Local Linearization Method in the Integration of Multibody Equations

Gibin Gil, Ricardo G. Sanfelice, Parviz E. Nikravesh

The University of Arizona, Department of Aerospace and Mechanical Engineering, Tucson AZ 85721-0119,
{gil,sricardo,pen}@email.arizona.edu

Abstract

Computational efficiency of solving the dynamics of highly oscillatory systems is an important issue due to the requirement of small step size of explicit numerical integration algorithms. A system is considered to have an oscillatory solution if it contains a fast solution that varies regularly about a slow solution. This paper investigates the use of the so-called Local Linearization Method (LLM) in the integration of multibody equations of motion that exhibit oscillatory behavior. The LLM is an exponential method that is based on the piecewise linear approximation of the equations through a first-order Taylor expansion at each time step, where the solution at the next time step is determined by the analytic solution of the approximated linear system. In this paper the LLM is applied to simple examples. The results show that the LLM can improve computational efficiency, without jeopardizing the accuracy, when the multibody system is highly oscillatory.

Keywords: *Highly oscillatory system, Numerical integration, Local Linearization Method, Local error estimation*

1 Introduction

One challenging problem in the numerical integration of ordinary differential equations is dealing with highly oscillatory systems. A system will be referred to as highly oscillatory if the time scale of the fast solution is much shorter than the interval of integration [1]. This type of problem arises in many scientific and engineering applications such as simulations of multibody dynamics with rigid and deformable bodies [2], electric networks [3], and molecular dynamics [4, 5]. Standard explicit numerical methods require a small step size to satisfy the absolute stability condition for the fast solution. The computational cost of solving the entire system is dictated by the subsystem with the fast time scale. The limitation of step size can be relaxed if implicit numerical methods are employed. However, iterative computation is required for implicit methods to solve a system of nonlinear equations at every time step. One possible approach to improve this problem is to apply the so-called local linearization method (LLM).

The LLM was initially suggested as an alternative to Runge-Kutta methods for achieving real-time simulation of time-invariant systems [6]. The application of the LLM is not limited to time-invariant systems. It can also be used for general non-autonomous ODEs [7, 8]. The LLM is an exponential method that is based on the piecewise linear approximation of the system's equation through a first-order Taylor expansion at each time step. The numerical solution is determined by the analytic solution of the approximated linear system.

For the non-autonomous differential equation $\dot{y} = f(t, y)$, the LLM [8] can be represented as the following equation

$$\hat{y}_{n+1} = \hat{y}_n + \phi(t_n, \hat{y}_n; h_n), \quad \hat{y}_n \in \mathbb{R}^m \tag{1}$$

starting with $y_0 = y(t_0)$, where

$$\phi(t_n, \hat{y}_n; h_n) = \int_0^{h_n} e^{f_y(t_n, \hat{y}_n)(h_n - u)} \left(f(t_n, \hat{y}_n) + f_t(t_n, \hat{y}_n)u\right) du, \tag{2}$$

and $h_n = t_{n+1} - t_n$. Here, f_y denotes the partial derivatives of f with respect to the variables y, namely $\dfrac{\partial f}{\partial y}$, and f_t denotes the partial derivatives of f with respect to time t, namely $\dfrac{\partial f}{\partial t}$. Equation (2) can be evaluated by applying Theorem 1 in [9]. This theorem proves that integrals involving matrix exponentials can be computed by the exponential of a block triangular matrix. More precisely, if the $m \times m$ block triangular matrix C is defined by

$$C = \begin{bmatrix} A_1 & B_1 & C_1 & D_1 \\ 0 & A_2 & B_2 & C_2 \\ 0 & 0 & A_3 & B_3 \\ 0 & 0 & 0 & A_4 \end{bmatrix} \tag{3}$$

then for $t > 0$

$$e^{Ct} = \begin{bmatrix} F_1(t) & G_1(t) & H_1(t) & K_1(t) \\ 0 & F_2(t) & G_2(t) & H_2(t) \\ 0 & 0 & F_3(t) & G_3(t) \\ 0 & 0 & 0 & F_4(t) \end{bmatrix} \tag{4}$$

where

$$F_j(t) = e^{A_j t} \qquad j = 1,2,3,4$$
$$G_j(t) = \int_0^t e^{A_j(t-s)} B_j e^{A_{j+1}s} ds \qquad j = 1,2,3$$
$$H_j(t) = \int_0^t e^{A_j(t-s)} C_j e^{A_{j+2}s} ds + \int_0^t \int_0^s e^{A_j(t-s)} B_j e^{A_{j+1}(s-r)} B_{j+1} e^{A_{j+2}r} dr ds \quad j = 1,2$$

and

$$K_1(t) = \int_0^t e^{A_1(t-s)} D_1 e^{A_4 s} ds + \int_0^t \int_0^s e^{A_1(t-s)} \left[C_1 e^{A_3(s-r)} B_3 + B_1 e^{A_2(s-r)} C_2 \right] e^{A_4 r} dr ds$$
$$+ \int_0^t \int_0^s \int_0^r e^{A_1(t-s)} B_1 e^{A_2(s-r)} B_2 e^{A_3(r-w)} B_3 e^{A_4 w} dw dr ds$$

