IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE)
e-ISSN: 2278-1676,p-ISSN: 2320-3331, Volume 11, Issue 5 Ver. III (Sep - Oct 2016), PP 92-96
www.iosrjournals.org

Akpan's Algorithm to Determine State Transition Matrix and Solution to Differential Equations with Mixed Initial and Boundary Conditions

Akpan, E. A.
Department of Electrical/Electronic/Computer Engineering, University of UYO, Uyo, Nigeria

Abstract: *Solving a system of linear constant coefficient differential equations with mixed initial and boundary values without the use of a computer is not a trivial endeavor if the order of the system is higher than two. This article delineates algorithms for computing the state transition matrix (STM) and the solution of systems of constant coefficient linear differential equations of any order. These algorithms obviate the shortcomings inherent in Leverrier's algorithm, Sylvester's expansion theorem, Cayley-Hamilton's theorem, and Putzer's algorithm. Furthermore, these algorithms do not require symbolic software since the STM and differential equation solution can be computed using regular Matlab or C++.*
Keywords: *Algorithm, state transition matrix, constant coefficient, differential equations*

I. Introduction

A model of a system of constant coefficient linear ordinary differential equations is given as

$$\dot{\mathbf{x}}(\mathrm{t}) = \mathbf{A}\mathbf{x}(\mathrm{t}) \tag{1}$$

with initial condition $x(t_0) = x_0 \neq 0$, where, $x(\mathrm{t})$ is a column vector with n-entries and each entry is a real number; and A is a constant $n \times n$ matrix with real or complex entries. In this article the state transition matrix (STM) is denoted by $\Phi(t)$. Since the STM is an important intermediate result in the solution of eqn. (1), several researchers have proposed algorithms and theorems to compute the STM. Notable among them are the Leverrier's algorithm [1], Sylvester's theorem [2], Cayley-Hamilton's theorem [3], and Putzer's algorithm [4].

In Leverrier's algorithm, a resolvent matrix $\Phi(s)$ is found for eqn. (1), expressed as

$$\Phi(s) = (sI - A)^{-1} \tag{2}$$

Then the STM of the system is the inverse Laplace transform of $\Phi(s)$. If n is large, taking the inverse Laplace transform of $\Phi(s)$ requires symbolic software. Thus, the Leverrier's algorithm does not yield the STM directly.

Sylvester's expansion theorem can be used to compute the STM for the case where the matrix A has distinct eigenvalues as follows.

$$\Phi(t) = \sum_{i=1}^{n} e^{\lambda_i t} F_i \tag{3}$$

where, λ_i, $i = 1,2,...,n$, are the eigenvalues of the matrix *A*, and

$$F_i = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{A - \lambda_j I}{\lambda_i - \lambda_j}$$

While F_i can be computed numerically, this algorithm is numerically unstable because it involves division by the difference of eigenvalues which may approach zero in some cases.

Cayley-Hamilton's theorem can be adapted to find the STM for the case of real and distinct eigenvalues. In this case the system of equations to solve for the vector $\alpha(t) = (\alpha_1(t) \quad \alpha_2(t) \quad \cdots \quad \alpha_n(t))$ is:

$$e^{\lambda_i t} = \alpha_0 + \alpha_1 \lambda_i + \alpha_2 \lambda_i^2 + \cdots + \alpha_{n-1} \lambda_i^{n-1}, \; i = 1,2,...,n$$

Then,

$$\Phi(t) = \alpha_0 I + \alpha_1 A_i + \alpha_2 A^2 + \cdots + \alpha_{n-1} A^{n-1} \tag{4}$$

The drawback of this approach is that finding α is not amenable to numerical method.
Finally, the Putzer algorithm can be used to determine the STM even for the case of repeated eigenvalues as follows.

$$\Phi(t) = \sum_{j=0}^{n-1} r_{j+1}(t) P_j \tag{5}$$

where, $P_0 = I$, and $P_j = \prod_{k=1}^{j}(A - \lambda_k I)$, $j = 1,2,...,n$

The vector $r(t) = (r_1(t) \quad r_2(t) \quad \cdots \quad r_n(t))$ is a solution of a separate system of ordinary differential equations from eqn. (1); and $r(t)$ is found manually or by using software capable of symbolic manipulations.
Eddie Akpan's algorithm, introduced in this article, is easily implemented by simple computer programs in Matlab, C++, and other computing software; and does not have the drawbacks inherent in the other algorithms, such as numerical instability and a need to possess expensive symbolic software.
This article is organized as follows: Algorithms for computing the state transition matrix for initial and mixed initial and boundary value problems are given in section 2. An algorithm for computing the solution of systems of differential equations as a function of eigenvalue exponentials is given in section 3. An example that demonstrates the application of the proposed algorithms appears in section 4. Section 5 is the conclusion.

