Algorithm 901: LMEF – A Program for the Construction of Linear Multistep Methods with Exponential Fitting for the Numerical Solution of Ordinary Differential Equations

D. S. VLACHOS and T. E. SIMOS
University of Peloponnese

LMEF is a program written in MATLAB, to calculate the coefficients of a linear multi-step method (explicit, implicit or backward differentiation formulas) with algebraic and/or exponential fitting, for the numerical solution of first order ordinary differential equations. Moreover, LMEF calculates the local truncation error and in the case of exponential fitting, the Taylor expansions of the coefficients that are necessary for the implementation of the method.

Categories and Subject Descriptors: G.1.7 [**Numerical Analysis**]: Ordinary Differential Equations—*Multistep and multivalued methods*

General Terms: Algorithms, Documentation

Additional Key Words and Phrases: Linear multistep methods, backward differentiation formulas, exponential fitting

ACM Reference Format:

Vlachos, D. S. and Simos, T. E. 2010. Algorithm 901: LMEF – A program for the construction of linear multistep methods with exponential fitting for the numerical solution of ordinary differential equations. ACM Trans. Math. Softw. 37, 1, Article 12 (January 2010), 10 pages.
DOI = 10.1145/1644001.1644013 http://doi.acm.org/10.1145/1644001.1644013

1. INTRODUCTION

Linear multi-step methods are widely used for the numerical solution of ordinary differential equations. They are implemented either explicitly or implicitly. Explicit methods calculate the unknown function at the next time step as a linear combination of already calculated values of the function and its derivative at previous time steps. Implicit methods need to solve an equation (usually nonlinear) that contains both the value of the unknown function

Authors' address: University of Peloponnese, Department of Computer Science and Technology, Terma O. Karaiskaki, 22100 Tripoli, Greece; email: {dvlachos;simos}@uop.gr.

DOI 10.1145/1644001.1644013 http://doi.acm.org/10.1145/1644001.1644013

and its derivative at the next time step. Both implementations need only one evaluation of the function derivative in every step and they can easily attain high orders, although large stability regions can be obtained only in implicit ones.

A refinement of linear multi-step methods is the exponential fitting, by which the method is forced to integrate exactly functions of the form $x^m e^{qx}$ where m is an integer called the *level of tuning* and q is either real or purely imaginary and it is called the *frequency of the method*. Traditionally, exponential fitting is used for the numerical solution of oscillatory problems. Many physical phenomena exhibit a pronounced oscillatory character: behavior of pendulum-like systems, vibrations, resonances and wave propagation are all phenomena of this type in classical mechanics, while the same is true for the typical behavior of quantum particles. Many exponential fitting methods (multi-step and hybrid methods) have been constructed and most were developed for second-order differential equations where the first derivative is absent. They have been successfully applied to the solution of the Schrödinger equation (a review of numerical methods can be found in Solin [2002]). On the other hand, Ixaru et al. [2001, 2002, 2003] have shown that the applicability of exponential fitting algorithms is much broader than thought. They have developed exponential fitting BDF algorithms combined with a frequency evaluation procedure for the solution of first-order equations. Some well-known stiff problems (like the Robertson [1996] and HIRES [Hairer and Wanner 1996] problems) which have no connection with oscillatory problems have been solved with this type of algorithm. Moreover, partitioned linear multi-step algorithms have been found to give stable solutions to Hamiltonian problems [Vlachos and Simos 2004].

Exponential fitting has found many applications in both orbital and stiff problems [Solin and Vigo-Aguiar 2001; Solin 2003]. These methods are forced to solve functions of the form $e^{\lambda t}$ exactly, where λ can be imaginary in order to account for trigonometric functions. Depending on how the parameter λ is selected, there are methods that use a stable λ through the whole integration and methods that modify it by adjusting the coefficients at every step (e.g., to minimize the local truncation error [Ixaru et al. 2002]).

In general, the construction of an exponentially fitted multi-step method has to do with the calculation of the coefficients of the method. As with every linear multi-step method, an exponentially fitted one should at least meet stability and consistency criteria. This means that the method, when applied to the problem $y'(x) = 0,\ \ y(x_0) = 0$, produces bounded solutions; in addition, it solves the problem $y'(x) = 1,\ \ y(x_0) = 0$ exactly. These are necessary conditions for a convergent method and every convergent method is consistent [Butcher 2003]. The rest of the conditions for the calculation of the coefficients can be obtained by forcing the method to integrate a number of selected functions exactly.