To use Theorem 1 in [9] to evaluate Eq. (2), note that Eq (2) can be rewritten in the form of H_1 as

$$\phi(t_n, \hat{y}_n; h_n) = \int_0^{h_n} e^{f_y(t_n, \hat{y}_n)(h_n - u)} f(t_n, \hat{y}_n) du + \int_0^{h_n} \int_0^u e^{f_y(t_n, \hat{y}_n)(h_n - u)} f_t(t_n, \hat{y}_n) dv du \tag{5}$$

Then, Eq. (5) can be obtained from

$$e^{C_n h_n} = \begin{bmatrix} F(t_n, \hat{y}_n; h_n) & g_1(t_n, \hat{y}_n; h_n) & \phi(t_n, \hat{y}_n; h_n) \\ 0 & 1 & g_2(t_n, \hat{y}_n; h_n) \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where

$$C_n = \begin{bmatrix} f_y(t_n, \hat{y}_n) & f_t(t_n, \hat{y}_n) & f(t_n, \hat{y}_n) \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(m+2)\times(m+2)}.$$

An advantage of the LLM can be seen in the analysis of absolute stability. For a given numerical method, the region of absolute stability is the region of the complex ξ-plane such that applying the method for the test equation $\dot{y} = \lambda y$, with $\xi = h\lambda$ from within this region, yields an approximate solution satisfying the absolute stability requirement, namely $|y_{n+1}| \leq |y_n|$ [10]. If the absolute stability region of a numerical method covers the whole left half ξ-plane, the numerical method is said to be A-stable method. For the test equation $\dot{y} = \lambda y$, the LLM gives an exact solution regardless of the step size h. Thus, the LLM is an explicit A-stable method. It should be noted that there is no explicit Runge-Kutta method satisfying A-stable property.

In this paper, the use of the LLM in the integration of multibody systems is investigated. To this end, a simple example of a double pendulum is considered and the numerical solutions are obtained using the LLM and several conventional integration methods. Then, computation time and constraint violations are compared for these numerical methods. Multibody systems with contact conditions are also a possible application area of the LLM. This paper considers a simple contact problem consisting of two bouncing balls and applies the LLM to integrate the equations of motion. Adaptive step size control is a common feature of conventional numerical integration methods. To apply adaptive step size control, there should be a way of estimating the local error introduced by the numerical method. This paper proposes a local error estimation method for the LLM and utilizes it for adaptive step size control.

2 Application of LLM in Multibody Systems

Unlike other explicit numerical integration methods, the LLM requires the system Jacobian $\partial f/\partial y$ of the differential equation $\dot{y} = f(t, y)$ to be integrated. This section presents an explicit expression of $\partial f/\partial y$ for general multibody

systems with constraints. Then, it demonstrates the application of the LLM to constrained multibody systems through an example of a double pendulum.

Let us consider the equations of motion for a multibody system with constraints as

$$\mathbf{M}\ddot{\mathbf{q}} - \mathbf{D}^\top \boldsymbol{\lambda} = \mathbf{h} \tag{7}$$

where $\ddot{\mathbf{q}}$ denotes the accelerations of bodies, $\mathbf{D}$ denotes the Jacobian of constraints, $\mathbf{M}$ denotes the mass matrix, and $\mathbf{h}$ denotes the force array. The constraints in acceleration level are expressed as

$$\mathbf{D}\ddot{\mathbf{q}} = \boldsymbol{\gamma}, \qquad \boldsymbol{\gamma} = -\dot{\mathbf{D}}\dot{\mathbf{q}} \tag{8}$$

The two sets of equations can be appended to obtain the solution for $\ddot{\mathbf{q}}$ as

$$\begin{bmatrix} \mathbf{M} & -\mathbf{D}^\top \\ \mathbf{D} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ \boldsymbol{\gamma} \end{bmatrix} \tag{9}$$

The integration variable y is defined as $y = [\mathbf{q}, \dot{\mathbf{q}}]^\top$ and the differential equation to be integrated is $\dot{y} = f(t, y) = [\dot{\mathbf{q}}, \ddot{\mathbf{q}}]^\top$. The system Jacobian $\partial f/\partial y$ has the following form:

$$\frac{\partial f}{\partial y} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{q}} & \frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} \end{bmatrix} \tag{10}$$

By taking the partial derivatives of Eq. (7), we can obtain the expressions of $\partial\ddot{\mathbf{q}}/\partial\mathbf{q}$ and $\partial\ddot{\mathbf{q}}/\partial\dot{\mathbf{q}}$ as

$$\frac{\partial \ddot{\mathbf{q}}}{\partial \mathbf{q}} = \mathbf{M}^{-1}\left[\left(\frac{\partial \mathbf{D}}{\partial \mathbf{q}}\right)^\top \boldsymbol{\lambda} + \mathbf{D}^\top \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{q}} + \frac{\partial \mathbf{h}}{\partial \mathbf{q}} - \frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}}\right] \tag{11}$$

$$\frac{\partial \ddot{\mathbf{q}}}{\partial \dot{\mathbf{q}}} = \mathbf{M}^{-1}\left(\mathbf{D}^\top \frac{\partial \boldsymbol{\lambda}}{\partial \dot{\mathbf{q}}} + \frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}}\right) \tag{12}$$

