Journal of the
Nigerian Mathematical Society

Vol. 37, Issue 3, pp. 175-201, 2018
©Nigerian Mathematical Society

AN EXPONENTIAL METHOD OF VARIABLE ORDER FOR GENERAL NONLINEAR (STIFF AND NONSTIFF) ODE SYSTEMS

C. C. JIBUNOH

ABSTRACT. In this paper, an explicit method, hereby called an Exponential Method of variable order, is derived from the earlier published Exponential Method of orders 2 and 3. The present method of variable order commands higher accuracy since it obtains numerical solutions which coincide with the exact theoretical solutions, to eight or more decimal places, in virtually all stiff and nonstiff, (linear and nonlinear) ODE systems. Numerical applications show that it has faster convergence and much higher accuracy than many existing methods. New formats are now introduced to make it easy to integrate any $K \times K$ systems. Other remarkable features include the use of the exact Jacobians of nonlinear systems; implementation of a phase to phase integration of stiff systems, with exact formulas for determining the terminal points of phases; avoidance of matrix inversions, LU decompositions and the cumbersome Newton iterations, since the method is explicit; solving oscillatory systems without additional refinements and a straight forward application of the method without starters. Implementations show that any program of the Exponential Method of variable order (e.g the QBASIC program) produces a very fast or instant output in automatic computation.

Keywords and phrases: Jibunoh correction for continuity, Transient eigenvalues, Exact Jacobian, Component by component integration, Diagonal Z variables, Two phase integration, QBASIC Codes.
2010 Mathematical Subject Classification: 14H40,15A18,65D30

1. INTRODUCTION

An Exponential Method (EM) of a maximum of order 3, was developed in Jibunoh [2] for the accurate and automatic integration of any nonlinear Ordinary Differential System (stiff or nonstiff), represented generally by the IVP;

$$y' = f(t, y), y(t_0) = y_0 \tag{1}$$

Received by the editors August 24, 2017; Revised: October 28, 2018; Accepted: November 16, 2018
www.nigerianmathematicalsociety.org; Journal available online at www.ojs.ictp.it

Although the EM gave satisfactory results in [2], its extension to higher orders e.g to order $p > 3$, was hindered by the labour of deriving successively higher powers of the general Jacobian, in the traditional way.

Previous works on nonlinear ODE systems had centred more on the applications of the fourth order RK method [3], Adomian decomposition method [6], including that of Rach, Adomian and Meyers [7]. Also in vogue was the spline collocation method, using integral equation reformulations [4], which was closely followed by the Differential Transform Method (DTM) in [5], for linear systems. Most of these methods were methods of fixed orders and their accuracies should need improvement.

A higher order method is expected to produce higher accuracy and if the order p is a variable, the adjustments of order will be straight forward during the integration of various forms of nonlinear ODE systems.

In this paper, we shall develop an improved version of the Exponential Method, given in Jibunoh [2], which will be of variable order and shall be applicable to the general nonlinear (stiff and nonstiff) ODE systems, with simplicity of computations. The mathematical simplicity of the method and its high accuracy will be evident from its construction and implementations.

The method may, perhaps, be more attractive than the recent works on Second Derivative Runge-Kutta Methods (SD-RKM) or the Second Derivative General Linear Methods (SD-GLM) such as those of [12] and [14], which are currently appearing in the literature, with tedious derivations. The EM does not employ any starter methods.

Our first task in constructing the Exponential Method of variable order for the system (1), is to find a simplified way of generating successively the m^{th} powers $(m \leq p-1)$ of the general Jacobian.

2. FORMULAS FOR THE POWERS OF THE GENERAL JACOBIAN A_n, IN TERMS OF THE ENTRIES

As proved in [2], the general offshoot of Jibunoh Spectral Decomposition [1] of order p, where $p \geq 1$ is an integer, can be deduced as

$$y_{n+1} = y_n + (hI + \frac{h^2}{2}A_n + \frac{h^3}{6}A_n^2 + \ldots + \frac{h^p}{p!}A_n^{p-1})f_n \tag{2}$$

where A_n is the general Jacobian at step n. This is an explicit one step method of order p which is an exponential function of the exact

Jacobian of the $K \times K$ system. The form (2) is now defined as the exponential method of variable order (EM) since p is a variable. We shall first demonstrate the procedure of obtaining the powers of A_n for any $K \times K$ system. Let

$$\begin{array}{ll} y_1' = f_1 = f_1(t, y_1, y_2, ..., y_k), & y_1(t_0) = y_{10} \\ y_2' = f_2 = f_2(t, y_1, y_2, ..., y_k), & y_2(t_0) = y_{20} \\ \multicolumn{2}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots} \\ \multicolumn{2}{c}{\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots} \\ y_k' = f_k = f_k(t, y_1, y_2, ..., y_k), & y_k(t_0) = y_{k0} \end{array} \tag{3}$$

be a general nonlinear $K \times K$ system. Then we obtain the general Jacobian at step n, as

$$A_n = \begin{pmatrix} \frac{\partial f_1}{\partial y_1} & \frac{\partial f_1}{\partial y_2} & ... & \frac{\partial f_1}{\partial y_k} \\ \frac{\partial f_2}{\partial y_1} & \frac{\partial f_2}{\partial y_2} & ... & \frac{\partial f_2}{\partial y_k} \\ ... & ... & ... & ... \\ \frac{\partial f_k}{\partial y_1} & \frac{\partial f_k}{\partial y_2} & ... & \frac{\partial f_k}{\partial y_k} \end{pmatrix} = \begin{pmatrix} D_1 & D_2 & ... & D_k \\ D_{k+1} & D_{k+2} & ... & D_{2k} \\ ... & ... & ... & ... \\ D_{k^2-k+1} & D_{k^2-k+2} & ... & D_{k^2} \end{pmatrix} \tag{4}$$

where $D_j, j = 1(1)k^2$, are, in general, functions of (t_n, y_n).
To find A_n^2, we write

$$A_n^2 = A_n A_n \tag{5}$$

Let $U_j, j = 1(1)k^2$ be the entries of A_n^2. Since we need to obtain $(U_1, U_2, ...U_{k^2})^T$ as easily as possible we shall use the Shortcut of Matrix Transpositions (SMT), as follows
Define

$$\begin{array}{c} (A_n^2)^T = [U_j] \\ (A_n)^T = [D_j] \end{array} \tag{6}$$

Then, since $A_n^2 = A_n A_n$, it follows that $(A_n^2)^T = A_n^T A_n^T$. Therefore

$$[U_j] = [D_j][D_j] \tag{7}$$

From examining the full matrix multiplication of (7), we find that the product of the matrices has K partitions with K equations in each partition. Each equation in each partition has K terms. The first partition gives the vector $(U_1, U_2, ...U_k)^T$, the second partition gives $(U_{k+1}, U_{k+2}, ..., U_{2k})^T$ etc, and the K^{th} partition gives the vector $(U_{k^2-k+1}, U_{k^2-k+2}, \dots U_{k^2})^T$. Let partition r, be denoted by p(r). Then we have the following formulas for U_j in each partition;

$$
\begin{aligned}
&p(1): U_j = D_j D_1 + D_{k+j} D_2 + D_{2k+j} D_3 + \cdots + D_{k^2-k+j} D_k, && j = 1(1)k \\
&p(2): U_{k+j} = D_j D_{k+1} + D_{k+j} D_{k+2} + D_{2k+j} D_{k+3} + \cdots + D_{k^2-k+j} D_{2k}, && j = 1(1)k \\
&p(3): U_{2k+j} = D_j D_{2k+1} + D_{k+j} D_{2k+2} + D_{2k+j} D_{2k+3} + \cdots + D_{k^2-k+j} D_{3k}, && j = 1(1)k
\end{aligned}
$$

$$
\begin{gathered}
p(k): U_{k^2-k+j} = D_j D_{k^2-k+1} + D_{k+j} D_{k^2-k+2} + D_{2k+j} D_{k^2-k+3} \\
+ \cdots + D_{k^2-k+j} D_{k^2}, \quad j = 1(1)k
\end{gathered} \tag{8}
$$

For a 2×2 system, for example, we have $[U_j] = [D_j][D_j], j = 1(1)4$. There are K = 2 partitions. Then applying the formulas (8) we obtain

$$
\begin{aligned}
p(1): U_j &= D_j D_1 + D_{2+j} D_2, \quad j = 1(1)2 \\
p(2): U_{2+j} &= D_j D_3 + D_{2+j} D_4, \quad j = 1(1)2
\end{aligned} \tag{9}
$$

Taking j = 1(1)2, in each partition, we obtain the **totality expansion** for U_j, i.e

$$
\begin{gathered}
U_1 = D_1^2 + D_3 D_2 \\
U_2 = D_2 D_1 + D_4 D_2 \\
U_3 = D_1 D_3 + D_3 D_4 \\
U_4 = D_2 D_3 + D_4^2
\end{gathered} \tag{10}
$$

Thus

$$
A_n^2 = \begin{pmatrix} U_1 & U_2 \\ U_3 & U_4 \end{pmatrix} \tag{11}
$$

For a 3×3 system, there are k = 3 partitions with 9 entries of the Jacobian. Applying (8) to $[U_j] = [D_j][D_j]$ we have the following, in 3 partitions.