1.1 Eddie Akpan's Algorithm for Finding the State Transition Matrix for Initial Value Problems

Let $\Phi(t)$ denote the state transition matrix (STM) of the autonomous linear system given in eqn. (1). Suppose the matrix *A* has distinct real or complex eigenvalues α_1, α_2, …, α_n. Define a vector of eigenvalues exponential as

$$E_\alpha(t) = \left(e^{\alpha_1 t} \quad e^{\alpha_2 t} \quad \cdots \quad e^{\alpha_n t}\right)^T .$$

Further, let P be the matrix whose columns are the eigenvectors of *A*. Additionally, let the vector m_{jk}, for fixed *j* and *k*, be given as

$$m_{jk} = \left\lfloor P_{ji} P_{ik}^{-1} \right\rfloor , \; i = 1,2,...,n \tag{6}$$

and,

$$M = \left\lfloor m_{jk} \right\rfloor , \; j,k = 1,2,...,n .$$

Then Eddie Akpan's formula for computing the STM is given

as

$$\Phi(t) = \left\lfloor m_{jk} E_\alpha(t) \right\rfloor , \; j,k = 1,2,...,n . \tag{7}$$

Taking $E_\alpha(t)$ outside the bracket on the right hand side of eqn. (7), the state transition matrix is written as

$$\Phi(t) = M \bullet E_\alpha(t) \tag{8}$$

where, the symbol "$\bullet$" implies element-by-element dot product of m_{jk} (the entries of *M*) with $E_\alpha(t)$.

1.2 Eddie Akpan's Algorithm for Determining the State Transition Matrix for Mixed Initial and Boundary Value Problems

In some applications, such as optimization problems, the optimal solutions can occur at the boundaries of the domain of solutions. When this happens, methods of solution of initial value problems are inadequate because mixed initial and boundary values are involved. These types of problems necessitate the use of Lagrange multipliers [2]. Consequently, for an n^{th}-order system, the size of the system of equations to solve is $q = 2n$.

Let $x(\mathrm{t})$ be the n-dimensional state vector for the system in eqn. (1), and $z(\mathrm{t})$ is a vector of Lagrange multipliers with the same dimension as $x(\mathrm{t})$. Then the system with mixed initial and boundary conditions is written as

$$\begin{bmatrix} \dot{\mathbf{x}}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A_1 & C_1 \\ C_2 & A_2 \end{bmatrix} \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} \tag{9}$$

with the accompanying initial and boundary conditions $x(t_0) = x_0 \neq 0$ and $x(t_f) = x_f \neq 0$, respectively. The matrices A_1 and A_2 are constant $n \times n$ matrices with real or complex entries; and the matrices C_1 and C_2 are coupling terms. Let

$$A = \begin{bmatrix} A_1 & C_1 \\ C_2 & A_2 \end{bmatrix} \quad \text{and} \quad \eta(t) = \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}.$$

Then the system of equations with mixed initial and boundary conditions is given by

$$\dot{\eta}(t) = A\eta(t) \tag{10}$$

In eqn. (10) the size of the A-matrix is $q \times q$. The STM for this case is

$$\Phi(t) = \lfloor m_{jk} E_\alpha(t) \rfloor, \; j,k = 1,2,...,q$$

(11) where,

$$m_{jk} = \lfloor P_{ji} P_{ik}^{-1} \rfloor, \; i = 1,2,...,q \;, \tag{12}$$

and $E_\alpha(t) = \begin{pmatrix} e^{\alpha_1 t} & e^{\alpha_2 t} & \cdots & e^{\alpha_q t} \end{pmatrix}^T$.

Eddie Akpan's Algorithm for Determining the Solution of Systems of Linear Differential Equations with Initial and Mixed Initial and Boundary Values

Let a vector C be given as

$$C = \left(C(i)\right)^T \tag{13}$$

where, for initial value problems, $i = 1,2,...,n$, and for mixed initial and boundary value problems, $i = 1,2,...,q$. For the initial value problem eqn. (1), the solution of the system of equations is given by

$$x(t) = \Phi(t)C \tag{14}$$

The vector C is a function of initial conditions, and satisfies

$$C = \Phi(t_0)^{-1} x(t_0) \tag{15}$$

If $t_0 = 0$,

$$x(t) = \Phi(t)x_0 \tag{16}$$

For mixed initial and boundary value problems, $\Phi(t)$ is a $q \times q$ matrix, and t_0 and t_f, respectively, represent the initial and final time. In that case, let $\tilde{\Phi}(t)$ denote the first n-rows of $\Phi(t)$. The vector C satisfies

$$C = \begin{bmatrix} \tilde{\Phi}(t_0) \\ \tilde{\Phi}(t_f) \end{bmatrix}^{-1} \begin{bmatrix} x(t_0) \\ x(t_f) \end{bmatrix} \tag{17}$$

Then the solution of (10) is

$$\eta(t) = \Phi(t)C \tag{18}$$

with the vector *C* given by eqn. (17).