In this work, a MATLAB program called LMEF, has been developed for the calculation of the coefficients of an exponential fitted linear multi-step method. The input to the program contains the number of steps of the method and the number of frequencies to fit. The program builds the conditions that must hold and calculates the unknown coefficients. The local truncation error is calculated by comparing the Taylor expansion of the estimated and exact solution.

Extensive examples are presented as well as a numerical test with methods constructed with the software.

2. THEORETICAL BACKGROUND

Consider the first order differential equation of the form

$$y'(x) = f(x, y(x)), \quad y(x_0) = y_0 \tag{1}$$

and a linear multi-step method for the numerical integration of (1)

$$\sum_{j=0}^{k} a_j y_{n-j} - h \sum_{j=0}^{k} b_j y'_{n-j} = 0, \tag{2}$$

where

$$y_n = \hat{y}(x_0 + nh)$$
$$y'_n = f(x_0 + nh, y_n)$$

where $\hat{y}(x)$ is the numerical estimation of $y(x)$ and h is the integration step. The quantity

$$L(y, x, h) = \sum_{j=0}^{k} a_j y(x - jh) - h \sum_{j=0}^{k} b_j y'(x - jh) \tag{3}$$

is the *residual* of the method associated with the differentiable function y at point x with stepsize h. The method (2) is said to integrate exactly the differentiable function y if $L(y, x, h) = 0$ for every x in the integration interval. The linearity of (2) implies that if the method integrates exactly the functions $g_1, g_2, \ldots, g_m$ then it integrates exactly any linear combination of them. The method (2) is said to have order p (or is algebraic fitted up to order p) if it integrates any linear combination of the functions $\{1, x, x^2, \ldots, x^p\}$ exactly. Usually, the conditions that the coefficients a's and b's must satisfy in order to obtain a method of order p are calculated by taking the Taylor expansions of $y(x - jh)$ and $y'(x - jh)$ in (3). In this work, a different approach is adopted. Let $a = (a_0, a_1, \ldots, a_k)^T$ and $b = (b_0, b_1, \ldots, b_k)^T$. Suppose now that a set of conditions of the form

$$C \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0 \tag{4}$$

holds for (2) to integrate some given function g exactly, that is, $L(g, x, h) = 0$. We now consider the set of conditions that must hold if (2) integrates the product $x \cdot g$ of x with the function g exactly. It can easily be shown that

$$L(x \cdot g, x, h) = x \cdot L(g, x, h) - h \cdot \left(\sum_{j=0}^{k} (j a_j + b_j) g(x - jh) - h \cdot \sum_{j=0}^{k} j b_j g'(x - jh) \right) \tag{5}$$

and since $L(g, x, h) = 0$, we find that the same conditions that hold for (a, b) must hold for (a', b') where $a'_j = j a_j + b_j$ and $b'_j = j b_j$. Let $N_{k+1} =$

$diag(0, 1, 2, \ldots, k)$ and the matrix Λ of order $(2k + 2)$ as

$$\Lambda = \begin{pmatrix} N_{k+1} & I_{k+1} \\ 0 & N_{k+1} \end{pmatrix} \tag{6}$$

It can be easily shown that

$$N_{k+1}^n = diag(0, 1, 2^n, \ldots, k^n) \tag{7}$$

$$\Lambda^n = \begin{pmatrix} N_{k+1}^n & nN_{k+1}^{n-1} \\ 0 & N_{k+1}^n \end{pmatrix} \tag{8}$$

We can now prove the following theorem

THEOREM 2.1. *If the conditions*

$$C \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0$$

imply that (2) *integrates the function* g *exactly then the conditions*

$$\begin{pmatrix} C \\ C\Lambda \\ C\Lambda^2 \\ \ldots \\ C\Lambda^p \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0$$

imply that (2) *integrates any linear combination of the functions* $(g, xg, \ldots, x^p g)$ *exactly.*