$$
\begin{aligned}
p(1): U_j &= D_j D_1 + D_{3+j} D_2 + D_{6+j} D_3, \quad j = 1(1)3 \\
p(2): U_{3+j} &= D_j D_4 + D_{3+j} D_5 + D_{6+j} D_6, \quad j = 1(1)3 \\
p(3): U_{6+j} &= D_j D_7 + D_{3+j} D_8 + D_{6+j} D_9, \quad j = 1(1)3
\end{aligned} \tag{12}
$$

Taking j = 1(1)3 in each partition, we obtain the totality expansion for U_j,
i.e

$$
\begin{gathered}
U_1 = D_1^2 + D_4 D_2 + D_7 D_3 \\
U_2 = D_2 D_1 + D_5 D_2 + D_8 D_3 \\
U_3 = D_3 D_1 + D_6 D_2 + D_9 D_3 \\
U_4 = D_1 D_4 + D_4 D_5 + D_7 D_6 \\
U_5 = D_2 D_4 + D_5^2 + D_8 D_6 \\
U_6 = D_3 D_4 + D_6 D_5 + D_9 D_6 \\
U_7 = D_1 D_7 + D_4 D_8 + D_7 D_9 \\
U_8 = D_2 D_7 + D_5 D_8 + D_8 D_9 \\
U_9 = D_3 D_7 + D_6 D_8 + D_9^2
\end{gathered} \tag{13}
$$

For any $K \times K$ system and for $j = 1(1)k^2$, define the matrix or the power of the matrix A_n, with its entries in brackets, as follows;

$A_n(D_j), A_n^2(U_j), A_n^3(W_j), A_n^4(V_j), A_n^5(G_j), \ldots, A_n^{p-1}(Q_j)$, where the exponent p, is the desired order of the method.

Now suppose $A_n^m = A_n^{m-1}A_n$ for all integral $m \geq 1$.
Then

$$(A_n^m)^T = A_n^T(A_n^{m-1})^T = [D_j](A_n^{m-1})^T \tag{14}$$

Using the general notation $[\cdot]$ for transpose of matrices we have successively;

$$\begin{array}{c} [U_j] = [D_j][D_j] \quad for A_n^2 \\ [W_j] = [D_j][U_j] \quad for A_n^3 \\ [V_j] = [D_j][W_j] \quad for A_n^4 \\ [G_j] = [D_j][V_j] \quad for A_n^5 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ [Q_j] = [D_j][R_j] \quad for A_n^{p-1} \end{array} \tag{15}$$

where R_j are the entries of A_n^{p-2}, and p is the desired order of the method. Each matrix entries on the LHS of (15) is obtained by applying the type of formulas (8) to the variables on the RHS in K partitions.

For example, W_j as the entries of A_n^3 for a 3×3 system, are given in 3 partitions. Since $[W_j] = [D_j][U_j]$ in (15), we have the following;

$$\begin{array}{ll} p(1) : W_j = D_jU_1 + D_{3+j}U_2 + D_{6+j}U_3, & j = 1(1)3 \\ p(2) : W_{3+j} = D_jU_4 + D_{3+j}U_5 + D_{6+j}U_6, & j = 1(1)3 \\ p(3) : W_{6+j} = D_jU_7 + D_{3+j}U_8 + D_{6+j}U_9, & j = 1(1)3 \end{array} \tag{16}$$

The totality expansions for W_j are obtained and retained in terms of the D_j and U_j variables.

3. THE INTEGRATION FORMULAS

The EM of order p is given by

$$y_{n+1} = y_n + (hI + \frac{h^2}{2}A_n + \frac{h^3}{6}A_n^2 + \ldots + \frac{h^p}{p!}A_n^{p-1})f_n \tag{17}$$

where $p \geq 1$, is any desired order. Let $U_1, U_2, \ldots, U_{k^2}$ be the variable entries of A_n^2 and $Q_1, Q_2, \ldots, Q_{k^2}$ be the variable entries of the matrix A_n^{p-1}, whose values are obtained by the procedures described in Section 2. Then on component by component basis, (17) dissolves into the forms

$$\begin{array}{c} y_{1,n+1} = y_{1n} + Z_1f_{1n} + Z_2f_{2n} + \cdots + Z_kf_{kn} \\ y_{2,n+1} = y_{2n} + Z_{k+1}f_{1n} + Z_{k+2}f_{2n} + \cdots + Z_{2k}f_{kn} \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\ y_{k,n+1} = y_{kn} + Z_{k^2-k+1}f_{1n} + Z_{k^2-k+2}f_{2n} + \cdots + Z_{k^2}f_{kn} \end{array} \tag{18}$$

where

$$
\begin{aligned}
Z_1 &= h + \frac{h^2}{2}D_1 + \frac{h^3}{6}U_1 + ... + \frac{h^p}{p!}Q_1 \\
Z_2 &= \frac{h^2}{2}D_2 + \frac{h^3}{6}U_2 + ... + \frac{h^p}{p!}Q_2 \\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
Z_{k+2} &= h + \frac{h^2}{2}D_{k+2} + \frac{h^3}{6}U_{k+2} + ... + \frac{h^p}{p!}Q_{k+2} \\
Z_{k+3} &= \frac{h^2}{2}D_{k+3} + \frac{h^3}{6}U_{k+3} + ... + \frac{h^p}{p!}Q_{k+3} \\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots \\
Z_{k^2} &= h + \frac{h^2}{2}D_{k^2} + \frac{h^3}{6}U_{k^2} + ... + \frac{h^p}{p!}Q_{k^2}
\end{aligned}
\tag{19}
$$

Observe that each of $Z_1, Z_{k+2}, , Z_{k^2}$ has h, as a separate term. We shall define such Z variables as *Diagonal Z variables*. They can be located by inspection of (19). Otherwise the Diagonal Z variables, in the general $K \times K$ system, are located more precisely at the K points,

$$j = n(K+1) + 1, n = 0, 1, 2, \dots, (K-1) \tag{20}$$

where K is the dimension of the system.
Hence for a 2×2 system, K = 2, then j = 3n+1, for n = 0, 1.
Thus, j = 1 and j = 4 when n = 0, 1 respectively, which give Z_1 and Z_4 as the diagonal variables. Therefore more compactly, we write

$$Z_j = h + \frac{h^2}{2}D_j + \frac{h^3}{6}U_j + ... + \frac{h^p}{p!}Q_j \tag{21}$$

are diagonal variables which are located at the points
$j = n(K+1) + 1, n = 0, 1, 2, \dots, (K-1)$
and
$Z_j = \frac{h^2}{2}D_j + \frac{h^3}{6}U_j + ... + \frac{h^p}{p!}Q_j$
are not diagonal variables located at other integral points of $j \in (1, K^2)$.
By QBASIC convention,

$$
\begin{array}{ccc}
y_{1,n+1} = X1 & y_{1n} = Y1 & f_{1n} = F1 \\
y_{2,n+1} = X2 & y_{2n} = Y2 & f_{2n} = F2 \\
- - - - - & - - - - - & - - - - \\
y_{k,n+1} = X_K & y_{kn} = Y_K, & f_{kn} = F_K
\end{array}
\tag{22}
$$

Then, using $y_1(t_0) = y_{10}, y_2(t_0) = y_{20}, \dots, y_k(t_0) = y_{k0}$ as the initial values (ie starting points) for $Y1, Y2, \dots Y_K$ respectively, the integration formulas corresponding to (18) in QBASIC Codes become

$$
\begin{array}{c}
X1 = Y1 + Z1 * F1 + \ \ + Z_K * F_K \\
X2 = Y2 + Z_{K+1} * F1 + \ \ + Z_{2K} * F_K \\
-------------- \\
-------------- \\
X_K = Y_K + Z_{K^2-K+1} * F1 + \cdots + Z_{K^2} * F_K
\end{array} \tag{23}
$$

which are programmed to obtain the automatic solutions of the $K \times K$ system.

4. PRELUDE TO THE INTEGRATION OF NONLINEAR STIFF AND NONSTIFF (ODE) SYSTEMS

4.1 Autonomous and Non-Autonomous Systems

Briefly (or more formally) a non-autonomous system is of the form of $\dfrac{dy}{dt} = f(t, y)$ where the differential equations of the system are functions of t and y.
If they are functions of y only i.e

$$\frac{dy}{dt} = f(y) \tag{24}$$

the system is called an autonomous system.

4.2 Choice of Stepsize for a Non-linear Non-Stiff System

Definition 4.1:
Let $A_0 = \dfrac{\partial f}{\partial y}(t_0, y_0)$ be the constant Jacobian of the nonstiff system at (t_0, y_0). Then, we define the stepsize h of the integration as follows;

- h = .0001, if each entry of A_0 is a maximum of two digits or less, to the nearest whole number or,

- $h = 10^{-r-1}$, if the largest entry of A_0, by absolute value, is a real number of r digits such that $r > 2$, to the nearest whole number. A nonstiff system having such $r > 2$ is said to be a **peculiar nonstiff system**.

4.3 Choice of Stepsize for a Nonlinear Stiff System

We shall obtain a **tentative initial stepsize** h_0^*, using the Jacobian A_0 and a **substantive initial stepsize** h_0 using the Jacobian A_1.

