begin{bmatrix} 1 \\ t \end{bmatrix}$, $t \geq 0$.

The exact solution of the above MIMO state-space model is:

$$x(t) = \begin{bmatrix} exact_1 \\ exact_2 \end{bmatrix} = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} -6 - 2t + 6e^{\frac{1}{2}t} \\ -5 - t + e^{-t} + 4e^{\frac{1}{2}t} \end{bmatrix}$$

When the algorithm (MM-SSEB) is applied, table (2) presents the comparison between the exact and numerical solution using block method for k=10, h=0.1 and $t_i = ih$, $i = 0,1,...,k$ depending on least square error (L.S.E.). The output variables y(t) of state space model by applying (MM-SSEB) algorithm is also tabulated.

Table (2) The solution $x(t)$ and the output variables y(t) of state space model for Ex.(2).

t	*$Exact_1$*	*Block* x_1(t)	*$Exact_2$*	*Block* x_2(t)	*Output* y_1(t)	*Block* y_1(t)	*Output* y_2(t)	*Block* y_2(t)
0	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.1	0.1076	0.1076	0.0099	0.0099	0.1275	0.1274	0.1099	0.1099
0.2	0.2310	0.2310	0.0394	0.0394	0.3099	0.3098	0.2394	0.2394
0.3	0.3710	0.3710	0.0882	0.0881	0.5473	0.5472	0.3882	0.3881
0.4	0.5284	0.5284	0.1559	0.1559	0.8403	0.8402	0.5559	0.5559
0.5	0.7042	0.7041	0.2426	0.2426	1.1894	1.1893	0.7426	0.7426
0.6	0.8992	0.8991	0.3482	0.3482	1.5956	1.5955	0.9482	0.9482
0.7	1.1144	1.1144	0.4729	0.4728	2.0601	2.0609	1.1729	1.1728
0.8	1.3509	1.3509	0.6166	0.6165	2.5842	2.5840	1.4166	1.4165
0.9	1.6099	1.6098	0.7798	0.7797	3.1695	3.1693	1.6798	1.6797
1	1.8923	1.8923	0.9628	0.9627	3.8179	3.8176	1.9628	1.9627
L.S.E.		0.153e-7	L.S.E.	0.419e-7	L.S.E.	0.283e-6	L.S.E.	0.419e-7

Example (3) :

Consider the following SISO control system equation for the Pitch controller [9]:

$$\dddot{x}+6\ddot{x}+11\dot{x}+6x=\dddot{u}+8\ddot{u}+17\dot{u}+8u$$

with initial conditions: $y(0)=0,\ \dot{y}(0)=1,\ \ddot{y}(0)=0$ and the forcing function $u(t)=2t^3,\ t\geq 0$.

The state-space equation was derived using eq.(14) and eq.(15) as follows :

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\\ \dot{x}_3(t)\end{bmatrix}=\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -6 & -11 & -6\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}+\begin{bmatrix}2\\ -6\\ 16\end{bmatrix}[u(t)]$$

$$y(t)=\begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}+u(t)$$

where the initial state of the state-space model is: $x(0)=\begin{bmatrix}x_1(0)\\ x_2(0)\\ x_3(0)\end{bmatrix}=\begin{bmatrix}0\\ 1\\ 0\end{bmatrix}$ and the forcing function $u(t)=2t^3,\ t\geq 0$.

The exact solution of the above SISO state-space model is:

$$x(t)=\begin{bmatrix}exact_1\\ exact_2\\ exact_3\end{bmatrix}=\begin{bmatrix}x_1(t)\\ x_2(t)\\ x_3(t)\end{bmatrix}=$$

$$\begin{bmatrix}-\frac{5}{2}e^{-2t}+\frac{89}{54}e^{-3t}-\frac{19}{2}e^{-t}+\frac{2}{3}t^3+\frac{7}{3}t^2-\frac{77}{9}t+\frac{559}{54}\\ 5e^{-2t}-\frac{89}{18}e^{-3t}+\frac{19}{2}e^{-t}-4t^3+2t^2+\frac{14}{3}t-\frac{77}{9}\\ -10e^{-2t}+\frac{89}{6}e^{-3t}-\frac{19}{2}e^{-t}+12t^3-12t^2+4t+\frac{14}{3}\end{bmatrix}$$

When the algorithm (SS-SSEB) is applied, table (3) presents the comparison between the exact and numerical solution using block method for k=10, h=0.01 and $t_i=ih$, $i=0,1,...,k$ depending on least square error (L.S.E.). The output variables y(t) of state space model by applying (SS-SSEB) algorithm is also tabulated.