PROOF. Obviously, the theorem holds for $p = 0$, since we have assumed that the method integrates the function g exactly. Let us now assume that the theorem holds for $p = k$. This means that the set of conditions

$$\begin{pmatrix} C \\ C\Lambda \\ C\Lambda^2 \\ \ldots \\ C\Lambda^k \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0$$

imply that (2) exactly integrates the function $\hat{g} = x^k g$. But we have shown in (5), that if a set of linear conditions $C \cdot (a, b)^T = 0$ implies that (2) exactly integrates a function $\hat{g}$, then the method exactly integrates the function $x\hat{g}$, if $C \cdot (a', b')^T = 0$, where $(a', b')^T = \Lambda \cdot (a, b)^T$. Thus, the method (2) exactly integrates the function $x\hat{g} = x^{k+1} g$ if

$$\begin{pmatrix} C \\ C\Lambda \\ C\Lambda^2 \\ \ldots \\ C\Lambda^k \end{pmatrix} \cdot \begin{pmatrix} a' \\ b' \end{pmatrix} = \begin{pmatrix} C \\ C\Lambda \\ C\Lambda^2 \\ \ldots \\ C\Lambda^k \end{pmatrix} \cdot \Lambda \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} C\Lambda \\ C\Lambda^2 \\ C\Lambda^3 \\ \ldots \\ C\Lambda^{k+1} \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0$$

Since $C\cdot(a,b)^T = 0, C\Lambda\cdot(a,b)^T = 0, C\Lambda^2\cdot(a,b)^T = 0, \ldots, C\Lambda^k\cdot(a,b)^T = 0$ hold from the induction assumption, the new condition $C\Lambda^{k+1}\cdot(a,b)^T = 0$ must also hold. Thus the set of conditions

$$\begin{pmatrix} C \\ C\Lambda \\ C\Lambda^2 \\ \cdots \\ C\Lambda^k \\ C\Lambda^{k+1} \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0 \tag{9}$$

implies that the method exactly integrates the function $x^{k+1}g$ and this completes the proof. □

The starting condition assumed here is

$$C_1 \begin{pmatrix} a \\ b \end{pmatrix} = 0\ ,\ C_1 = (\underbrace{1,1,\ldots,1}_{k+1},\underbrace{0,0,\ldots,0}_{k+1}) \tag{10}$$

which implies that (2) exactly integrates the function $g(x) = 1$. Algebraic order p can be achieved now if

$$\begin{pmatrix} C_1 \\ C_1\Lambda \\ C_1\Lambda^2 \\ \cdots \\ C_1\Lambda^p \end{pmatrix} \cdot \begin{pmatrix} a \\ b \end{pmatrix} = 0 \tag{11}$$

If (2) is to be explicit, we need to add the extra condition

$$C_0 \begin{pmatrix} a \\ b \end{pmatrix} = 0\ ,\ C_0 = (\underbrace{0,0,\ldots,0}_{k+1},1,\underbrace{0,\ldots,0}_{k}.) \tag{12}$$

Consider now the function $g(x) = e^{qx}\ ,\ q \in R$. The conditions that must hold for (2) to integrate the function g exactly are

$$C_e^q \begin{pmatrix} a \\ b \end{pmatrix} = 0\ ,\ C_e^q = (1, e^{-qh}, \ldots, e^{-kqh}, -qh, -qhe^{-qh}, \ldots, -qhe^{-kqh}) \tag{13}$$

Finally, consider the functions $g_1(x) = e^{qx}$ and $g_2(x) = e^{-qx}\ ,\ q \in iR$. This case is known as *trigonometric* fitting. The conditions that must hold for (2) to

integrate both g_1 and g_2 exactly are

$$C_t^q \begin{pmatrix} a \\ b \end{pmatrix} = 0$$

$$C_t^q = \begin{pmatrix} 1 & cosh(qh) & \dots & cosh(kqh) & 0 & qh \cdot sinh(qh) & \dots & qh \cdot sinh(kqh) \\ 0 & sinh(qh) & \dots & sinh(kqh) & qh & qh \cdot cosh(qh) & \dots & qh \cdot cosh(kqh) \end{pmatrix} \tag{14}$$

Consider now the general case where we have M_e different real exponents and M_t different imaginary exponents. The following theorem can now be easily proved.