The expression of $\boldsymbol{\lambda}$ is obtained by combining Eqs. (7) and (8) as

$$\left(\mathbf{D}\mathbf{M}^{-1}\mathbf{D}^\top\right)\boldsymbol{\lambda} = \boldsymbol{\gamma} - \mathbf{D}\mathbf{M}^{-1}\mathbf{h} \tag{13}$$

Taking the partial derivatives of Eq. (13) yields the expressions of $\partial\boldsymbol{\lambda}/\partial\mathbf{q}$ and $\partial\boldsymbol{\lambda}/\partial\dot{\mathbf{q}}$ as

$$\frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{q}} = \left(\mathbf{D}\mathbf{M}^{-1}\mathbf{D}^\top\right)^{-1}\left[\frac{\partial \boldsymbol{\gamma}}{\partial \mathbf{q}} - \frac{\partial \mathbf{D}}{\partial \mathbf{q}}\ddot{\mathbf{q}} - \mathbf{D}\mathbf{M}^{-1}\frac{\partial \mathbf{h}}{\partial \mathbf{q}} - \mathbf{D}\mathbf{M}^{-1}\frac{\partial \mathbf{D}^\top}{\partial \mathbf{q}}\boldsymbol{\lambda} + \mathbf{D}\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \mathbf{q}}\ddot{\mathbf{q}}\right] \tag{14}$$

$$\frac{\partial \boldsymbol{\lambda}}{\partial \dot{\mathbf{q}}} = \left(\mathbf{D}\mathbf{M}^{-1}\mathbf{D}^\top\right)^{-1}\left[\frac{\partial \boldsymbol{\gamma}}{\partial \dot{\mathbf{q}}} - \mathbf{D}\mathbf{M}^{-1}\frac{\partial \mathbf{h}}{\partial \dot{\mathbf{q}}}\right] \tag{15}$$

In order to demonstrate the application of LLM to constrained multibody systems, a simple example of a double pendulum, as shown in Fig. 1, is considered. This system contains a very stiff torsional spring of the coefficient $k = 3000Nm/deg$ between its two links resulting into a highly oscillatory system. The equations of motion are constructed by the body-coordinate formulation [11] and the system Jacobian $\partial f/\partial y$ is evaluated by Eq. (10). The numerical solutions are obtained using LLM and conventional integration methods; ODE45[1], ODE15s[2], and ODE23s[3] [12]. ODE15s and ODE23s are implicit integration methods for stiff systems. For those methods, the expression of $\partial f/\partial y$ is provided for better efficiency. Figure 2 shows the numerical solution obtained by the LLM and compares the total energy variation, constraint violation and computation time from those numerical methods. The rotation angle of the lower link changes at high frequency due to the stiff torsional spring. As a result the system shows the characteristic of a highly oscillatory system. In the case of the LLM, a uniform step size $h = 0.02$ is used. From Fig. 2 it can be seen that the LLM causes the similar level of total energy variation and less constraint violation compared to other methods with this choice of step size. Furthermore, it turns out that the LLM takes less computation time than other methods. Specifically, the computation time of the LLM is about 40% less than that of ODE45. The better numerical efficiency can be achieved because a larger step size can be used while maintaining comparable accuracy.

[1]Non-stiff differential equations solver with variable step size, based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair

[2]Stiff differential equations solver, variable order method based on the numerical differentiation formulas (NDFs)

[3]Stiff differential equations solver, based on a modified Rosenbrock formula of order 2

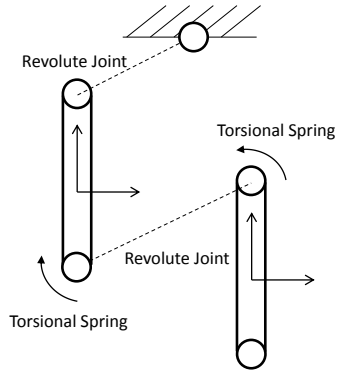


Figure 1. Configuration of a double pendulum

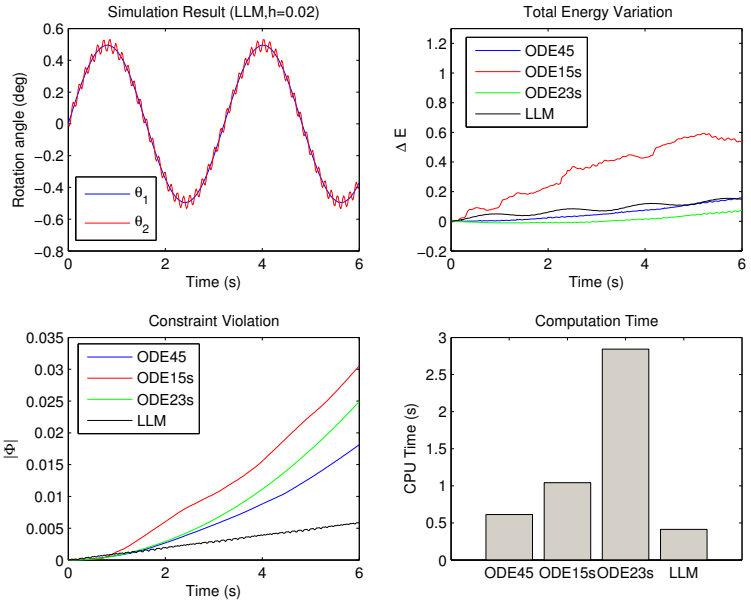


Figure 2. Comparison of simulation results

Despite the high frequency dynamics of the system, the LLM can solve this system with fairly large step size $h = 0.02$, because the oscillatory behaviour can be described well by the matrix exponential of $\partial f/\partial y$. The exponential of λt, where λ is an eigenvalue of $\partial f/\partial y$, can be written as