Let a K-dim nonlinear **stiff** system be given by

$$y' = f(t, y), y(t_0) = y_0 \tag{25}$$

such that the general Jacobian at step n, is

$$A_n = \frac{\partial f}{\partial y}(t_n, y_n) \tag{26}$$

Definition 4.2(a):
The Jacobian (26) at the initial point (t_0, y_0) is given by

$$A_0 = \frac{\partial f}{\partial y}(t_0, y_0) = \begin{pmatrix} \alpha_{11} & \alpha_{12} & ... & \alpha_{1k} \\ \alpha_{21} & \alpha_{22} & ... & \alpha_{2k} \\ ... & ... & ... & ... \\ \alpha_{k1} & \alpha_{k2} & ... & \alpha_{kk} \end{pmatrix} \tag{27}$$

where the α_{ij} are real entries. For some i,j, let α_{ij} be such that $|\alpha_{ij}|$ is the largest modulus of all the entries of A_0.
Then we define the *tentative initial stepsize* h_0^* as follows:

- if $|\alpha_{ij}| < 1$, then $h_0^* = 0.001$.

- if $|\alpha_{ij}|$ is an integer of r digits, to the nearest integer,

$$h_0^* = min(0.001, 10^{(-r-1)}) \tag{28}$$

if the system is **autonomous** or

$$h_0^* = min(0.0001, 10^{(-r-1)}) \tag{29}$$

if the system is **non-autonomous**

Definition 4.2(b)
Suppose $A_0 = 0$ in Definition 4.2(a), and $f_0 \neq 0$. Let the function values at (t_0, y_0) be given by $f_0 = (f_{10}, f_{20}, ...f_{k_0})^T$
Let $f_{j0}, 1 \leq j \leq K$, be such that $|f_{j0}|$ is the largest of the modulus of the components of f_0, then we define the *tentative initial stepsize* as follows:

- If $|f_{j0}| < 1$, then $h_0^* = 0.001$.
- If $|f_{j0}|$ is an integer of s digits, to the nearest integer,

then
$h_0^* = min(0.001, 10^{-s-1})$, if the system is **autonomous**
$h_0^* = min(0.0001, 10^{(-s-1)})$, if the system is **non-autonomous**

If $A_0 = A_1 = ... = A_n = 0, i.e \frac{\partial f}{\partial y} = 0.$ and $f_0 \neq 0$, the EM integration formula reduces to a method of order one (which coincides with the Euler Scheme) i.e $y_{n+1} = y_n + hIf_n$, where for the relevant system, we define $h = h_0^*$, as given by Definition 4.2(b) above. Also f_n is always the corrected f_n, if the system is non-autonomous. The correction is done by applying *Jibunoh correction for continuity* [1]

In order to obtain the substantive initial stepsize h_0 for the integration, we apply the Exponential Method (EM) of order 2 say, to the nonlinear stiff system (25), and solve for y_1 using A_0 and the tentative initial stepsize h_0^*.

From (2) the EM (order 2) gives the first step solution as

$$y_1 = y_0 + \left(h_0^* I + \frac{h_0^{*2}}{2} A_0 \right) f_0 \tag{30}$$

Having obtained y_1, then substituting (t_1, y_1) in the general Jacobian (26), we obtain the constant Jacobian

$$A_1 = \frac{\partial f}{\partial y}(t_1, y_1) = \begin{pmatrix} \beta_{11} & \beta_{12} & ... & \beta_{1k} \\ \beta_{21} & \beta_{22} & ... & \beta_{2k} \\ ... & ... & ... & ... \\ \beta_{k1} & \beta_{k2} & ... & \beta_{kk} \end{pmatrix} \tag{31}$$

where β_{ij} are real entries.

Definition 4.3
For some i, j, let β_{ij} in (31) be such that $|\beta_{ij}|$ is the largest modulus of all the entries of A_1. Then we define the *substantive initial stepsize* h_0, as follows:

- If $|\beta_{ij}| < 1$, then $h_0 = 0.001$.
- If $|\beta_{ij}|$ is an integer of r digits, to the nearest integer,

$$h_0 = min(0.001, 10^{-r-1}) \tag{32}$$

if the system is **autonomous**, or

$$h_0 = min(0.0001, 10^{-r-1}) \tag{33}$$

if the system is **non-autonomous**.
Thus h_0, which is substantive, is the required initial stepsize for the integration of the nonlinear system (25).

4.4 The Most Transient Eigenvalue of A_1

Definition 4.4
For the K-dim stiff system (25), let $\lambda_1, \lambda_2, \ldots, \lambda_k$ be the eigenvalues of the constant Jacobian A_1, defined by (31). Let $Re\lambda_1$ say, be the most negative of all the eigenvalues of A_1 i.e
$|Re\lambda_1| > |Re\lambda_j|$ for all j = 2(1)K.

Then λ_1 is said to be the most transient eigenvalue of the Jacobian A_1.

4.5 A Good Substitute for the Most Transient Eigenvalue of A_1

The following definition is not intended to give the exact numerical value of the most transient eigenvalue of A_1. This is not necessary for our purpose since the magnitude only is sufficient.

Definition 4.5
Let the constant Jacobian A_1 be as defined by (31). A good substitute or a good representative of the most transient eigenvalue of A_1, is the most negative entry i.e the negative entry βrs such that $|\beta rs|$ is the largest modulus of all the negative entries of A_1.

This definition is best applicable for practical solutions of stiff systems, using the Exponential Method, rather than evaluating for the exact eigenvalues which are more of theoretical interest.

5. PHASE TO PHASE INTEGRATION OF STIFF SYSTEMS

Let a time interval, $t_0 \leq t \leq t_N$, be given for the integration of a stiff system. Suppose that the interval is divided into two parts, defined as $[t_0, t_m]$ and $(t_m, t_N]$ respectively, such that we can carry out the integration in the first and second intervals separately. Then, the integration is said to be a two-phase integration. The point t_m is crucial, since it has to be located mathematically between t_0 and t_N such that the first interval $[t_0, t_m]$ is the transient phase (Phase I) containing the transient eigenvalues of the Jacobian, while the second interval $(t_m, t_N]$ is the steady-state phase (Phase 2).

A phase to phase integration of more than two phases is possible, eg. a three-phase integration. Our concern (in order to enhance stability and convergence) is to vanish the transient eigenvalues in Phase I, by integrating with a very small stepsize h_0, obtained from Definition 4.3 and then using a larger stepsize $h \geq h_0$ from Definition 5.1 below for phase 2 or other subsequent phases of the integration. Reference may be made to the work in [2], for preliminary details.

Implementation of a phase to phase integration is illustrated in Examples

1 and 2 (Section 7), for autonomous and non-autonomous stiff systems respectively.

We note that if $h_0 = .001$ or $.0001$, and the ultimate step is not considered too large, we can institute a one-phase integration and take $h_0 = h$ directly. For a phase to phase integration the initial point of a new phase is always the ultimate point of the last phase.

Definition 5.1
Let the initial stepsize h_0 obtained from Definition 4.3, be applied in phase I of the integration of a stiff system, then the required stepsize for Phase 2 and any other subsequent phases is;
h= 0.001 or h = 0.0001, if the system is autonomous,
h = 0.0001, if the system is non-autonomous
In general, $h_0 \leq h$.

5.1 Exact Formulas for Determining the Ultimate Step m and the Corresponding Timepoint t_m of Phase I

While developing the logic in Jibunoh [2], we first assumed for simplicity that the system under consideration is 2-dimensional (i.e, K = 2) with λ_1 and λ_2 as the eigenvalues of the constant Jacobian A_0 (now A_1) such that λ_1 is transient and λ_2 is not, and λ_1 is seen to vanish at the ultimate step m, of Phase I, at the corresponding point t_m. The work in Jibunoh [2], may be consulted for the background logic.

By λ_1 vanishing at step m of Phase I we mean that we can select a smallest positive integer m, possibly of high magnitude, such that

$$e^{Re(\lambda_1)h_0 m} = 0 \; exactly, \tag{34}$$

where λ_1 is the most transient eigenvalue of the Jacobian A_1. Practically we replace $Re(\lambda_1)$ in (34), by the most negative entry of the Jacobian A_1, according to Definition 4.5.

In the general case in which dim $K > 2$, there may be other transient eigenvalues of A_1 apart from the most transient. The most transient eigenvalue λ_1 as defined, is usually vanished when

$$e^{Re(\lambda_1)h_0\overline{m}} = b \times 10^{-r} = 0 \; approximately, \tag{35}$$

where $0 < b \leq 10$ and $r \geq 14$, at a step $\overline{m} << m$, in (34). Therefore, in the remaining steps $m - \overline{m}$, of Phase I, other transient eigenvalues of smaller magnitudes are assumed to have vanished. Hence (34) is global for vanishing all transient eigenvalues.

To determine the number m, in (34), we define -λ_1 as the most negative entry of the Jacobian A_1.

If (34) must hold exactly, two simultaneous equations must be solved, i.e

$$m = \frac{t_m - t_0}{h_0}$$
$$e^{-\lambda_1 h_0 m} = 0 \tag{36}$$

Solving, we have

$mh_0 = t_m - t_0$.

Then

$e^{-\lambda_1 h_0 m} = e^{-\lambda_1 (t_m - t_0)} = e^{-\lambda_1 t_m} e^{\lambda_1 t_0} = 0$

i.e

$$e^{-\lambda_1 t_m} = 0$$

or

$$e^{-\lambda_1 t_m} = 10^{-100(10^r)} \tag{37}$$

where $r \geq 0, is\ an$ integer.