THEOREM 2.2. *For the method* (2) *to integrate exactly any linear combination of the functions*

$$\{1, x, x^2, \dots x^p, e^{q_j x}, xe^{q_j x}, \dots, x^{r_j} e^{q_j x}, e^{\pm \hat{q}_l x}, xe^{\pm \hat{q}_l x} \dots, x^{t_l} e^{\pm \hat{q}_l x}\}$$

for $q_j \in R, j = 1, 2, \dots, M_e$ *and* $\hat{q}_l \in iR, l = 1, 2, \dots, M_t$ *the following conditions must apply*

$$\begin{pmatrix} C_1 \\ C_1\Lambda \\ \dots \\ C_1\Lambda^p \\ C_e^{q_j} \\ \dots \\ C_e^{q_j}\Lambda^{r_j} \\ C_t^{\hat{q}_l} \\ \dots \\ C_t^{\hat{q}_l}\Lambda^{t_l} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = 0.$$

If the method (2) *is explicit, the extra condition*

$$C_0 \begin{pmatrix} a \\ b \end{pmatrix} = 0$$

must also hold.

In practice, the a's are given and the construction of the method reduces to the calculation of the b's. Although there is no systematic way to define the a's for a new method, the following conditions must apply

(1) We assume that

$$C_1 \begin{pmatrix} a \\ b \end{pmatrix} = \sum_{j=0}^{k} a_j = 0.$$

(2) Consider the differential equation

$$y'(x) = 0$$

Method (2), when applied to the above equation reduces to the finite difference equation

$$\sum_{j=0}^{k} a_j \, y_{n-j} = 0,$$

which gives bounded solutions if the roots of the characteristic polynomial

$$p(z) = \sum_{j=0}^{k} a_j z^{k-j}$$

lie inside the unit disc $R = \{z| \, |z| \leq 1\}$, and those roots that lie on the unit circle $\partial R = \{z| \, |z| = 1\}$ must have multiplicity one.

On the other hand, in the case of Backward Differentiation Formulas, the method reduces to

$$\sum_{j=0}^{k} a_j \, y_{n-j} = h \cdot y'_n \tag{15}$$

which means that the b's are known and we have to calculate the a's. In this case, maximum algebraic order defines the values of the a'a.

3. IMPLEMENTATION DETAILS AND TEST EXAMPLE

LMEF is a MATLAB program that uses the result of Theorem (2.2) to calculate the coefficients of a linear multi-step method with exponential fitting. The program calculates the vector c of coefficients, their Taylor expansion up to a predefined order and the residual of the method.

Both the Taylor expansions of the c's and the calculation of the residual of the method require either MATLAB build-in functions or external functions from MAPLE like (for example, *mtaylor*) which may fail to compute the expansion especially in multiple frequency fitting although the limit of the expression under expansion always exists (as frequencies or step size tends to zero). To solve this problem, a function called *my_mtaylor* has been developed which separates the numerator and denominator of the expression under expansion, calculates the Taylor expansions of those two new expressions and finally calculates the Taylor expansion of their quotient. It can be easily proved that the expansion calculated in this way is exact up to the minimum order of expansion of the nominator and denominator.

LMEF requires as input the a's of the method, so it can calculate the b's. In the case of backward differentiation formulas, the values of the given a's are neglected and only the dimension of the a-vector is used (the number of steps).

LMEF performs some basic tests on the input parameters in the case of explicit or implicit methods. Using the coefficients a, it calculates the roots of the characteristic polynomial of the method and checks the following:

—Unity must be a root of the characteristic polynomial.

—The roots must lie in the unit disc, and those that lie on the unit circle must have multiplicity one, otherwise an error message is displayed and the calculations stop.

As a test example, we calculate the motion of two bodies in a reference system that is fixed in one of them. Moreover, the motion is planar, thus, we only have to calculate the x and y coordinates of the second body. The differential equations are

$$\begin{aligned} \ddot{x} &= -\frac{x}{\sqrt{x^2+y^2}^3} \\ \ddot{y} &= -\frac{y}{\sqrt{x^2+y^2}^3} \end{aligned}$$

and the initial conditions are

$$\begin{aligned} x(0) &= 1-\epsilon \qquad \dot{x}(0) = 0 \\ y(0) &= 0 \quad \dot{y}(0) = \sqrt{(1+\epsilon)/(1-\epsilon)}, \end{aligned}$$

where ϵ is the eccentricity. By setting $y_1 = x$, $y_2 = y$, $y_3 - y_1'$ and $y_4 = y_2'$, we get the first order system

$$\begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}' = \begin{pmatrix} y_3 \\ y_4 \\ -\frac{y_1}{\sqrt{y_1^2+y_2^2}^3} \\ -\frac{y_2}{\sqrt{y_1^2+y_2^2}^3} \end{pmatrix}. \tag{16}$$

