Efficient symplectic-energy-momentum integration of Hamiltonian dynamical systems

Yosi Shibberu
Mathematics Department
Rose-Hulman Institute of Technology
Terre Haute, IN 47803
www.rose-hulman.edu/∼shibberu/DTH_Dynamics/DTH_Dynamics.htm

Abstract— *The implicit equations of a symplectic-energy-momentum integrator are not easily solved, especially for small time steps. An inefficient, nested iteration scheme has typically been used to solve these equations. We describe new, more efficient, iteration schemes which avoid nested iterations. We present simulation results comparing five different second-order, integration methods for two different types of three-body trajectories. Symplectic-energy-momentum integration is shown to be as efficient as the leapfrog method for one of these trajectories.*

Keywords: symplectic, energy, momentum, integration, Hamiltonian, dynamics.

1. Introduction

Hamiltonian dynamical systems arise in a wide variety of applications ranging in scope from quantum mechanics to optimal control theory. Numerical integration methods which preserve the special properties of Hamiltonian dynamical systems are, therefore, of considerable interest.

A symplectic-energy-momentum (SEM) integrator is a special type of numerical integrator (differential equations solver) which preserves the following key properties of Hamiltonian systems: i) symplecticness ii) conservation of energy iii) conservation of momentum.

The importance of preserving symplecticness can be explained by the following property of symplectic integrators: *If a numerical integration method is symplectic, then it can be derived from a discrete variational principle [1].* The discrete variational principles of symplectic integrators give symplectic integrators, special, coordinate invariant (geometric) properties. In fact, symplectic integrators behave like Hamiltonian dynamical systems, each described by a Hamiltonian function [2].

The term "symplectic-energy-momentum" integrator was coined by Kane, Marsden and Ortiz [3]. My own work on SEM integration—known as discrete-time Hamiltonian (DTH) dynamics [4]—predates the work of Kane, et al. DTH dynamics originated from my effort to obtain the exact energy and momentum conserving properties of the discrete mechanics of Greenspan [5], [6], from the variational principle used by Lee [7], [8] in his discrete mechanics.

The equations of a SEM integrator are implicitly defined and difficult to solve, especially for small time steps. Until recently, I have been using a computationally expensive, nested, Newton iteration scheme. In this article, I describe a new iteration scheme which is nearly six times faster. I present simulation results comparing SEM integration using this new iteration scheme to four other, second-order methods: 1) the widely used leapfog method, 2) the explicit second-order Runge-Kutta method, 3) the implicit midpoint method and 4) the discrete mechanics method of Greenspan. Table 1 lists the special properties of each of the methods compared in this article.

Property	Method				
	lf	rk2	mid	dmg	dth
applicable to general Hamiltonians		⋆	⋆		⋆
explicit	⋆	⋆			
variable time step					⋆
symplectic	⋆		⋆		⋆
conserves energy				⋆	⋆
conserves angular momentum	⋆		⋆	⋆	⋆
conserves linear momentum	⋆	⋆	⋆	⋆	⋆

lf	—	leapfrog method
rk2	—	second-order Runge-Kutta method
mid	—	implicit midpoint method
dmg	—	discrete mechanics of Greenspan
dth	—	DTH dynamics

Table 1
PROPERTIES OF THE METHODS COMPARED IN THIS ARTICLE.

2. Foundations of DTH Dynamics

In this section, I derive the DTH equations of SEM integration, equations (3)–(4), from the *DTH Principle of Stationary Action*, the variational principle that defines DTH dynamics. We begin with preliminary definitions.

Let $H(t, q_1, \ldots, q_n, p_1, \ldots, p_n)$ be the Hamiltonian function of an n degree-of-freedom Hamiltonian dynamical system. (Often, the Hamiltonian function is simply the energy function of a dynamical system.) Let $\mathsf{q} = (q_1, \ldots, q_n, t)$ be the position coordinates and $\mathsf{p} = (p_1, \ldots, p_n, \wp)$ be the momentum coordinates in extended phases space where the time, t, and the momentum conjugate to time, $\wp$, are on an equal footing with the other coordinates. Let $\mathsf{z}_k = (\mathsf{q}_k, \mathsf{p}_k)$ be a sequence of points in extend phase space which are the vertices of a piecewise-linear, continuous trajectory. (See Figure 1.) Let $\Delta\mathsf{q}_k = \mathsf{q}_{k+1} - \mathsf{q}_k$ and $\Delta\mathsf{p}_k = \mathsf{p}_{k+1} - \mathsf{p}_k$. Define the *one-step action* of a discrete-time Hamiltonian