It follows that

$-\lambda_1 t_m \log_{10} e = -100(10^r)$

Therefore

$$t_m = \frac{100(10^r)}{|\lambda_1| \log_{10} e} \tag{38}$$

where $r \geq 0$ is the integral index called the *adjusting index*, meant for approximating t_m to any number of decimal places.

Knowing t_m, we now obtain

$$m = \frac{t_m - t_0}{h_0} \tag{39}$$

We define t_m as a real number which will comply with any of the following conditions.

If the system is **autonomous**, t_m must be approximated to **3 decimal places**, by adjusting the index r to $r \geq 0$, or simply by approximating t_m directly to 3 decimal places and making r = 0, whichever case applies.

If the system is **non-autonomous**, we adopt the same procedure as above to approximate t_m to **4 decimal places**.

To illustrate the above conditions of t_m, for an autonomous system, we have;

Condition I: Suppose from (38), $t_m = 0.00012578 \times 10^r$. Taking r = 1, we have, $t_m = 0.0012578$ then to 3 decimal places $t_m = 0.001$. Also taking r = 2, $t_m = 0.012578$. Then approximating to 3 decimal places, we obtain $t_m = 0.013$. It is also possible to have $t_m = 0.12578$, by taking r = 3 and therefore obtaining $t_m = 0.126$ to 3 decimal places. It is often preferable to **use the least t_m possible** in which case, we take $t_m = 0.001$, as the required t_m, for the system, instead of 0.013 or 0.126.

Condition 2: Suppose $t_m = 0.2275812 \times 10^r$. Then the approximation is straightforward. In this case, we take $t_m = 0.228$ to 3 decimal places, with r = 0. Also if $t_m = 4.52 \times 10^r$ say, then we make r = 0 and write $t_m = 4.520$, to 3 decimal places.

The above conditions also apply *mutatis mutandis* to **non-autonomous** systems where t_m must be approximated to **4 decimal places**.

By Definition 5.1, the integration stepsize for Phase 2 or other subsequent phases is h = 0.001 or 0.0001, if the system is autonomous and h = 0.0001, if the system is non-autonomous. Therefore, the restrictions on the decimal numbers of t_m, which are to conform to conditions 1 and 2 above, are meant to avoid fractional step-numbers in all phases of the integration.

5.2 Locating a Numerical Solution Corresponding to the Theoretical Solution y(t) for a given Real Number t, in any Phase of the Integration

Defintion 5.2
Let there be a two-phase integration of a stiff system in the interval $t_0 \leq t \leq t_N$, where the first and second phases of the integration occur in the subintervals $[t_0, t_m]$ and $(t_m, t_N]$ respectively. Then, for any $t > t_0$, the bounds of t with the stepnumber formulas for points of t, in each phase are defined as follows:

$$\begin{aligned} Phase\ I : t_0 < t \leq t_m,\ n+1 &= \frac{t-t_0}{h_0} \\ Phase\ 2 : t_m < t \leq t_N,\ n+1 &= \frac{t-t_m}{h} \end{aligned} \tag{40}$$

where t_m is known in (38), and h_0 and h are the stepsizes employed for the integrations in Phase I and Phase 2 respectively, as obtained from Definitions 4.3 and 5.1. The correspondence of the given t, with step $n+1$, implies that the theoretical solution y(t) corresponds to the numerical solution, y_{n+1}.

The points to be noted here are as follows:

- The particular real number t, for which the stepnumber n+1, is to be found should be approximated to no more than the number of decimal places of h_0 in phase I, or no more than the number of decimal places of h in phase 2, depending on which of the phases the point t, is located. The number of decimal places could be less or much less than the maximum of that of h_0 or h. For example t, could be an integer such as 5, which has a zero number of decimal places.

Conversely, a given stepnumber n+1, could be used to determine a corresponding t, by applying

$$\begin{aligned} t &= t_0 + (n+1)h_0, \quad in \quad Phase\ I \\ or\ t &= t_m + (n+1)h, \quad in \quad Phase\ 2 \end{aligned} \tag{41}$$

- An r-phase integration (where $r > 2$) will have stepnumber formulas determined for each phase by using the terminal points of phases.

Remark 5.1

- For a NONSTIFF system, the general stepsize for all integration is $h = .0001$ or the stepsize dictated in Definition 4.1, for peculiar cases of nonstiff systems.
  If the ultimate step of the integration for the nonstiff system is very large, we may institute a two phase integration (not to vanish any eigenvalue in Phase I) but to make the ultimate step of each phase smaller for computer evaluation. In this case, the terminal timepoint of Phase I is taken as t_1, which may be chosen arbitrarily, or by the rule $t_1 = \frac{1}{2}t_N$, where t_N is the ultimate timepoint of the integration. The point t_1 should be a whole number or approximated to no more than 4 decimal places, since $h = 0.0001$ (or less for peculiar cases of nonstiff systems). For an r-phase integration, we define $t_1 = \frac{1}{r}t_N$, where t_1 is a whole number or is approximated to at most 4 decimal places, being that h = 0.0001. Then the *terminal points* of phases from Phase I, shall be $t_1, 2t_1, \ldots, rt_1$ where $rt_1 = t_N$. For a **stiff system**, the terminal points of phases are $t_m, t_2, 2t_2, \ldots, (r-1)t_2$ where $(r-1)t_2 = t_N$, since after Phase I, with the terminal point t_m, we define $t_2 = \frac{1}{r-1}t_N$, where t_2 is a whole number or is approximated as t_m, by the procedures in section 5.1. The case r = 2, corresponds to a two-phase integration.

- For a LINEAR system, the constant Jacobian A coincides with the Jacobian A_1 of the nonlinear system such that all definitions pertaining to A_1 applies to the Jacobian A, of the linear system.

6. OPTIMAL ORDERS FOR SYSTEMS

The choice of orders of the EM for the integration of any system may not be arbitrary. From experiments, we are able to find the optimal orders for categories of systems which are summarized in Table 6.1. The optimal orders are verified but higher orders may be used at will.

Table 6.1 Optimal Orders for systems

Systems	Optimal order
All Nonlinear/linear NONSTIFF systems.	3
Nonlinear/linear STIFF (autonomous) systems with real eigenvalues.	4
Nonlinear/Linear STIFF (autonomous) systems with complex eigenvalues	6
All other nonlinear/linear STIFF non-autonomous systems, including stiff oscillatory systems and stiff systems with uncertain nature	6

7. NUMERICAL APPLICATIONS

EXAMPLE 1: Nonlinear Stiff System

$y_1' = f_1 = .01 - (.01 + y_1 + y_2)[1 + (1000 + y_1)(1 + y_1)]$

$y_2' = f_2 = .01 - (.01 + y_1 + y_2)(1 + y_2^2)$

$y_1(0) = y_2(0) = 0. \quad 0 \le t \le 100$

This stiff nonlinear 2×2 autonomous system with real eigenvalues was obtained from Lambert [10] and Fatunla [11] where it was integrated by other numerical methods. No theoretical solution is available but a theoretical solution at the terminal point t = 100, is deemed to be found after a strenuous application of the Explicit Runge Kutta method of order 4, with h = .0005, as reported in Lambert [10]. This problem was also solved in Jibunoh [2], using the EM of order 3. Here we shall solve by applying the EM of order 4.

By following the procedures outlined in sections 3, 4 and 5 the integration is carried out in two phases in the interval $0 \le t \le 100$.

The inputs of the Phases are now as follows

PHASE I

$t_0 = 0, \ \ y_0 = (0, 0)^T, \ \ h_0 = 0.00001$

$D_1 = -1011.01 - 2002.02y_{1n} - 3y_{1n}^2 - 1001y_{2n} - 2y_{1n}y_{2n}$

$D_2 = -1001 - 1001y_{1n} - y_{1n}^2$

$D_3 = -1 - y_{2n}^2$

$D_4 = -1 - 0.02y_{2n} - 2y_{1n}y_{2n} - 3y_{2n}^2$

$f_{1n} = .01 - (.01 + y_{1n} + y_{2n})[1 + (1000 + y_{1n})(1 + y_{1n})]$

$f_{2n} = .01 - (.01 + y_{1n} + y_{2n})(1 + y_{2n}^2)$

Using the most negative entry of the Jacobian A_1 as the most transient eigenvalue we obtain $t_m = 0.228$, as the ultimate t of Phase I, corresponding to step $m = N_1$
Step: $n + 1 = \dfrac{t - t_0}{h_0} = \dfrac{t}{0.00001}, 0 < t \leq 0.228.$
Ultimate step $N_1 = \dfrac{0.228}{0.00001} = 22,800 = m$
Steps of integration: n = 0 to 22,799 i.e n = 0 to $N_1 - 1$, since when n $= N_1 - 1$, $y_{n+1} = y_{N_1}$.

PHASE 2

Initial point of Phase 2 is the ultimate point of Phase I,
i.e $t_0 = t_m = .228$, $y_0 = y_m = y_{22,800}$ of Phase I, h = .001
D_1, D_2, D_3, D_4 and f_{1n}, f_{2n} remain unchanged as in Phase I, since the system is autonomous.
Step: $n + 1 = \dfrac{t - t_m}{h} = \dfrac{t - .228}{0.001}, .228 < t \leq 100$
Ultimate step $N_2 = \dfrac{100 - .228}{.001} = 99,772$
Steps of integration: n = 0 to 99,771

Applying the automatic integration formulas (23) of **order 4** to generate the first and second phases respectively in the interval $0 \leq t \leq 100$, the automatic numerical solutions are obtained and compared with those of EM(order 3) and Fatunla [11], in Table 7.1(a).