In the first case (low eccentricity $= 0.2$) we consider the explicit method $EM-4$:

```
[c,ct,e]=lmef('e',[1,-1,1,-1]');
```

and the method $EM-4_{TF}$ with trigonometric fitting:

```
[c_o,ct_o,e_o]=lmef('e',[1,-1,1,-1]',[2]);
```

Figure 1 presents the calculated position of the second body using the methods $EM-4$ and $EM-4_{TF}$ and step size $h = 0.075$ for 112.5 periods. The positive effect of trigonometric fitting is obvious from this figure since it keeps the second body on the correct orbit. Moreover, the relative error in energy and angular momentum is almost one order of magnitude smaller in the case of trigonometric fitting.

In the second case (high eccentricity $= 0.75$) we consider the explicit method $EM-7$

```
[c,ct,e]=lmef('e',[1,-1,1,0,-1,1,-1]');
```

the method $EM-7_{TF}$ with trigonometric fitting

```
[c_o,ct_o,e_o]=lmef('e',[1,-1,1,0,-1,1,-1]',[2]);
```

and the BDF method $BDF-5$

```
[c_b,ct_b,e_b]=lmef('b',[1,1,1,1,1]');
```

Figure 2 presents the calculated position of the second body using methods $EM-7$, $EM-7_{TF}$ and $BDF-5$ with step size $h = 0.01$ for 7.5 periods. Simple

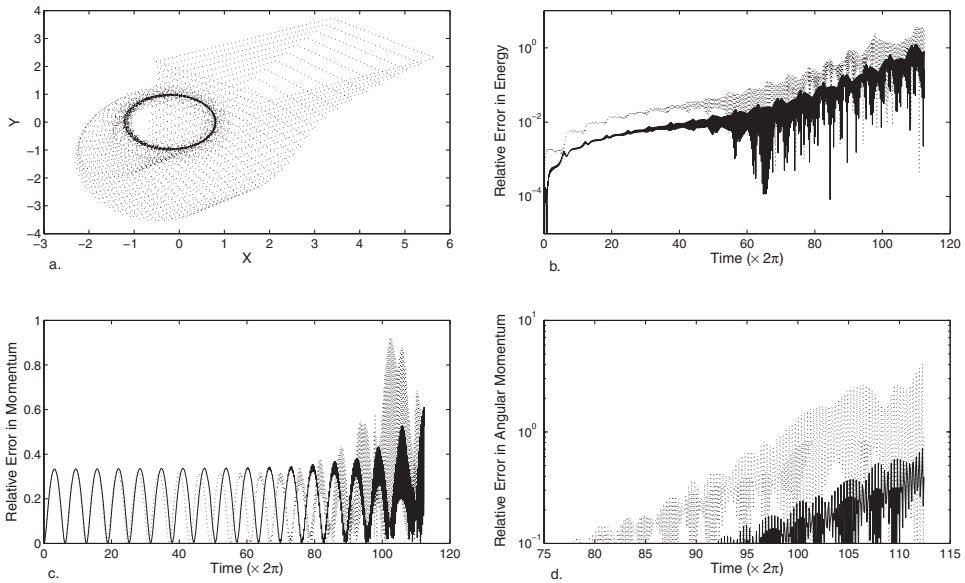

Fig. 1. Integration of the 2-body problem for eccentricity $\epsilon = 0.2$, step size $h = 0.075$ for 112.5 periods. (a) The calculated position of the second body using methods $EM - 4$ (dotted line) and $EM - 4_{TF}$ (solid line), (b) the relative error in energy for the method $EM - 4$ (dotted line) and $EM - 4_{TF}$ (solid line), (c) the relative error in momentum for the method $EM - 4$ (dotted line) and $EM - 4_{TF}$ (solid line) and (d) the relative error in angular momentum for the method $EM - 4$ (dotted line) and $EM - 4_{TF}$ (solid line).