$$e^{\lambda t} = e^{Re(\lambda)t}\left(\cos(Im(\lambda)t) + j\sin(Im(\lambda)t)\right) \tag{16}$$

and, hence, the matrix exponential of system Jacobian contains the oscillating characteristic of the system. However, the LLM should take very small step size when the system contains severe nonlinearity since the local linear approximation is valid only for small period of time. In multibody systems, the nonlinearity normally comes from the trigonometric functions, which do not have severe nonlinearity. Therefore, the LLM may be useful for solving a multibody system which has eigenvalues of large imaginary part and has mild nonlinearity.

3 Application of LLM in Contact Problems

The penalty method is a popular methodology to model the contact phenomenon in multibody systems. The method introduces a fictitious spring of high stiffness to impose the non-penetration condition. Quite often, conventional numerical methods have difficulty in dealing with the fictitious spring due to its high stiffness. In this section, the LLM is applied to a simple contact problem and its performance is compared to that of other conventional methods.

Let us consider the two bouncing balls moving in vertical direction under gravity as shown in Fig. 3. The contact between the balls and between the ball and the ground are modeled by penalty method as shown in Fig. 4. The spring coefficient is chosen to be $10^5 N/m$. Although the equations of motion are linear during contact and non-contact periods, the system Jacobian changes significantly at the instant that bodies come into or come out of contact. Thus, time step should be adjusted such that the new step to start exactly when the contact begins or ends. The non-penetration condition for contact can be expressed as

$$\Psi(y(t)) \leq 0 \tag{17}$$

for some appropriately defined function Ψ capturing the distance between bodies. When the numerical solution $\hat{y}_{n+1}$ is obtained, it is checked if $sign\left(\Psi(\hat{y}_{n+1})\right) = sign\left(\Psi(\hat{y}_n)\right)$. When the signs are different, it indicates that bodies come into or out of contact between t_n and t_{n+1}. The contact instant t^* can be determined by solving the nonlinear equation $\Psi(\hat{y}(t)) = 0$. The Newton-Raphson or Bi-section method can be used for solving this nonlinear equation. $\Psi(\hat{y}(t))$ can be evaluated without additional numerical error from interpolation since the LLM formula gives the numerical solution $\hat{y}(t)$ for $t \in [t_n, t_{n+1}]$ as

$$\hat{y}(t) = \hat{y}_n + \phi(t_n, \hat{y}_n; t - t_n) \quad \text{for } t \in [t_n, t_{n+1}] \tag{18}$$

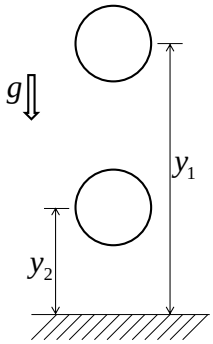
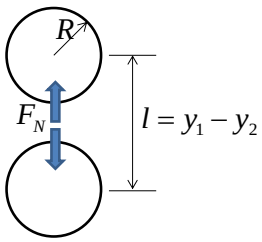


Figure 3. Two bouncing balls



Penetration depth

$$\delta = 2R - l$$

Contact force

$$F_N = \begin{cases} K\delta & if \quad \delta > 0 \\ 0 & if \quad \delta \leq 0 \end{cases}$$

Figure 4. Contact model using a fictitious spring

Once t^* is determined, the time step t_{n+1} is reset to t^*. This process is repeated until the end of the time domain. If the system has multiple contacts, the contact instant for all possible contact should be determined and the step size should be adjusted to the one that occurs first. For the two bouncing balls, the function $\Psi(y)$ is defined as

$$\Psi_1 = 2R - (y_1 - y_2), \quad \Psi_2 = R - y_2 \tag{19}$$

The contact instant t_1^* and t_2^* should be determined as mentioned above for each contact condition, respectively. Then, the time step t_{n+1} is reset to $t^* = \min(t_1^*, t_2^*)$. The state variable of the system can be written as

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} \tag{20}$$

Assuming that both bodies have unit mass, the dynamics of the system is given by

$$\dot{y} = \begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ -9.81 + F_1 \\ -9.81 - F_1 + F_2 \end{bmatrix} \tag{21}$$

where

$$F_1 = \begin{cases} -K\,(y_1 - y_2 - 2R) & \text{if } \Psi_1 > 0 \\ 0 & \text{if } \Psi_1 \leq 0 \end{cases}$$

$$F_2 = \begin{cases} -K\,(y_2 - R) & \text{if } \Psi_2 \geq 0 \\ 0 & \text{if } \Psi_2 \leq 0 \end{cases}$$

The dynamic simulation of the two bouncing balls is performed with ODE45, ODE23s, and the LLM. Figure 5 shows the position of the first ball with respect to time and the total energy variation of the system. The result of ODE45 is quite different from the results of other two numerical methods. The result of ODE45 contains significant error which can be observed in the plot of total energy variation. The total energy of the system should be constant since there is no damping. The total energy of the numerical solution by the LLM is well conserved. With the step size adjustment, the LLM can solve this problem without error because the dynamics of the system is piecewise-linear. The result in Fig. 5 shows that ODE23s and the LLM give similar results in terms of the displacement. However, the computation time of the LLM is much smaller than that of ODE23s. These results demonstrate that the LLM can be a good option for solving contact problems when the penalty method is employed.