Table (3) The solution $x(t)$ and the output variables y(t) of state space model for Ex.(3).

t	*Exact₁*	*Block* x_1(t)	*Exact₂*	*Block* x_2(t)	*Exact₃*	*Block* x_3(t)	*Output* y(t)	*Block* y(t)
0	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.01	0.0100	0.0100	0.9995	0.9995	-0.1070	-0.1070	0.0100	0.0100
0.02	0.0200	0.0200	0.9979	0.9979	-0.2083	-0.2083	0.0200	0.0200
0.03	0.0300	0.0300	0.9953	0.9953	-0.3040	-0.3040	0.0300	0.0300
0.04	0.0399	0.0399	0.9918	0.9918	-0.3944	-0.3944	0.0400	0.0400
0.05	0.0498	0.0498	0.9874	0.9874	-0.4797	-0.4797	0.0500	0.0500
0.06	0.0596	0.0596	0.9822	0.9822	-0.5601	-0.5601	0.0601	0.0601
0.07	0.0695	0.0695	0.9762	0.9762	-0.6357	-0.6357	0.0701	0.0701
0.08	0.0792	0.0792	0.9694	0.9694	-0.7067	-0.7067	0.0802	0.0802
0.09	0.0889	0.0889	0.9619	0.9619	-0.7734	-0.7734	0.0903	0.0903
0.1	0.0985	0.0985	0.9538	0.9538	-0.8358	-0.8358	0.1005	0.1005
L.S.E.		0.159e-11	L.S.E.	0.249e-10	L.S.E.	0.299e-9	L.S.E.	0.159e-11

6. Conclusion:

Block method has been presented to find the numerical solution for different types of nth-order state-space equations (SSE) of linear continuous-time control system. The results show a marked improvement in the least square errors (L.S.E.). From solving some numerical examples the following points are included:

1- Block method solves the SSE of the SISO system as well as MIMO system.

2- Block method gives a better accuracy and consistent to the solution of different types of nth-order state-space equations.

3- The good approximation depends on the size of h, if h is decreased then the number of points (knots) increases and the L.S.E. approaches zero where this gives the advantage in numerical computation.

7. References:

1. Ogata, Katsuhiko, 1997"Modern Control Engineering", Third Edition, Printed in the New Jersey.

2. Fairman , Frederick W. 1989 , Linear Control Theory and The State Space Approach, John Wiley & Sons , Inc., USA 1998.

3. Vladimir, Strejc , State Space Theory of Discrete Linear Control , Printed in Czechoslovak Academy of Sciences,.

4. Amir, Karniel and Gideon F. 2004 Inbar, "Linear System Description", www.visl.technion.ac.il/karniel/pub/karnielInbar/ModernTechNs/ch21.pdf.,.

5. Lewis,F. L., 2000, State Variable Systems, www.cis.hut.fi/EE4314-controlsystems/lecture1.pdf. ,.

6. Masanao, Aoki , 1987, State Space Modeling of Time Series, Printed in Germany, Springer-Verlag.

7. Phillips ,Charles L. and Nagle ,H. Troy , 1992, Digital Control System Analysis and Design , Englewood Cliffs, N. J. ,Prentice Hall, Inc.,.

8. Wiberg, Donald M. ., 1981 , State Space and Linear Systems, McGraw-Hill ,Inc. in USA.

9. John, J., 2000, State Space Design for The Pitch Controller, www.krellinst.org/UCES/archive/classes/control/examples/pitch/invSS. htm1 - 14k ,.

10. James, Dk. , 2001, State Space Control of The Cruise Control Model , www.krellinst.org/UCES/archive/classes/control/examples/cruise/ccSS.html-7k .

11. Lambert, J.D., 1979, "Computational Methods in Ordinary Differential Equations", John Wiley & Sons Ltd., New York,.

12. Shampin ,L.and Watts H., , 2002," The Stability of The Block Methods", J. Number Anal.13:101-114.

طريقة البلوك لحل معادلات فضاء الحالة لأنظمة السيطرة الخطية المستمرة الزمن

رغد كاظم صالح*　　شيماء حسين صالح*　　أثير جواد كاظم*

*قسم العلوم التطبيقية الجامعة التكنولوجية

الخلاصة:

يقدم البحث طريقة مطورة مع خوارزميات جديدة لإيجاد الحل العددي لمعادلات فضاء الحالة الخطية المستمرة الزمن لأنظمة السيطرة باستخدام طريقة البلوك. حيث تمت معالجة أنظمة فضاء الحالة عددياً للمنظومات الفردية المدخل والمخرج مثلما للمنظومات المتعددة المداخل والمخارج باستخدام طريقة البلوك من الرتبة الرابعة. بالإضافة إلى ذلك تم أيجاد النتائج العددية لمعادلة الإخراج لتمثيل فضاء الحالة. استخدمت لغة (Matlab) لبرمجة هذه الطريقة. كما تمت مقارنة النتائج العددية و الحقيقية لأنواع مختلفة من معادلات فضاء الحالة من خلال بعض الأمثلة وقد تم الحصول على نتائج جيدة.