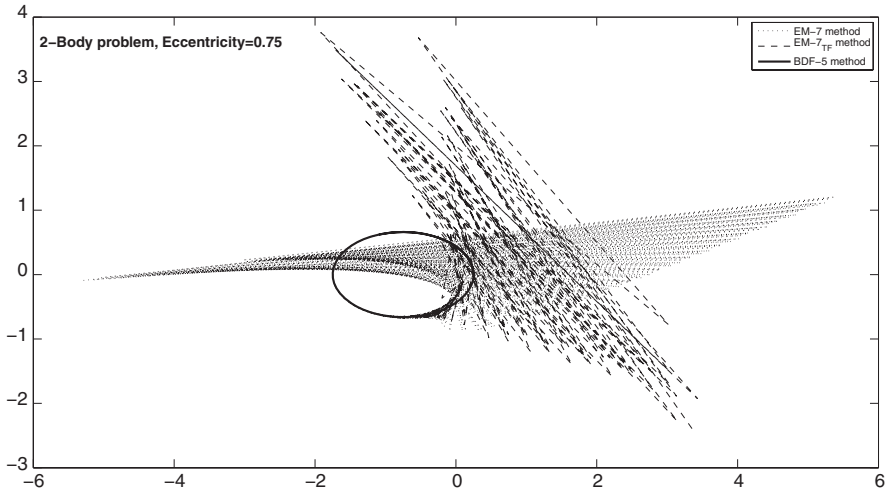

Fig. 2. The calculated position of the second body using methods $EM - 7$ (dotted line), $EM - 7_{TF}$ (dashed line) and $BM - 5$ (solid line) with step size $h = 0.01$ for 7.5*periods*. Both explicit method $EM-7$ and trigonometric fitted method $EM-7_{TF}$ deviate from the correct orbit, while the backward differentiation formula $BDF-5$ manages to handle the stiffness introduced by the high eccentricity by keeping the second body on the correct orbit.

trigonometric fitting cannot stabilize the solution, although the implicit BDF method keeps the second body on the expected orbit.

REFERENCES

BUTCHER, J. C. 2003. *Numerical Methods for Ordinary Differential Equations*. Wiley, New York, ISBN 0-471-96758-0.

HAIRER, E. AND WANNER, G. 1996. *Solving Ordinary Differential Equations II: Stiff and Differential Algebraic Problems*. Springer-Verlag, Berlin, Germany, second revised edition.

IXARU, L. GR. , RIZEA, M., VANDEN BERGHE, G., AND DE MEYER, H. 2001. Weights of the exponential fitting multistep algorithms for first order ODEs. *J. Comp. Appl. Math. 132*, 83–93.

IXARU, L. GR., VANDEN BERGHE, G., AND DE MEYER, H. 2002. Frequency evaluation in exponential fitting multistep algorithms for ODEs. *J. Comp. Appl. Math. 140*, 423–434.

IXARU, L. GR., VANDEN BERGHE, G., AND DE MEYER, H. 2003. Exponential fitted variable two-step BDF algorithm for first order ODEs. *J. Comp. Appl. Math. 150*, 116–128.

ROBERTSON, H. H. 1966. *The Solution of a Set of Reaction Rate Equations in Numerical Analysis: An Introduction.* Academic Press, London, UK, 178–182.

SIMOS, T. E. 2002. Numerical methods for 1D, 2D and 3D differential equations arising in chemical problems. *Proc. Roy. Soc. Chem. 2*, 170–270.

SIMOS, T. E. 2003. Dissipative trigonometrically fitted methods for the numerical solution of orbital problems. *New Astron. 9*, 59–68.

SIMOS, T. E. AND VIGO-AGUIAR, J. 2001. An exponentially-fitted high order method for long-term integration of periodic initial-value problems. *Comp. Phys. Commun. 140*, 358–365.

VLACHOS, D. S. AND SIMOS, T. E. 2004. Partitioned linear multistep method for long-term integration of the N-body problem. *Appl. Numer. Anal. Computat. Math. 3*, 540–546.

Received October 2007; revised July 2008 and July 2009; accepted August 2009