4 Local Error Estimation for LLM

Adaptive step size selection schemes are common feature of conventional numerical integration methods. Such schemes change the step size h_n at each time step instead of using a constant step size h, so that the number of steps required

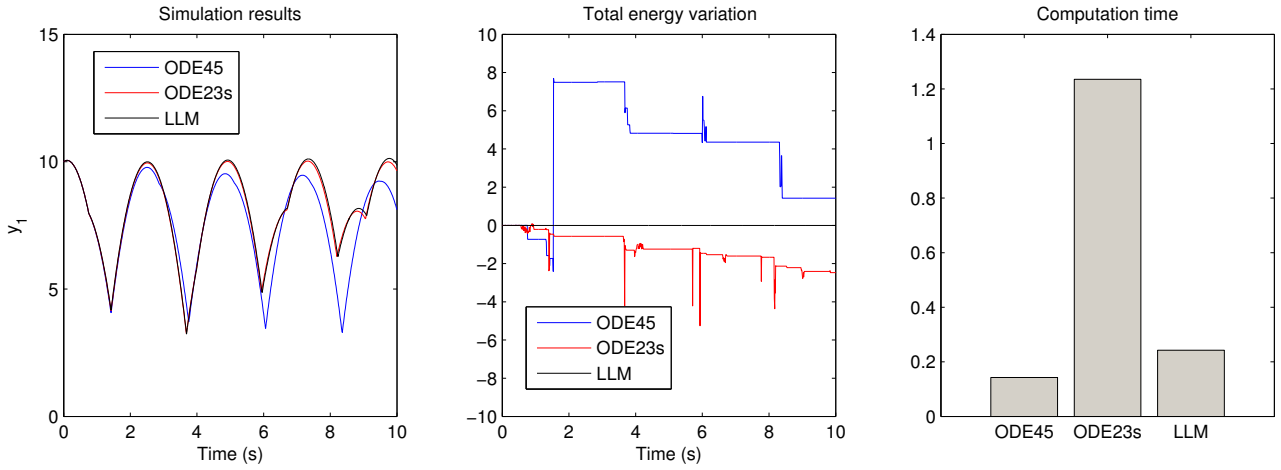


Figure 5. Comparison of simulation results for two bouncing balls

to solve the ODEs are reduced, in turn, improving performance. The step size h_n should be determined such that the local error of numerical solution is maintained within a given tolerance. To this end, a way of estimating the local error introduced by the numerical method at each time step is essential. In this section, a model providing an estimate of the local error of the LLM is proposed and validated by numerical examples. For simplicity, we consider the autonomous ordinary differential equation

$$\dot{y} = f(y), \qquad y(t_n) = \hat{y}_n \tag{22}$$

The linear approximation of Eq. (22) is given as

$$\dot{\hat{y}} = g(\hat{y}) = f(\hat{y}_n) + \frac{\partial f}{\partial y}\Big|_{y=\hat{y}_n} (\hat{y} - \hat{y}_n), \qquad A_n \equiv \frac{\partial f}{\partial y}\Big|_{y=\hat{y}_n} \tag{23}$$

We define the local error as $E = y - \hat{y}$. Then, the dynamics of E can be expressed as

$$\dot{E} = f(y) - g(\hat{y}) = f(y) - g(y) + g(y) - g(\hat{y}) = r + A_n E \tag{24}$$

where

$$r = f(y) - g(y)$$

The solution for this system from $E(0) = 0$ can be expressed as

$$E(t) = \int_{t_n}^{t} e^{A_n(t-\tau)} r(\tau) d\tau \qquad \text{for } t \geq t_n \tag{25}$$

where

$$r(t) = f(y(t)) - g(y(t))$$

We need the expression for $r(t)$ to estimate the local error. Let us consider the Taylor series of $r(t)$

$$r(t) = r(t_n) + \frac{dr}{dt}\Big|_{t=t_n} (t - t_n) + \frac{d^2 r}{dt^2}\Big|_{t=t_n} (t - t_n)^2 + HOT \tag{26}$$

By the definition of $r(t)$ we have, using Eq. (23),

$$r(t_n) = f(y(t_n)) - g(y(t_n)) = f(\hat{y}_n) - f(\hat{y}_n) = 0 \tag{27}$$

$$\frac{dr}{dt}\Big|_{t=t_n} = \frac{df(y)}{dt}\Big|_{t=t_n} - \frac{dg(y)}{dt}\Big|_{t=t_n} = \frac{\partial f}{\partial y}\Big|_{t=t_n} \dot{y}(t_n) - \frac{\partial f}{\partial y}\Big|_{t=t_n} \dot{y}(t_n) = 0 \tag{28}$$