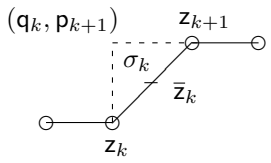

Fig. 1
A PIECEWISE-LINEAR, CONTINUOUS TRAJECTORY.

dynamical system to equal the extended-phase space function $\mathcal{A}(\mathsf{q}_{k+1}, \mathsf{p}_k)$ where

$$\mathcal{A}(\mathsf{q}_{k+1}, \mathsf{p}_k) \quad = \quad \frac{1}{2}\Delta\mathsf{q}_k^\top \Delta\mathsf{p}_k. \tag{1}$$

(It is possible to show that the one-step action $\mathcal{A}(\mathsf{q}_{k+1}, \mathsf{p}_k)$ equals the *symplectic area* of the triangle shown in Figure 1.) Let $\mathcal{H}(\mathsf{z})$ be an extended-phase space Hamiltonian function. (Typically, we define $\mathcal{H}(\mathsf{z})$ to equal $H(t, q_1, \ldots, q_n, p_1, \ldots, p_n) + \wp$.) Define $\bar{\mathsf{z}}_k = \frac{1}{2}(\mathsf{z}_k + \mathsf{z}_{k+1})$. The variational principle used to define DTH dynamics is stated below:

> *DTH Principle of Stationary Action:*
>
> *A DTH trajectory is a piecwise-linear, continuous trajectory determined by the vertices z_k for which the one-step action $\mathcal{A}(\mathsf{q}_{k+1}, \mathsf{p}_k)$ is stationary for variations satisfying the constraint $\mathcal{H}(\bar{\mathsf{z}}_k) = 0$.*

The method of Lagrange multipliers can be used to derive the equations which determine DTH trajectories. Define the Lagrangian function $\mathcal{L}(\mathsf{q}_{k+1}, \mathsf{p}_k, \lambda_k)$ as follows:

$$\mathcal{L}(\mathsf{q}_{k+1}, \mathsf{p}_k, \lambda_k) \quad = \quad \mathcal{A}(\mathsf{q}_{k+1}, \mathsf{p}_k) + \lambda_k \mathcal{H}(\bar{\mathsf{z}}_k). \tag{2}$$

The DTH Principle of Stationary Action implies that the gradient of the Lagrangian function, $\nabla\mathcal{L}$, must equal zero along DTH trajectories. Now, it is possible to show that the gradient of the one-step action, $\nabla\mathcal{A}$, equals $\frac{1}{2}J\Delta\mathsf{z}_k$, where

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix}$$

is the skew-symmetric matrix commonly used in symplectic geometry and I is the $(n+1) \times (n+1)$ identity matrix. Therefore, we have

$$\begin{aligned} \nabla\mathcal{L} \quad &= \quad \begin{pmatrix} \nabla\mathcal{A} + \lambda_k \nabla\mathcal{H} \\ \mathcal{H} \end{pmatrix} \\ &= \quad \begin{pmatrix} \frac{1}{2}J\Delta\mathsf{z}_k + \lambda_k \frac{1}{2}\mathcal{H}_z \\ \mathcal{H} \end{pmatrix}. \end{aligned}$$

Since J^2 equals the negative of the $(2n+2)\times(2n+2)$ identity matrix, the equation $\nabla\mathcal{L} = 0$ implies

$$\Delta\mathsf{z}_k \quad = \quad \lambda_k J\mathcal{H}_\mathsf{z}(\bar{\mathsf{z}}_k) \tag{3}$$

$$\mathcal{H}(\bar{\mathsf{z}}_k) \quad = \quad 0. \tag{4}$$

Equations (3)–(4) are the DTH equations of SEM integration and were first derived in [4]. The proof of the existence and uniqueness of solutions to equations (3)–(4) is nontrivial [4], [9]. I have been able to show that, for sufficiently small time steps, a locally unique solution to the DTH equations always exists for regions of extended phase space where the function

$$\psi(\mathsf{z}) = (J\mathcal{H}_\mathsf{z})^\top \mathcal{H}_{\mathsf{zz}} (J\mathcal{H}_\mathsf{z})$$

is bounded away from zero. DTH trajectories crossing $\psi(\mathsf{z}) = 0$ manifolds need to be regularized in some fashion [9]. A symplectic-energy-momentum preserving, regularization of such trajectories is described in [10].