Table 7.1(b) compares the solutions of different methods at the terminal point t = 100, with that of y(t) as found.

Table 7.1(a): Comparing EM solutions of Example I with those of Fatunla [11] (Given EM solutions are domiciled in Phase 2 of the integration)

t	$n+1=\dfrac{t-.228}{.001}$	EM (order 4)	EM(Order 3)	Fatunla [11]
10	9772	-0.1097544	-0.1100537	-0.1131583
		0.0997768	0.1000761	0.1031919
20	19772	-0.2095082	-0.2098074	-0.2140978
		0.1995334	0.1998327	0.2041358
50	49772	-0.5084115	-0.5087100	-0.5177467
		0.4984520	0.4987505	0.5078083
100	99772	-0.9916421	-0.9918163	-0.9990020
		0.9833364	0.9835357	0.9940184

Table 7.1(b) Comparing Solutions of different methods at the terminal point of Example I

t	y(t) (As found)	EM (order 4)	EM(Order 3)	Fatunla [11]	Lambert [10]
100	-0.9916	-0.9916	-0.9918	-0.9990	-0.9990
	0.9833	0.9833	0.9835	0.9940	0.9940

The EM solutions are far superior to the solutions of Fatunla [11] in Table 7.1(a) especially when the solutions at the terminal point t = 100 are compared to four decimal places in Table 7.1(b) with the theoretical solutions, y(t) as found. The solutions of EM (order 4) coincide with the exact theoretical solutions at t = 100. The solutions of Fatunla [11] and Lambert [10] are largely in error.

We, therefore, conclude that the EM (order 4) produced the exact theoretical solutions in the whole interval $0 \leq t \leq 100$.

The sample programs in QBASIC Codes with automatic outputs for this problem, at Phase I and Phase 2 integrations respectively, are given in Appendices A_1 and A_2.

EXAMPLE 2: Linear Stiff Oscillatory System

$y_1' = f_1 = 9y_1 + 24y_2 + 5cost - \frac{1}{3}sint$

$y_2' = f_2 = -24y_1 - 51y_2 - 9cost + \frac{1}{3}sint$

$y_1(0) = \frac{4}{3}, \quad y_2(0) = \frac{2}{3}$

This is a moderately stiff non-homogeneous and oscillatory linear system from Burden and Faires [15], p. 314 which has trigonometric functions on the RHS. It has been solved also in [1], using Jibunoh Spectral Decomposition.

The theoretical solutions are given by
$y_1(t) = 2e^{-3t} - e^{-39t} + \frac{1}{3}cost$
$y_2(t) = -e^{-3t} + 2e^{-39t} - \frac{1}{3}cost$
Now the Jacobian is the constant matrix

$$A = \begin{pmatrix} D_1 & D_2 \\ \\ D_3 & D_4 \end{pmatrix} = \begin{pmatrix} 9 & 24 \\ \\ -24 & -51 \end{pmatrix}$$

with real eigenvalues λ_1 = -39, λ_2 = -3. The most negative entry of the Jacobian is -51.
Since the system is non-autonomous, the initial step-size $h_0 = min(0.0001, 10^{-3}) = 0.0001$, where we have taken h_0 directly (without first obtaining h_0^*) because the Jacobian is a constant matrix which does not change at all points of the integration. See also Remark 5.1.

A two-phase integration is carried out in the interval $0 \leq t \leq 10.75$.
The inputs of the phases are now as follows;

PHASE 1:
$t_0 = 0, \quad y_0 = \left(\frac{4}{3}, \ \frac{2}{3} \right)^T, h_0 = 0.0001$, by Definition 4.3.
$D_1 = 9$
$D_2 = 24$
$D_3 = -24$
$D_4 = -51$
Since the system is non-autonomous, we have at step n,
$t_n = t_0 + nh_0 + \frac{h_0}{2} = 0.0001n + 0.00005$, where $\frac{h_0}{2}$ is *Jibunoh correction for continuity* [1]

Therefore, from the system,
$f_{1n} = 9y_{1n} + 24y_{2n} + 5R_{1n} - \frac{1}{3}R_{2n}$

$f_{2n} = -24y_{1n} - 51y_{2n} - 9R_{1n} + \frac{1}{3}R_{2n}$
where
$R_{1n} = cos(0.0001n + 0.00005)$
$R_{2n} = sin(0.0001n + 0.00005)$

Let $R_{1n} = R_1, R_{2n} = R_2, f_{1n} = F1$ and $f_{2n} = F2$

The QBASIC program requires that trigonometric functions (or functions of functions) be evaluated first, before substitution into equations. Thus R1 and R2 will precede F1 and F2 in the program. $t_m = 4.5148$, is the ultimate t of Phase I, corresponding to step m = N_1

Step: $n + 1 = \frac{t - t_0}{h_0} = \frac{t}{.0001}, 0 < t \leq 4.5148$

Ultimate step $N_1 = \frac{4.5148}{.0001} = 45,148 = m$

Steps of integration: n = 0 to 45,147

PHASE 2:

The inputs are as follows:
$t_0 = t_m = 4.5148$, $y_0 = y_m = y_{45,148}$ of Phase I, h = 0.0001 by Definition 5.1, for Phase 2 of a non-autonomous stiff system.
D_1, D_2, D_3, D_4 remain unchanged as in Phase I, since they are constants and independent of t
At step n, in this Phase,

$$t_n = t_m + nh + \frac{h}{2} = 4.5148 + 0.0001n + 0.00005 = 0.0001n + 4.51485$$

Thus,

$$f_{1n} = 9y_{1n} + 24y_{2n} + 5R_{1n} - \frac{1}{3}R_{2n}$$

$$f_{2n} = -24y_{1n} - 51y_{2n} - 9R_{1n} + \frac{1}{3}R_{2n}$$

where now in Phase 2
$R_{1n} = cos(0.0001n + 4.51485)$
$R_{2n} = sin(0.0001n + 4.51485)$

Step: $n + 1 = \frac{t - t_m}{h} = \frac{t - 4.5148}{.0001}, \; 4.5148 < t \leq 10.75$

Ultimate step $N_2 = \frac{10.75 - 4.5148}{.0001} = 62,352$

Steps of integration: n = 0 to 62,351

Applying the integration formulas (23) of **Order 6** to the system, at Phase I and Phase 2 respectively, the automatic numerical solutions are obtained to 8 decimal places and compared with the theoretical solutions in Table 7.2, in the interval $0 \leq t \leq 10.75$. The table indicates the Phase in which each numerical solution is obtained.

Table 7.2 Comparing the solutions of EM (order 6) with the theoretical solutions of Example 2

t	Phase	n + 1	y(t)	y_{n+1}:EM (order 6)
0.001	1	10	1.36559145	1.36559145
			0.59316376	0.59316376
1.0	1	10,000	0.27967491	0.27967491
			-0.22988784	-0.22988784
1.6	1	16,000	0.00672632	0.00672632
			0.00150343	0.00150342
4.5148	1	45,148	-0.06543264	-0.06543264
			0.06543395	0.06543395
8.4561	2	39,413	-0.18879652	-0.18879652
			0.18879652	0.18879652
10.75	2	62,352	-0.08103781	-0.08103781
			0.08103781	0.08103781

Phase I: $n+1 = \dfrac{t}{0.0001}, 0 < t \leq 4.5148,$ Phase 2: $n+1 = \dfrac{t-4.5148}{0.0001}, 4.5148 < t \leq 10.75$

Clearly from Table 7.2, the EM solutions coincide with the exact theoretical solutions to 8 decimal places at all points of the integration. This shows the efficiency of the EM in handling stiff oscillatory systems.

EXAMPLE 3: Nonlinear Nonstiff System

$$y_1' = f_1 = \frac{y_1 - y_2}{y_3 - t}, \qquad y_1(0) = 4.693147181$$

$$y_2' = f_2 = \frac{y_1 - y_2}{y_3 - t}, \qquad y_2(0) = 3.693147181$$

$$y_3' = f_3 = y_1 - y_2 + 1, \qquad y_3(0) = 2$$

This is a nonlinear nonstiff (non-autonomous) 3×3 system adapted from Krasnov et al [9], p. 215.

The theoretical solutions are given by
$y_1(t) = ln|t + 2| + 4$
$y_2(t) = ln|t + 2| + 3$
$y_3(t) = 2(t + 1)$

With the usual stepsize $h = 0.0001$ for a nonstiff system, a one phase integration is carried out in the interval $0 \leq t \leq 10$. The system is non-autonomous. Therefore at step n, $t_n = t_0 + nh + \frac{h}{2} = 0.0001n + 0.00005$. Applying the QBASIC integration formulas of **order 3**, the automatic numerical solutions are obtained and compared with the theoretical solutions in Table 7.3.