So, $r(t)$ can be approximated around $t = t_n$ as a quadratic function by neglecting the higher order terms, namely

$$r(t) \approx v_n \, (t - t_n)^2, \qquad v_n = \frac{d^2 r}{dt^2}\Big|_{t=t_n} \tag{29}$$

Then, using Eq. (29), we approximate v_n as

$$v_n = \frac{r(t_{n+1})}{(t_{n+1}-t_n)^2} = \frac{f(y(t_{n+1})) - g(y(t_{n+1}))}{(t_{n+1}-t_n)^2} \approx \frac{f(\hat{y}_{n+1}) - g(\hat{y}_{n+1})}{(t_{n+1}-t_n)^2} \tag{30}$$

The local error between t_n and t_{n+1} is obtained as

$$l_{n+1} = E(t_{n+1}) = \int_{t_n}^{t_{n+1}} e^{A_n(t_{n+1}-\tau)} v_n \cdot (\tau - t_n)^2 d\tau = \int_0^{h_n} e^{A_n(h_n-u)} v_n u^2 du = \int_0^{h_n}\int_0^{u}\int_0^{r} e^{A_n(h_n-u)} 2v_n ds dr du \tag{31}$$

The integral in Eq. (31) can be computed by using the Eq. (4) as

$$e^{h_n L} = \begin{bmatrix} e^{h_n A_n} & G_1 & H_1 & l_{n+1} \\ 0 & I & h_n & \frac{1}{2}h_n^2 \\ 0 & 0 & I & h_n \\ 0 & 0 & 0 & I \end{bmatrix} \tag{32}$$

where

$$L = \begin{bmatrix} A_n & 2v_n & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{(m+3)\times(m+3)},\ H_1 = \int_0^{h_n} 2v_n e^{A_n(t-s)} s ds,\ G_1 = \int_0^{h_n} 2v_n e^{A_n(t-s)} ds \tag{33}$$

When the state variable y is scalar, the integral of Eq. (31) can be computed analytically as

$$l_{n+1} = \int_0^{h_n} e^{A_n(h_n-u)} v_n u^2 du = \frac{2v_n}{A_n^3}\left(e^{A_n h_n} - 1 - A_n h_n - \frac{(A_n h_n)^2}{2}\right) = \frac{v_n}{3A_n^3}(A_n h_n)^3 + O(h_n^4) \tag{34}$$

This result indicates that the order of convergence of the LLM is two, which is consistent with the result of [8].

To check the validity of the proposed local error estimate model, we consider the differential equation

$$\dot{x}_1 = -\frac{x_2}{x_3^2}, \quad \dot{x}_2 = \frac{x_1}{x_3^2}, \quad \dot{x}_3 = 1 \tag{35}$$

with an initial condition

$$x\left(1/\pi\right) = \left[0, -1, 1/\pi\right]^\top$$

The analytic solution for this ODEs is given as

$$x(t) = \left[\sin\left(1/t\right),\ \cos\left(1/t\right),\ t\right]^\top \tag{36}$$

The differential equation (35) is solved with the LLM for one time step and the local error is estimated using the proposed local error estimate model. The estimated local error and the exact local error are compared for the various values of h in Fig. 6. This figure shows that the proposed local error estimation method can provide a very good estimate for the local error. The proposed local error estimation model can be used for any ordinary differential equation and provides the local error for each state variable. Thus, we can specify the local error tolerance for individual state variable if desired.

5 Adaptive step size Control

Using the proposed estimation model, the adaptive step size control is implemented so that the step size can be adjusted adaptively and the computational efficiency can be maximized while satisfying the accuracy requirement. If the order of the numerical integration method is p, the local error and the step size have an asymptotic relationship

$$l_{n+1} = \Phi(t_n, y_n) h_n^{p+1} + O(h_n^{p+2}) \tag{37}$$

where Φ is the principal error function [13]. Based on this relationship, a standard algorithm for step size control is given as [14]