Until recently, I have been using a nested, Newton iteration scheme to solve equations (3)–(4). In this nested scheme, the function $\bar{\mathsf{z}}_k = \bar{\mathsf{z}}(\mathsf{z}_k, \lambda_k)$ implicitly defined by equation (3) is first evaluated using an inner Newton iteration. Then $\bar{\mathsf{z}}(\mathsf{z}_k, \lambda_k)$ is used in an outer Newton iteration to solve the now decoupled equation (4), i.e. the equation $g(\lambda_k) = \mathcal{H}(\bar{\mathsf{z}}(\mathsf{z}_k, \lambda_k)) = 0$ is solved for λ_k. In the next section, I derive a more efficient iteration scheme which avoids nested interations. I also describe an iteration that does not require computation of the Hessian matrix, $\mathcal{H}_{\mathsf{zz}}$.

3. Efficient Iteration Schemes

In this section, I will use the simplified notation $\mathsf{z} = \bar{\mathsf{z}}_k$, $\mathsf{z}_0 = \mathsf{z}_k$ and $\lambda = \lambda_k$. By taking half a time step, the DTH equations (3)–(4) can be expressed as

$$\mathsf{f}(\mathsf{z}, \lambda) \quad = \quad \mathsf{z} - \mathsf{z}_0 - \frac{1}{2}\lambda J\mathcal{H}_\mathsf{z}(\mathsf{z}) = 0 \tag{5}$$

$$g(\mathsf{z}, \lambda) \quad = \quad \mathcal{H}(\mathsf{z}) = 0 \tag{6}$$

The Jacobian matrix needed for a direct application of Newton's method to equations (5)–(6) is

$$\begin{pmatrix} \mathsf{f}_\mathsf{z} & \mathsf{f}_\lambda \\ g_\mathsf{z}^\top & g_\lambda \end{pmatrix} = \begin{pmatrix} I - \frac{1}{2}\lambda J\mathcal{H}_{\mathsf{zz}} & -\frac{1}{2}J\mathcal{H}_\mathsf{z} \\ \mathcal{H}_\mathsf{z}^\top & 0 \end{pmatrix}. \tag{7}$$

We can show that the Jacobian matrix given by (7) is singular when the time step, λ, equals zero, as follows. Multiplying the first $2n+2$ rows of the Jacobian matrix by $\mathcal{H}_z^\top$ and using the fact that $-\frac{1}{2}\mathcal{H}_z^\top J\mathcal{H}_z = 0$ yields the row-equivalent row

$$\begin{pmatrix} \mathcal{H}_z^\top - \frac{1}{2}\lambda\mathcal{H}_z^\top J\mathcal{H}_{\mathsf{zz}} & 0 \end{pmatrix}. \tag{8}$$

When $\lambda = 0$, (8) is identical to the last row of (7) implying that the Jacobian matrix is singular. Not surprisingly, a direct application of Newton's method to the DTH equations (5)–(6) performs poorly for small time steps. It is possible to show that the nested iteration scheme, outlined at the end of the previous section, avoids ill-conditioned matrices [4]. Nested iterations, however, are computationally expensive.

We now derive more efficient iteration schemes. Linearizing equations (5)–(6) we have

$$\mathsf{f} + \mathsf{f}_\mathsf{z} d\mathsf{z} + \mathsf{f}_\lambda d\lambda \quad = \quad 0 \tag{9}$$

$$g + g_\mathsf{z}^\top d\mathsf{z} + g_\lambda d\lambda \quad = \quad 0. \tag{10}$$

Solving equation (9) for $d\mathsf{z}$ yields the equation

$$d\mathsf{z} = \mathsf{f}_\mathsf{z}^{-1}\left(-\mathsf{f} - \mathsf{f}_\lambda d\lambda\right). \tag{11}$$

(Note that, for sufficiently small λ, $f_z = I - \frac{1}{2}\lambda J H_{zz}$ is a well-posed matrix.) Substituting equation (11) into equation (10) yields

$$g + g_\mathsf{z}^\top \mathsf{f}_\mathsf{z}^{-1}\left(-\mathsf{f} - \mathsf{f}_\lambda d\lambda\right) + g_\lambda d\lambda = 0 \tag{12}$$