Table 7.3 Comparing the solutions of EM (order 3) with the theoretical solutions of Example 3

t	$n + 1 = \frac{t}{.0001}$	y(t)	EM (order 3), y_{n+1}
5.6	56000	6.02814820	6.02814820
		5.02814820	5.02814820
		13.20000000	13.20000000
7.835	78350	6.28594750	6.28594750
		5.28594750	5.28594750
		17.67000000	17.67000000
10	100000	6.48490670	6.48490670
		5.48490670	5.48490670
		22.00000000	22.00000000

We find from Table 7.3, that the numerical solutions coincide with the theoretical solutions (to 8 decimal places or more) at all points of the integration, which shows, as in [2], that the EM is, in general, efficient for nonstiff nonlinear ODE systems.

EXAMPLE 4: The Robertson (Nonlinear stiff) Chemical Problem

$y_1' = -.04y_1 + 10^4 y_2 y_3, \quad y_1(0) = 1$

$y_2' = .04y_1 - 10^4 y_2 y_3 - 3 \times 10^7 y_2^2, \quad y_2(0) = 0$

$y_3' = 3 \times 10^7 y_2^2, \quad y_3(0) = 0$

This is a stiff (nonlinear) autonomous 3×3 system obtained from [13] p.51 and also from [12]. The interval of integration is $0 \leq t \leq 400$. By following the procedure in Example I, the system is integrated in two phases with the **EM (Order 4)**. The automatic outputs for selected points of t are exhibited in Table 7.4 and compared with the results of the Second Derivative GLM of order 4 of Butcher and Hojjati [12]. No theoretical solution is available. We find from the entries of the Jacobian A_1 that $h_0 = 0.00001$. Then applying (38) we obtain $t_m = 0.096$, to 3 decimal places, since the system is autonomous. Therefore m = 9,600 is the ultimate step of phase I

Table 7.4 Comparing solutions of EM (Order 4) with those of Butcher and Hojjati [12] for the Robertson Chemical Problem

t	Phase	n + 1	EM(Order 4)	Butcher and Hojjati [12]
0.09	1	9000	$9.964630170502634 \times 10^{-1}$	———
			$3.587457435343882 \times 10^{-5}$	———
			$3.501108358154222 \times 10^{-3}$	———
0.4	2	304	$9.851721141312358 \times 10^{-1}$	$9.85172113862063 \times 10^{-1}$
			$3.38639535827037 \times 10^{-5}$	$3.38639537959540 \times 10^{-5}$
			$1.479402191935487 \times 10^{-2}$	$1.47940221359022 \times 10^{-2}$
4	2	3904	$9.05518679180256 \times 10^{-1}$	$9.05518678434419 \times 10^{-1}$
			$2.240475689897206 \times 10^{-5}$	$2.24047569380437 \times 10^{-5}$
			$9.44589160512483 \times 10^{-2}$	$9.44589159917086 \times 10^{-2}$
40	2	39904	$7.323942195485953 \times 10^{-1}$	$7.15827069891020 \times 10^{-1}$
			$1.820462563171801 \times 10^{-6}$	$9.18553464163141 \times 10^{-6}$
			$2.676039616552051 \times 10^{-1}$	$2.84163750795415 \times 10^{-1}$
400	2	399904	$7.214754660561276 \times 10^{-1}$	$4.50518690834087 \times 10^{-1}$
			$-4.758033751248198 \times 10^{-13}$	$3.22290106126097 \times 10^{-6}$
			$2.78525358013144 \times 10^{-1}$	$5.49478203523904 \times 10^{-1}$

Phase I: $n+1 = \dfrac{t}{0.00001}$, $0 < t \leq 0.096$, Phase 2: $n+1 = \dfrac{t - 0.096}{0.001}$, $0.096 < t \leq 400$

There is a fair agreement between the EM (Order 4) and Butcher and Hojjati results (of order 4) up to t = 4. At t = 40 and t = 400, the disparity is evident.

Generally, there is a steady decrease of the first component solution for the two methods, as t increases. However, the EM **second** component solution converges rapidly to zero from t = 40 to t = 400. At t = 400, the EM solution is $-4.758033751248198 \times 10^{-13}$, which is essentially zero.

In contrast, the Butcher and Hojjati second component solution at t = 400 is $3.2290106126097 \times 10^{-6}$ which is not necessarily zero. The second and third component solutions of the EM at t = 40 and t = 400 respectively, are less in magnitude than the counterpart Butcher and Hojjati solutions. The differences are much clearer at t = 400.

The antecedents of the Exponential Method, e.g obtaining exact theoretical solutions, up to the given number of decimal places, in Examples 1,2 and 3 now compel us to assert that the EM(Order 4) results should, in all probability, be the exact theoretical solutions of the Robertson equations.

8. CONCLUSION

The Exponential Method of variable order, developed in this paper, has demonstrated its efficiency and simplicity of application. The variable order of the method obviously paved the way for higher accuracy especially with reference to examples 1, 2, and 3 in section 7, in which the numerical solutions coincided with the exact theoretical solutions, to eight or more decimal places. The exactitude of the solutions in the three cited examples leads to the obvious deduction that the solutions of example 4 (Robertson chemical reaction problem), being given to more than 15 decimal places, ought invariably to be the exact theoretical solutions.

The remarkable features of the method include; the use of the exact Jacobians of nonlinear systems; the phase to phase integration of stiff systems, in which the transient eigenvalues are vanished in phase I; avoidance of matrix inversions, LU decompositions and the cumbersome Newton iterations, since the method is explicit; the easy handling of autonomous and non-autonomous systems without any orchestrated show of disparity; the solving of oscillatory systems without additional refinements and a straightforward application of the method without starters. It is evident from numerical applications that the Exponential Method has faster convergence and much higher accuracy than many existing methods. The method is also capable of solving small and large (stiff and nonstiff) ODE systems which are nonlinear or linear.

A first order scalar ODE with initial point (t_0, y_0) can be solved by the Exponential Method either singly or as a 2×2 system, after incorporating a dummy first order scalar ODE defined with the same initial point (t_0, y_0). In particular, two or more independent first order scalar equations with a common time point t_0, in their initial values, can be solved simultaneously. All scalar equations of higher orders are generally solved by the method after their simple reduction to first order systems. Hence, we may declare that the Exponential Method of variable order is omnibus.

Observe that the order of the EM can be changed easily in the program for automatic computation. Suppose we have presently a method of order 3. The variables of order 3, are D_j and U_j respectively, $j = 1(1)k^2$, as defined in section 2, as the entries of A_n and A_n^2. These variables appear as numbered lines of the program. To move to order 4, we derive the W_j variables, which are entries of A_n^3 and type them into the program to follow the U_j variables. For example, using (15), a 2×2 system has the following W_j variables;

$$\begin{aligned} W_1 &= D_1U_1 + D_3U_2 \\ W_2 &= D_2U_1 + D_4U_2 \\ W_3 &= D_1U_3 + D_3U_4 \\ W_4 &= D_2U_3 + D_4U_4 \end{aligned} \tag{42}$$

We next include the term $\dfrac{h^4}{24}W_j$ as an additional term of order 4, in the Z_j variables, $j = 1(1)k^2$. After this, the resulting method becomes a method of order 4. Likewise, we can move from order 4 to order 5, sequentially, etc, by copying and pasting (42) in the next lines. Since V_j are the entries of A_n^4 and $[V_j] = [D_j][W_j]$ by (15), we give command to the computer to change W to V and U to W in the pasted lines. We then increase the terms of the Z_j variables by adding $\dfrac{h^5}{120}V_j$,$j = 1(1)k^2$, to finally create a method of order 5. Reducing the method from a higher to a lower order is by deleting the relevant variables of the next higher orders. Therefore, change of order is achieved without stress.

Implementations show that any program of the Exponential Method of variable order (e.g the QBASIC program) produces a very fast or instant output in automatic computation.

ACKNOWLEDGEMENTS

The author wishes to thank the referees for their incisive and useful comments which contributed to the improvement of the original version of this paper.