$$h^* = \gamma \left(\frac{tol}{r_{n+1}}\right)^{1/k} h_n \tag{38}$$

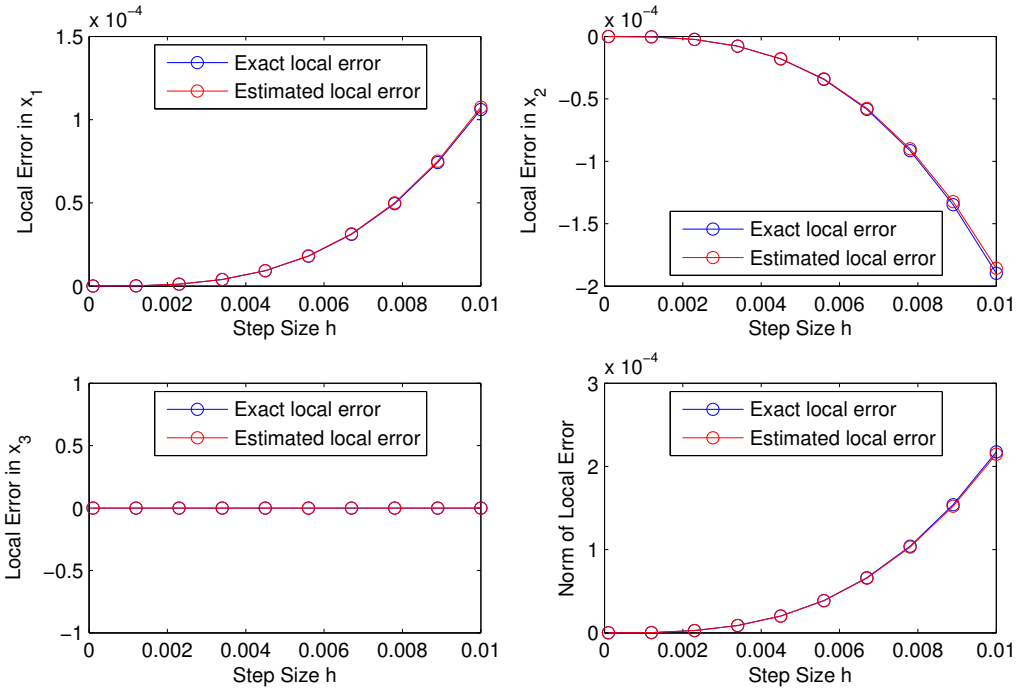


Figure 6. Comparison of exact and estimated local error

where h_n is the step size, h^* is the new step size, tol is the tolerance specified by user, k is a positive parameter of the algorithm, and $\gamma \in (0, 1)$ is a safety factor for reducing the risk of a rejected step. Conventionally, the estimated error r is defined as

$$r_{n+1} = \begin{cases} \|l_{n+1}\|, & \text{error per step (EPS)} \\ \|l_{n+1}\|/h_n, & \text{error per unit step (EPUS)} \end{cases} \tag{39}$$

The reason to use EPUS is to have the same accumulated global error for the same integration time regardless of the number of steps taken [14]. The parameter k is determined as $k = p + 1$ (for EPS) or $k = p$ (for EPUS), where p is the order of the numerical integration method. At every time step, the local error of the numerical solution is estimated and the condition $r_{n+1} \leq tol$ is checked. If this condition is not satisfied then the obtained numerical solution is rejected and the new solution is computed again with a smaller step size. The new step size h^* is determined by Eq. (38) and this process is repeated until the condition $r_{n+1} \leq tol$ is satisfied. Once this condition is satisfied, the step size for the next step is set as $h_{n+1} = h^*$ and the numerical method proceeds to the next step.

This algorithm is applied to the numerical solution of the double pendulum problem discussed in section 2. We use the EPUS definition of r_{n+1} and the parameter k of Eq. (38) is set as $k = 2$ since the order of the LLM is two. The safety factor γ is set to be 0.9. The numerical solution is obtained by the LLM with the tolerance specified as $tol = 0.1$ and $tol = 0.05$, respectively. Figure 7 shows the change of step size and the estimated error for the given tolerances. The result indicates that the smaller step size is taken and the number of steps increases as the more strict tolerance is imposed. The plot of the estimated error confirms that the error is maintained within the specified value. With the help of the adaptive step size scheme, the user does not need to choose the step size by trial-and-error and the step size is chosen automatically as large as possible while the error tolerance is satisfied.

In this research, an elementary algorithm for step size control as in Eq. (38) is considered. There were some attempts to apply other technique of control theory to adaptive step size selection [14, 15]. In [15], the PI control strategy was used for the step size selection in explicit Runge-Kutta methods. The PI controller takes the form:

$$h_{n+1} = \left(\frac{tol}{r_{n+1}}\right)^{k_I} \left(\frac{r_n}{r_{n+1}}\right)^{k_P} h_n \tag{40}$$

and it was suggested to overcome the issue of oscillating step size sequence, which often occurs when stability rather than accuracy dictates the step size. This issue can be seen when the system is stiff and the step size yields $\xi = h_n\lambda$ which is located around the boundary of absolute stability region of the numerical method. It was shown that the PI controller