Solving (12) for $d\lambda$ gives the following iteration:

$$d\lambda = \frac{g - g_\mathsf{z}^\top \mathsf{f}_\mathsf{z}^{-1}\mathsf{f}}{g_\mathsf{z}^\top \mathsf{f}_\mathsf{z}^{-1}\mathsf{f}_\lambda - g_\lambda} \tag{13}$$

$$d\mathsf{z} = -\mathsf{f}_\mathsf{z}^{-1}\left(\mathsf{f} + \mathsf{f}_\lambda d\lambda\right). \tag{14}$$

I have found that it is more efficient to approximate $\mathsf{f}_\mathsf{z}^{-1}$ by using the matrix perturbation lemma

$$\mathsf{f}_\mathsf{z}^{-1} = \left(I - \frac{1}{2}\lambda J\mathcal{H}_{\mathsf{zz}}\right)^{-1} \approx I + \frac{1}{2}\lambda J\mathcal{H}_{\mathsf{zz}}.$$

I have also found that it is more efficient to evaluate the Hessian matrix, $\mathcal{H}_{\mathsf{zz}}$, only once, at the beginning of each time step and use it repeatedly for all subsequent iterations. In fact, it is possible to avoid computing the Hessian matrix $\mathcal{H}_{\mathsf{zz}}$ altogether by using the following Broyden estimate [11] for $\mathcal{H}_{\mathsf{zz}}$ at the beginning of each time step:

$$\mathcal{H}_{\mathsf{zz}} \approx \frac{\left(\mathcal{H}_\mathsf{z}^+ - \mathcal{H}_\mathsf{z}\right)\delta\mathsf{z}^\top}{\delta\mathsf{z}^\top\delta\mathsf{z}}$$

where $\mathcal{H}_\mathsf{z}^+$ is the value of the gradient of $\mathcal{H}$ at $\mathsf{z}_+ = \mathsf{z}_k + \frac{1}{2}\lambda_{k-1}J\mathcal{H}_\mathsf{z}(\mathsf{z}_k)$ and $\delta\mathsf{z} = \mathsf{z}_+ - \mathsf{z}_k$.

4. Numerical Results

In this section, I present numerical simulation results for the five, second-order methods listed in Table 1. First, I present numerical results for a two-dimensional, three-body, periodic orbit described in [12]. (See Figure 2.) Simo describes this figure-eight orbit as having remarkable stability properties. We should expect this orbit to behave robustly to discretization errors. Since this orbit does not cross $\psi(\mathsf{z}) = 0$ manifolds, DTH trajectories for this orbit do not need to be regularized.

Next, I present numerical results for an entirely different type of three-body trajectory—one involving the near-collision and subsequent scattering of the three bodies. This near-collision trajectory is plotted in Figure 3. The near-collision trajectory crosses $\psi(\mathsf{z}) = 0$ manifolds several times, so DTH trajectories need to be regularized for this trajectory. How this is done is beyond the scope of this article, but I describe how I do this in detail in [10].

Table 2 lists the normalized, average CPU time per time step for five different integration methods and several different iteration schemes, for the figure-eight orbit. The leapfrog and Runge-Kutta methods, which are the only explicit methods, are significantly faster than the other methods. All methods become faster, on a per time step basis, as the time step becomes smaller. For the explicit methods, this is probably do to more efficient utilization of the CPU. For the implicit

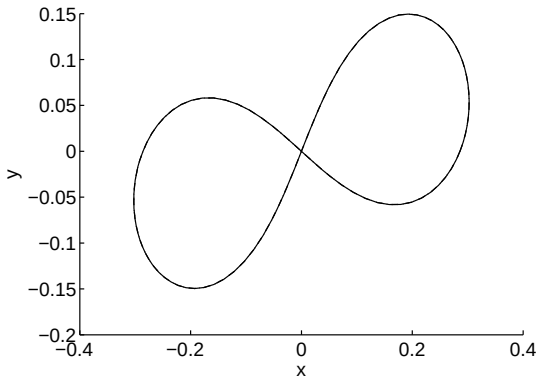

Fig. 2
THREE-BODY FIGURE-EIGHT ORBIT.

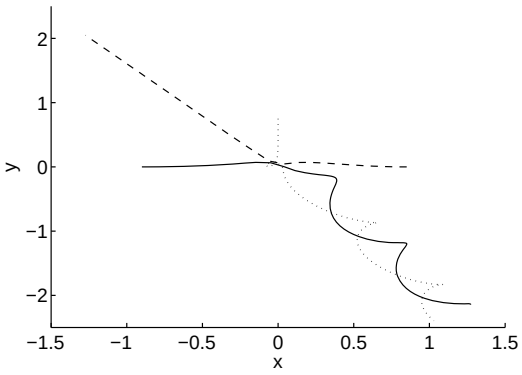

Fig. 3
THREE BODY NEAR-COLLISION TRAJECTORY.

methods, the need for fewer iterations to achieve convergence as the time step is reduced, likely plays a larger role.