Although the solutions of eqns. (1) and (10) can be written in the form of eqns. (14) and (18), it is preferable to express $x(t)$ andr $\eta(t)$ succinctly as functions of the eigenvalues exponential [5]. Eddie Akpan's formula for the solution of systems of constant coefficient linear ordinary differential equations is as follows:
For fixed *j* and *k*, let

$$w_{jk} = \sum_{i=1}^{q} C(i) m_{ji}(k) \tag{19}$$

where, m_{ji} is given as eqn. (6) or (12), and C is specified as eqn. (15) or (17). Also, let

$$W = \lfloor w_{jk} \rfloor, \quad j,k = 1,2,\ldots,q \tag{20}$$

Then, $\eta(t)$, or $x(t)$ in the case of initial value problems, is

$$\eta(\mathrm{t}) = W \times E_{\alpha}(\mathrm{t})$$
(21)

Demonstration Example for a Mixed Initial and Boundary Values Problem

For an example that demonstrates the application of the algorithms enumerated in this article, consider a scalar system given in [2]:

$$\dot{\mathbf{x}}(\mathrm{t}) = -\mathbf{x}(\mathrm{t}) - 0.5z(t) \tag{22}$$

The initial and boundary values are $x(0) = 1$ *and* $x(1) = 0$, respectively. The variable $z(t)$ is a Lagrange multiplier having the dynamics

$$\dot{z}(\mathrm{t}) = -2\mathbf{x}(\mathrm{t}) + z(t) \tag{23}$$

Hence, the system of equations to solve is:

$$\dot{\eta}(t) = \begin{bmatrix} -1 & -0.5 \\ -2 & 1 \end{bmatrix} \eta(t) \tag{24}$$

The eigenvalues of the A-matrrix are $\pm\sqrt{2}$. Therefore,

$$E_{\alpha}(t) = \left(e^{-\sqrt{2}t} \quad e^{\sqrt{2}t}\right)^{T} \tag{25}$$

$P = \begin{bmatrix} -0.7701 & 0.2028 \\ -0.6380 & -0.9792 \end{bmatrix}$, and $P^{-1} = \begin{bmatrix} -1.1084 & -0.2296 \\ 0.7221 & -0.8717 \end{bmatrix}$.

$M = \lfloor m_{jk} \rfloor$, *j*, *k* = 1,2; so by eqn. (6), $m_{11}(1) = -0.7701 \times -1.1084 = 0.8536$, and $m_{11}(2) = 0.2028 \times 0.7221 = 0.1464$, thus, $m_{11} = [0.8536 \quad 0.1464]$; and m_{12}, m_{21} and m_{22} are computed similarly. Hence,

$$M = \begin{bmatrix} [0.8536 \quad 0.1464] & [0.1768 \quad -0.1768] \\ [0.7071 \quad -0.7071] & [0.1464 \quad 0.8536] \end{bmatrix}.$$

The STM is:

$$\Phi(t) = \begin{bmatrix} 0.8536e^{-\sqrt{2}t} + 0.1464e^{\sqrt{2}t} & 0.1768(e^{-\sqrt{2}t} - e^{\sqrt{2}t}) \\ 0.7071(e^{-\sqrt{2}t} - e^{\sqrt{2}t}) & 0.1464e^{-\sqrt{2}t} + 0.8536e^{\sqrt{2}t} \end{bmatrix}.$$

Using the initial and boundary conditions,

$\tilde{\Phi}(t_0) = [1 \quad 0]$ and $\tilde{\Phi}(t_f) = [0.8097 \quad -0.6842]$.

$$\Rightarrow C = \begin{bmatrix} 1 & 0 \\ 0.8097 & -0.6842 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1.1854 \end{bmatrix}.$$

By eqn. (19), $w_{11} = C(1) \times m_{11}(1) + C(2) \times m_{12}(1) = 1.0632$; w_{12}, w_{21} and w_{22} are computed similarly. Consequently,

$$\begin{bmatrix} x(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} 1.0632 \\ 0.8806 \end{bmatrix} e^{-\sqrt{2}t} - \begin{bmatrix} 0.0632 \\ -0.3048 \end{bmatrix} e^{\sqrt{2}t}.$$

II. Conclusion

Efficient algorithms for computing the state transition matrix (STM) and solution of systems of constant coefficient linear differential equations with initial or mixed initial and boundary values have been presented. The STM algorithm utilizes the eigenvalues and eigenvectors of the system; and does not have the numerical instability problem or need for symbolic software which is the case with existing algorithms, such as the Leverrier's algorithm, Sylvester's expansion theorem, Cayley-Hamilton's theorem, and Putzer's algorithm. The solution of the systems of equations is expressed in terms of the eigenvalues exponential. These algorithms can be implemented using regular computing software such as Matlab or C++. Finally, a mixed initial and boundary value example that highlights the use of these algorithms has been exhibited.

References

[1]. Kailath, T., Linear Systems, Prentice Hall, New Jersey, 1980.
[2]. Schultz D. G. and Melsa, J. L., State Functions and Linear Control Systems, McGraw-Hill, New York, 1967.
[3]. Jacob B., Linear Algebra, W. H. Freeman and Company, New York, 1990.
[4]. Waltman P., A Second Course in Elementary Differential Equations, Academic Press, Orlando, 1986.
[5]. Braun M., Differential Equations and Their Applications, 4th edn., Springer-Verlag, New York, 1993.