APPENDIX A1

EM ORDER 4: Sample Program and Automatic Output for Example I: Phase 1

```
10 OPTION BASE 1
20 REM
30 OPEN "O", #1, "C:OUTPUT JEMNONLINEARORDER42X2
PHASE1"
40 REM
50 DEFDBL A-B
60 DEFDBL Y
70 DEFDBL X
80 REM****************************************
90 REM SEGMENT TO READ THE VALUES OF THE
100 REM FIRST AND SECOND ARRAY
110 REM***************************************
120 DIM A(2, 2), B(2, 2)
130 FOR ROW = 1 TO 2
  140 FOR COL = 1 TO 2
    150 READ A(ROW, COL)
  160 NEXT COL
170 NEXT ROW
 230 READ B(ROW1, COL1)
240 NEXT COL1
250 NEXT ROW1
260 DATA 1,0
270 DATA 0,1
290 REM
300 REM
310 REM****************************************
320 REM SEGMENT TO READ THE VALUES OF Y(N)
330 REM FOR N EQUAL TO 1,2
340 REM******************************************
350 LET Y1 = 0
360 LET Y2 = 0
370 REM
390 CLS
400 FOR N = 0 TO 22799
  401 LET D1 = (-1011.01 - 2002.02 * Y1 - 3 * Y1 ^ 2 - 1001 *
Y2 - 2 * Y1 * Y2)
  402 LET D2 = (-1001 - 1001 * Y1 - Y1 ^ 2)
  403 LET D3 = (-1 - Y2 ^ 2)
  404 LET D4 = (-1 - 0.02 * Y2 - 2 * Y1 * Y2 - 3 * Y2 ^ 2)
  405 LET U1 = D1 ^ 2 + D3 * D2
  406 LET U2 = D2 * D1 + D4 * D2
  407 LET U3 = D1 * D3 + D3 * D4
  408 LET U4 = D2 * D3 + D4 ^ 2
  409 LET W1 = D1 * U1 + D3 * U2
  410 LET W2 = D2 * U1 + D4 * U2
  411 LET W3 = D1 * U3 + D3 * U4
  412 LET W4 = D2 * U3 + D4 * U4
  417 LET F1 = (.01 - (.01 + Y1 + Y2) * (1 + (1000 + Y1) * (1 +
Y1)))
  418 LET F2 = (.01 - (0.01 + Y1 + Y2) * (1 + Y2 ^ 2))
  419 LET Z1 = (0.00001 + (0.00001 ^ 2 / 2) * D1 + (0.00001 ^ 3
/ 6) * U1 + (0.00001 ^ 4 / 24) * W1)
  420 LET Z2 = ((0.00001 ^ 2 / 2) * D2 + (0.00001 ^ 3 / 6) * U2 +
(0.00001 ^ 4 / 24) * W2)
  421 LET Z3 = ((0.00001 ^ 2 / 2) * D3 + (0.00001 ^ 3 / 6) * U3 +
(0.00001 ^ 4 / 24) * W3)
  422 LET Z4 = (0.00001 + (0.00001 ^ 2 / 2) * D4 + (0.00001 ^ 3
/ 6) * U4 + (0.00001 ^ 4 / 24) * W4)
  423 LET X1 = Y1 + Z1 * F1 + Z2 * F2
  424 LET X2 = Y2 + Z3 * F1 + Z4 * F2
  440 LET X3 = A(1, 1) * Y1 + A(1, 2) * Y2
  450 LET X4 = A(2, 1) * Y1 + A(2, 2) * Y2
  470 LET X5 = B(1, 1) * X1 + B(1, 2) * X2
  480 LET X6 = B(2, 1) * X1 + B(2, 2) * X2
  500 LET Y1 = X3 + X5
  510 LET Y2 = X4 + X6
  530 REM
  540 REM
  550 REM
  560 PRINT #1, "N="; N; ","; "Y"; N + 1; "="; Y1; Y2
570 NEXT N
580 END
```

```
N= 0 ,Y 1 =-9.949619197868742D-05  4.983175677653051D-10
N= 1 ,Y 2 =-1.979917436267424D-04  1.986575276256232D-09
N= 2 ,Y 3 =-2.954969196707531D-04  4.454808326325055D-09
N= 3 ,Y 4 =-3.92021880492845D-04  7.893154055003103D-09
N= 4 ,Y 5 =-4.875766639068942D-04  1.229185087885786D-08
N= 5 ,Y 6 =-5.821712209384714D-04  1.76412372790226D-08
N= 6 ,Y 7 =-6.75815380884106D-04  2.393175074856601D-08
N= 7 ,Y 8 =-7.685188869148563D-04  3.115392725794065D-08
N= 8 ,Y 9 =-8.602913588750118D-04  3.929839878411652D-08
N= 9 ,Y 10 =-9.511423217479760-04  4.835589345315014D-08

N= 999 ,Y 1000 =-1.006914043656563D-02  8.9789117707228D-05
N= 1000 ,Y 1001 =-1.006924502484933D-02  8.988891124108990-05
N= 1001 ,Y 1002 =-1.006934956583326D-02  8.99887048212808D-05
N= 1002 ,Y 1003 =-1.006945405999002D-02  9.008849844780306D-05
N= 1003 ,Y 1004 =-1.006955850777374D-02  9.0188292120659D-05
N= 1004 ,Y 1005 =-1.006966290964774D-02  9.028808583985087D-05
N= 1005 ,Y 1006 =-1.006976726606616D-02  9.0387879605381D-05
N= 1006 ,Y 1007 =-1.006987157747383D-02  9.048767341725160-05
N= 1007 ,Y 1008 =-1.006997584431552D-02  9.058746727546488D-05
N= 1008 ,Y 1009 =-1.007008006702681D-02  9.068726118002305D-05
N= 1009 ,Y 1010 =-1.007018424605251D-02  9.078705512161516D-05
N= 1010 ,Y 1011 =-1.007028838181896D-02  9.0886849109556SD-05
N= 1011 .Y 1012 =-1.007039247475242D-02  9.098664314384921D-05
N= 2971 ,Y 2972 =-1.026642426360832D-02  2.8659097178111D-04
N= 2972 ,Y 2973 =-1.026652405990553D-02  2.866907701058862D-04
N= 2973 ,Y 2974 =-1.026662385620271D-02  2.867905684306624D-04
N= 2974 ,Y 2975 =-1.026672365249981D-02  2.868903667554386D-04
N= 2975 ,Y 2976 =-1.026682344879688D-02  2.869901650802148D-04
N= 2976 ,Y 2977 =-1.026692324509391D-02  2.870899634049911D-04
N= 2977 ,Y 2978 =-1.026702304139088D-02  2.871897617297673D-04
N= 2978 ,Y 2979 =-1.026712283768781D-02  2.872895600545435D-04
N= 2979 ,Y 2980 =-1.026722263398466D-02  2.873893583793198D-04

N= 2980 ,Y 2981 =-1.026732243028148D-02  2.87489156704096D-04
N= 2981 ,Y 2982 =-1.026742222657827D-02  2.875889550288722D-04
N= 2982 ,Y 2983 =-1.026752202287498D-02  2.876887533536484D-04
N= 2983 ,Y 2984 =-1.026762181917163D-02  2.877885516784247D-04
N= 2984 ,Y 2985 =-1.026772161546827D-02  2.87888350003201D-04
N= 2985 ,Y 2986 =-1.026782141176485D-02  2.879881483279771D-04
N= 2986 ,Y 2987 =-.0102679212080614  2.880879466527533D-04

N= 2987 ,Y 2988 =-1.026802100435785D-02  2.881877449775296D-04
N= 2988 ,Y 2989 =-1.026812080065428D-02  2.882875433023058D-04
N= 2989 ,Y 2990 =-1.026822059695068D-02  2.88387341627082D-04
N= 2990 ,Y 2991 =-1.026832039324701D-02  2.884871399518582D-04
N= 2991 ,Y 2992 =-1.026842018954326D-02  2.885869382766345D-04
N= 2992 ,Y 2993 =-1.026851998583951D-02  2.886867366014107D-04

N= 22795 ,Y 22796 =-1.224478197495052D-02  2.264989031743318D-03
N= 22796 ,Y 22797 =-1.224488177082733D-02  2.265088829667631D-03
N= 22797 ,Y 22798 =-1.224498156671336D-02  2 265188627591945D-03
N= 22798 ,Y 22799 =-1.224508136259936D-02  2.265288425516258D-03
N= 22799 ,Y 22800 =-1.224518115847603D-02  2.265388223440571D-03
```