The Newton and Broyden iteration schemes for the DTH equations roughly take the same CPU time per time step. The Broyden iteration has the advantage of not requiring computation of the Hessian matrix, $\mathcal{H}_{\mathsf{zz}}$, so we use the Broyden iteration for all subsequent comparisons.

Simulation results showed that the midpoint method did not benefit from a Broyden estimate of the Hessian matrix, $\mathcal{H}_{\mathsf{zz}}$. Functional iterations, in which the Hessian matrix, $\mathcal{H}_{\mathsf{zz}}$, is simply replaced by the identity matrix, were slightly faster for the midpoint method.

The leapfrog method exploits the special structure of the Hamiltonian function of the three-body problem expressed in Cartesian coordinates. (The leapfrog method is, in general, implicit for arbitrary Hamiltonian functions.) The iteration scheme for the midpoint method can be modified to also exploit this special structure of the three-body, Hamiltonian function when expressed in Cartesian coordinates. I do this by explicitly solving for the momentum coordinates in terms of the position coordinates. The modified iteration results in

	Time Steps per Orbit					
	100	200	400	800	1600	3200
leapfrog	3.3	1.2	1.2	1.1	1.0	1.0
Runge Kutta (2nd order)	3.4	1.9	1.8	1.8	1.7	1.8
midpoint (Newton)	13.9	11.6	10.5	10.1	8.9	8.9
midpoint (functional)	14.4	10.1	8.6	7.5	6.5	6.4
midpoint (Newton-modified)	11.0	8.2	7.6	7.3	7.9	7.6
midpoint (functional-modified)	9.3	8.1	6.1	4.7	4.6	4.7
discrete mechanics (functional)	29.8	20.5	17.7	15.6	13.2	13.4
DTH dynamics (nested-Newton)	201.8	182.4	159.9	147.0	126.6	112.5
DTH dynamics (Newton)	34.9	23.9	22.1	19.5	19.2	19.3
DTH dynamics (Broyden)	40.1	27.0	23.0	20.7	19.0	18.9

Table 2

AVERAGE CPU TIME PER TIME STEP, OVER ONE ORBIT. TIMES ARE NORMALIZED BY THE SMALLEST AVERAGE CPU TIME.

approximately a 25% reduction in CPU time. I was unable to derive a similarly modified iteration for the DTH equations. (The Lagrange multiplier in the DTH equations introduces a coupling of position and momentum coordinates.) The function evaluations of the discrete mechanics method of Greenspan are significantly different from the other methods. Do to time constraints, a derivation of a modified iteration exploiting the special structure of the three-body problem in Cartesian coordinates was not attempted for the discrete mechanics of Greenspan.

The efficiency of a method is not only determined by CPU time per time step, but also the accuracy of the time step. To compare the computational efficiency of each method, I plot CPU time vs coordinate accuracy in Figure 4. (Coordinate accuracy is computed by comparing to the trajectory computed with Matlab's Dormand-Prince 4–5th order Runge-Kutta method.) From Figure 4, we can see that the leapfrog method is the most efficient method by a significant margin. Despite their special properties (see Table 1), the discrete mechanics of Greenspan and DTH dynamics are the least efficient methods for this trajectory.

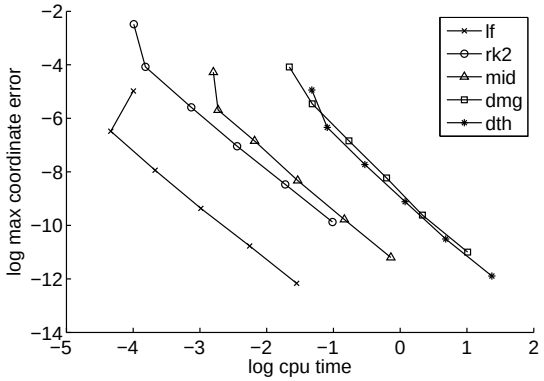

Fig. 4

ACCURACY VS CPU TIME FOR ONE ORBIT OF THE FIGURE-EIGHT TRAJECTORY.