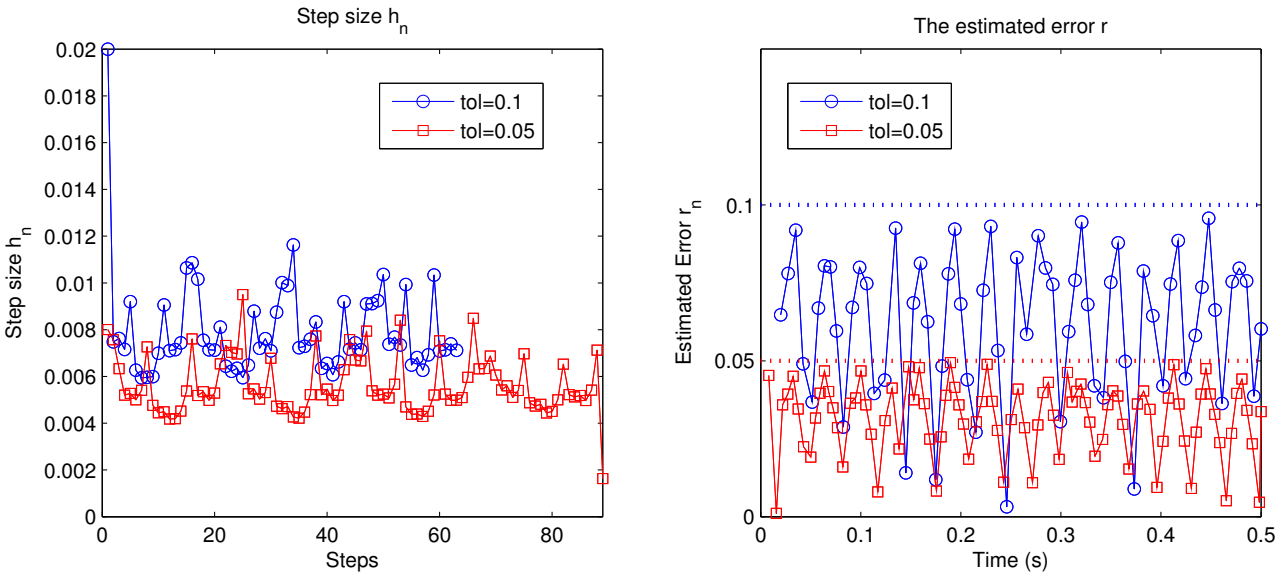


Figure 7. step size and estimated error with $tol = 0.1$ and $tol = 0.05$

can produce a smoother step size sequence and avoid frequent step size rejection. The LLM is an A-stable method and its absolute stability region covers the whole left half ξ-plane. Thus, the LLM does not suffer from the issue of oscillating step size sequence even when very stiff system is handled. For the double pendulum example, the elementary algorithm of Eq. (38) turns out to perform well and the problem of oscillating step size is not observed.

6 Conclusion

The LLM is a numerical integration method that uses the successive linear approximation and the analytic solution form of a linear dynamic system. The LLM is an explicit A-stable method and its good dynamic properties are well known to applied mathematic community, but there have been few attempts to apply it to multibody dynamics.

In this study, the LLM is applied to some examples of multibody systems with kinematic constraints and contact condition. The result shows that the LLM is a good candidate for solving the dynamics of a multibody system which has highly oscillatory behavior because a larger step size can be employed compared to other conventional explicit methods.

This paper also proposes a new local error estimation method for the LLM. It gives an estimate of local error for each state variable that is introduced by local linearization process at each time step. The proposed estimate model can provides not only the indication of numerical error included in the numerical solution by the LLM, but also a basis for the implementation of the adaptive step size control.

Acknowledgments

G. Gil would like to acknowledge the financial support of Hankook Tire in this research project. Research by R. G. Sanfelice has been partially supported by the National Science Foundation under CAREER Grant no. ECS-1150306 and by the Air Force Office of Scientific Research under Grant no. FA9550-12-1-0366.

References

[1] L. R. Petzold, L.O. Jay, and J. Yen. Numerical solution of highly oscillatory ordinary differential equations. *Acta Numerica*, pages 437–483, 1997.

[2] S. S. Shome, E. J. Haug, and L. O. Jay. Dual-rate integration using partitioned runge-kutta methods for mechanical systems with interacing subsystems. *Mechanics Based Design of Structures and Machines*, 32(3):253–282, 2004.

[3] M. Günther and P. Rentrop. Multirate row methods and latency of electric circuits. *Appl. Numer. Math.*, 13:83–102, 1993.

[4] A. Kopf, W. Paul, and B. Dunweg. Multiple time step integrators and momentum conservation. *Computer Physics Communications*, 101:1–8, 1997.

[5] M. E. Tuckerman and B. J. Berne. Molecular dynamics in systems with multiple time scales: Systems with stiff and soft degrees of freedom and with short and long range forces. *J. Chem. Phys.*, 95(11):8362–8364, 1991.

[6] G. Cook and C. F. Lin. Comparison of a local linearization algorithm with standard numerical integration methods for real-time simulation. *IEEE Transactions on Industrial Electronics and Control Instrumentation*, IECI-27, 1980.

[7] J. C. Jimenez, R. Biscay, C. Mora, and L. M. Rodriguez. Dynamic properties of the local linearization method for initial-value problems. *Applied Mathematics and Computation*, 126:63–81, 2002.

[8] J. C. Jimenez and F. Carnodell. Rate of convergence of local linearization schemes for initial-value problems. *Applied Mathematics and Computation*, 171:1282–1295, 2005.

[9] C Van Loan. Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404, 1978.

[10] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.

[11] P. E. Nikravesh. *Planar Multibody Dynamics*. CRC Press, 2008.

[12] MATLAB. *version 7.13.0 (R2011b)*. The MathWorks Inc., Natick, Massachusetts, 2011.

[13] S. Fatunla. *Numerical methods for initial value problems in ordinary differential equations*. Academic Press, Boston, 1988.

[14] K. Gustafsson. Control theoretic techniques for stepsize selection in explicit runge-kutta methods. *ACM Transactions on Mathematical Software*, 17(4):533–554, Dec 1991.

[15] K. Gustaffsson, M. Lundh, and G. Söderlind. A pi stepsize control for the numerical solution of ordinary differential equations. *BIT*, 28(2):270–287, 1988.