APPENDIX A2

EM ORDER 4: Sample Program and Automatic Output for Example I: Phase 2

```
10 OPTION BASE 1
20 REM
30 OPEN "O", #1, "C:OUTPUT JEMNONLINEARORDER42X2 PHASE2"
40 REM
50 DEFDBL A-B
60 DEFDBL Y
70 DEFDBL X
80 REM*****************************************
90 REM SEGMENT TO READ THE VALUES OF THE
100 REM FIRST AND SECOND ARRAY
110 REM*****************************************
120 DIM A(2, 2), B(2, 2)
130 FOR ROW = 1 TO 2
  140 FOR COL = 1 TO 2
    150 READ A(ROW, COL)
  160 NEXT COL
170 NEXT ROW
180 DATA 0,0
190 DATA 0,0
200 REM
210 FOR ROW1 = 1 TO 2
  220 FOR COL1 = 1 TO 2
    230 READ B(ROW1, COL1)
  240 NEXT COL1
250 NEXT ROW1
260 DATA 1,0
270 DATA 0,1
290 REM
300 REM
310 REM*****************************************
320 REM SEGMENT TO READ THE VALUES OF Y(N)
330 REM FOR N EQUAL TO 1,2
340 REM*****************************************
350 LET Y1 = -1.224518115847603D-02
360 LET Y2 = 2.265388223440571D-03
370 REM
390 CLS
400 FOR N = 0 TO 99771
  401 LET D1 = (-1011.01 - 2002.02 * Y1 - 3 * Y1 ^ 2 - 1001 * Y2 - 2 * Y1 * Y2)
  402 LET D2 = (-1001 - 1001 * Y1 - Y1 ^ 2)
  403 LET D3 = (-1 - Y2 ^ 2)
  404 LET D4 = (-1 - 0.02 * Y2 - 2 * Y1 * Y2 - 3 * Y2 ^ 2)
  405 LET U1 = D1 ^ 2 + D3 * D2
  406 LET U2 = D2 * D1 + D4 * D2
  407 LET U3 = D1 * D3 + D3 * D4
  408 LET U4 = D2 * D3 + D4 ^ 2
  409 LET W1 = D1 * U1 + D3 * U2
  410 LET W2 = D2 * U1 + D4 * U2
  411 LET W3 = D1 * U3 + D3 * U4
  412 LET W4 = D2 * U3 + D4 * U4
  417 LET F1 = (.01 - (.01 + Y1 + Y2) * (1 + (1000 + Y1) * (1 + Y1)))
  418 LET F2 = (.01 - (0.01 + Y1 + Y2) * (1 + Y2 ^ 2))
  419 LET Z1 = (0.001 + (0.001 ^ 2 / 2) * D1 + (0.001 ^ 3 / 6) * U1 + (0.001 ^ 4 / 24) * W1)
  420 LET Z2 = ((0.001 ^ 2 / 2) * D2 + (0.001 ^ 3 / 6) * U2 + (0.001 ^ 4 / 24) * W2)
  421 LET Z3 = ((0.001 ^ 2 / 2) * D3 + (0.001 ^ 3 / 6) * U3 + (0.001 ^ 4 / 24) * W3)
  422 LET Z4 = (0.001 + (0.001 ^ 2 / 2) * D4 + (0.001 ^ 3 / 6) * U4 + (0.001 ^ 4 / 24) * W4)
  423 LET X1 = Y1 + Z1 * F1 + Z2 * F2
  424 LET X2 = Y2 + Z3 * F1 + Z4 * F2
  440 LET X3 = A(1, 1) * Y1 + A(1, 2) * Y2
  450 LET X4 = A(2, 1) * Y1 + A(2, 2) * Y2
  470 LET X5 = B(1, 1) * X1 + B(1, 2) * X2
  480 LET X6 = B(2, 1) * X1 + B(2, 2) * X2
  500 LET Y1 = X3 + X5
  510 LET Y2 = X4 + X6
  530 REM
  540 REM
  550 REM
  560 PRINT #1, "N="; N; ","; "Y"; N + 1, "="; Y1; Y2
570 NEXT N
    580 END
```

```
N= 0 ,Y 1 =-.0122551607472709  2.27536801625241D-03
N= 1 ,Y 2 =-1.226514033518952D-02  2.285347809072406D-03
N= 2 ,Y 3 =-1.227511992369275D-02  2.295327601901546D-03
N= 3 ,Y 4 =-1.228509951243018D-02  2.305307393808223D-03
N= 4 ,Y 5 =-1.229507910029135D-02  2.315287185723342D-03
N= 5 ,Y 6 =-1.230505868815193D-02  2.325266977646687D-03
N= 6 ,Y 7 =-1.231503827601191D-02  2.335246769578541D-03
N= 7 ,Y 8 =-1.232501786323477D-02  2.345226560588215D-03
N= 8 ,Y 9 =-1.233499745016224D-02  2.355206351606048D-03
N= 9 ,Y 10 =-1.234497703708911D-02  2.365186142632674D-03
N= 10 ,Y 11 =-.0123549566240154  2.375165933667525D-03
N= 11 ,Y 12 =-1.236493621065063D-02  2.385145724710886D-03
N= 12 ,Y 13 =-.0123749157975201  2.395125514832066D-03
N= 13 ,Y 14 =-1.238489:38351326D-02  2.405105304961406D-03
N= 14 ,Y 15 =-.0123948749692154  2.41508509509954D-03
N= 15 ,Y 16 =-1.240485455579263D-02  2.42506488524653D-03
N= 16 ,Y 17 =-1.241483414114751D-02  2.435044674470424D-03
N= 17 ,Y 18 =-.012424813726207  2.445024463702762D-03
N= 18 ,Y 19 =-1.243479331155631D-02  2.455004252943608D-03
N= 19 ,Y 20 =-.0124447728966146  2.464984042193247D-03
N= 20 ,Y 21 =-1.245475248138184D-02  2.474963831451395D-03
N= 21 ,Y 22 =-1.246473206638333D-02  2.484943619786796D-03
N= 22 ,Y 23 =-.0124747116505085  2.49492340813064D-03
N= 23 ,Y 24 =-.0124846912346331  2.504903196482992D-03
N= 24 ,Y 25 =-1.249467081875707D-02  2.514882984844138D-03
N= 25 ,Y 26 =-1.250465040224398D-02  2.52486277228282D-03
N= 26 ,Y 27 =-1.251462998602074D-02  2.534842559729728D-03

N= 9771 ,Y 9772 =-.1097543568481156  .0997767741237857
N= 9772 ,Y 9773 =-.1097643339566924  9.978675148284244D-02
N= 9773 ,Y 9774 =-.109774311064656З  9.979672884191584D-02
N= 9774 ,Y 9775 =-.1097842881726194  .0998067062010057
N= 9775 ,Y 9776 =-.109794265280553  .0998166835591816
N= 9776 ,Y 9777 =-.1098042423875832  9.982666091737415D-02

N= 19771 ,Y 19772 =-.2095082089338047  .1995334493939586
N= 19772 ,Y 19773 =-.2095181823703S2  .199543423147367
N= 19773 ,Y 19774 =-.2095281558062573  .1995533969008008
N= 19774 ,Y 19775 =-.2095381292415207  .1995633706542601
N= 19775 ,Y 19776 =-.2095481026764928  .199573344406815

N= 49771 ,Y 49772 =-.508411501545373  .498452019597099
N= 49772 ,Y 49773 =-.508421450143718  .498461969011 7358
N= 49773 ,Y 49774 =-.5084313987412736  .498471918425499
N= 49774 ,Y 49775 =-.5084413473374093  .4984818678374576
N= 49775 ,Y 49776 =-.508451295931969  .4984918172476114
N= 49776 ,Y 49777 =-.508461244525739  .4985017666557338
N= 49777 ,Y 49778 =-.508471193116619  .4985117160620507

N= 99765 ,Y 99766 =-.9916069853651068  .9832963249325498
N= 99766 ,Y 99767 =-.991612838221326  .9833030013976897
N= 99767 ,Y 99768 =-.9916186889241335  .9833096762196821
N= 99768 ,Y 99769 =-.9916245374719235  .9833163493979067
N= 99769 ,Y 99770 =-.991630383863985  .9833230209314321
N= 99770 ,Y 99771 =-.991636228098251S  .9833296908193266
N= 99771 ,Y 99772 =-.9916420701733375  .9833363590606584
```

REFERENCES

[1] C.C. Jibunoh, *Accurate and Automatic Integration of Stiff and Non Stiff (ODE) systems via Spectral Decomposition*, Journal of Nigerian Mathematical Society (JNMS)Vol 29, 21-39, 2010.

[2] C.C. Jibunoh, *An exponential method for accurate and automatic integration of nonlinear (stiff and nonstiff) ODE systems*, J. Nigerian Math. Soc. **34**:143-159, 2015. Also available online in http://www.sciencedirect.com/science/article/pii/S0189896514000079

[3] B.S. Desale, N.R. Dasre, *Numerical Solutions of the system of six coupled nonlinear ODEs by Runge Kutta fourth order method*, Journal of Applied Mathematical Sciences Vol 7, 289 305, 2013.

[4] A. Pedas, E. Tamme, *Numerical Solution of nonlinear fractional differential equations by spline collocation methods*, Journal of Computer and Applied Mathematics Vol 255, 216 230, 2014.

[5] A.A. Opanuga, S.O. Edeki, H.I. Okagbue, G.O. Akinlabi, *On numerical solutions of systems of ODEs by numerical-analytical method*, Journal of Applied Mathematical Sciences, Vol I, no 164; 8199 8207, 2014.

[6] B.S. Desale, N.R. Dasre, *Numerical Solution to system of six coupled nonlinear ODEs by Adomian decomposition method*, International Journal of Mathematics and Computer research Vol3, issue 2, 876-887, Feb. 2015.

[7] J. Duan,*Higher order numeric solutions for nonlinear systems based on the Rach-Adomian-Meyers modified decomposition method*, Journal of Applied Mathematics and Physics Vol2, 1-7, 2014.

[8] E. Suli, *Numerical solutions of ordinary differential equations*, Mathematical Institute, University of Oxford, 2014.

[9] M.L. Krasnov, A.I. Kiselyov and G.I. Makarenko, *A Book of Problems in Ordinary Differential Equations*, Mir Publishers, Moscow, 1989.

[10] J.D. Lambert,*Non linear Methods for Stiff Systems of Ordinary Differential Equations Conference on the Numerical Solutions of Differential Equations*, (Dundee, 1973) Lecture Notes in Math Vol 363, Springer-Verlag, Berlin and New York, Pp 75-88, 1974.

[11] S.O. Fatunla, *An Implicit Two-Point Numerical Integration Formula for Linear and Nonlinear Stiff Systems of Ordinary Differential Equations*, Mathematics of Computation,.Vol 32 No 141, pp1-11, 1978.

[12] J.C. Butcher and G. Hojjati,*Second Derivative Methods with RK stability*, Numerical Algorithms 40, 415 429 Springer, 2005.

[13] Z. Jackiewicz, *General Linear Methods for Ordinary Differential Equations*, John Wiley and Sons Inc., Publications, 2009.

[14] R.I. Okuonghae and M.N.O. Ikhile,*Second Derivative General Linear Methods*, Numerical Algorithms, DOI 10.1007/S11075-013-9814-8, Springer, 2013.

[15] R.L. Burden and J.D. Faires,*Numerical Analysis (Fifth Edition)*, PWS Publishing Co., Boston MA. 1993

School of Science and Technology, National Open University of Nigeria, Asaba (S.C.), Nigeria.
E-mail address: chafachid@gmail.com