Good long-time behavior is a hallmark of symplectic integration methods. To assess long-time behavior, I plot in Figure 5, the error in energy conservation (at vertex points) as a function of time. Even though the discrete mechanics of Greenspan is not symplectic, it conserves energy up to round-off error/convergence tolerance and has the best long-time energy behavior at vertex points. The second-order Runge-Kutta method is the other method which is not symplectic. It has the worst long-time behavior.

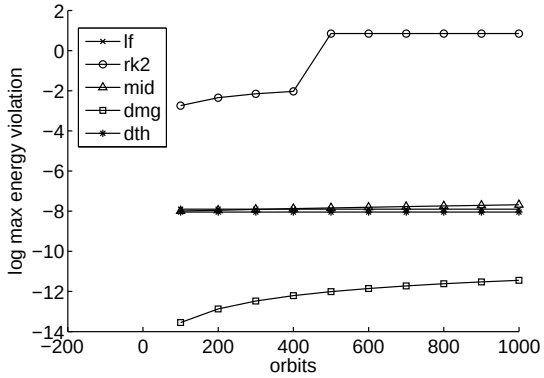

Fig. 5

ENERGY CONSERVATION ERROR FOR THE FIGURE-EIGHT ORBIT.

Finally, I present simulation results for the near-collision trajectory plotted in Figure 3. Figure 6 shows that, for small time steps, the variable time step capability of DTH dynamics makes it as efficient as the leapfrog method.

5. Conclusions

Despite having many desirable properties, simulation results in this article show that SEM integration, currently, can not reliably beat the widely used leapfrog method. Even more efficient iteration schemes will be required before this is possible. SEM integration may prove competitive for other applications for which the leapfrog method is implicit. One

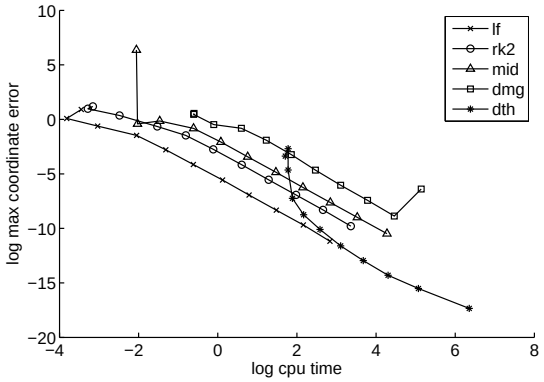

Fig. 6
ACCURACY VS CPU TIME FOR THE NEAR-COLLISION TRAJECTORY.

such application is molecular dynamics simulations of protein molecules in ϕ–ψ internal coordinates.

I conclude by pointing out that other iterations schemes for DTH dynamics have been derived by Ander Murua [13]. These iterations schemes will be included in future studies.

References

[1] Y. Wu, "The discrete variational approach to the Euler-Lagrange equations," *Computers and Mathematics with Applications*, vol. 20, pp. 61–75, 1990.

[2] J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems*. Chapman and Hall, 1994.

[3] C. Kane, J. Marsden, and M. Ortiz, "Symplectic-energy-momentum preserving variational integrators," *Journal of Mathematical Physics*, vol. 40, July 1999.

[4] Y. Shibberu, "Discrete-time Hamiltonian dynamics," Ph.D. dissertation, Univ. of Texas at Arlington, 1992, www.rose-hulman.edu/∼shibberu/DTH_Dynamics/DTH_Dynamics.htm.

[5] D. Greenspan, *Discrete Numerical Methods in Physics and Engineering*. Academic Press, 1974.

[6] ——, *Arithmetic Applied Mathematics*. Pergamon Press, 1980.

[7] T. D. Lee, "Can time be a discrete dynamic variable?" *Physics Letters Physics*, vol. 122B, pp. 217–220, 1983.

[8] ——, "Difference equations and conservation laws," *Journal of Statistical Physics*, vol. 46, pp. 843–860, 1987.

[9] Y. Shibberu, "Is symplectic-energy-momentum integration well-posed." August 2006, arXiv:math-ph/0608016v1.

[10] ——, "How to regularize a symplectic-energy-momentum integrator," July 2005, arXiv:math/0507483v1.

[11] J. J.E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996.

[12] C. Simó, "Dynamical properties of the figure eight solution of the three-body problem," in *Celestial Mechanics, Dedicated to Donald Saari for his 60th Birthday*, A. Chenciner, R. Cushman, C. Robinson, and Z. J. Xia, Eds., 2002, pp. 209–+.

[13] A. Murua, 1997, private communication.