1 Introduction

Geometric Numerical Integration, when considered on the grand timeline of Mathematics, is a field still in its younger days. While work on the numerical treatment of differential equations had started off towards the end of the 19th Century, it wasn't till the 1980s that numerical analysts shifted their focus onto Geometric Numerical Integration. A better part of the century had been involved in improving quantity - producting methods that could produce accurate results without being computationally strenuous. But the interest in running computer simulations over longer time intervals revealed that quality was equally important. Most methods could accurately reflect properties of a system, such as its asymptotic time behaviour or its sensitivity to changes in initial conditions; not all could preserve attributes such as total enery or momentum - properties that are invariants to the flow of the differential equation.

Consider, for example [1], the following set of Lotka-Volterra prey-predator equations:

$$\frac{du}{dt} = u - uv, \qquad \frac{dv}{dt} = uv - 2v \tag{1.1}$$

where $u(t)$ is the population of the prey over time, and $v(t)$ is that of the predator. Dividing the two equations in 1.1, we get that

$$\left(1 - \frac{2}{u}\right) du = \left(\frac{1}{v} - 1\right) dv$$

Integrating the two sides of the equation, we get that

$$I(u,v) = u - 2\ln u + v - \ln v = const \tag{1.2}$$

giving us that the solutions to 1.1 lie on the level curves of 1.2.

We solve the differential equation 1.1 with initial points $(u_0, v_0) = (2, 4)$ using three different numerical methods: Explicit Euler, Implicit Euler, and Symplectic Euler. The results can be found in Figure 1.1.

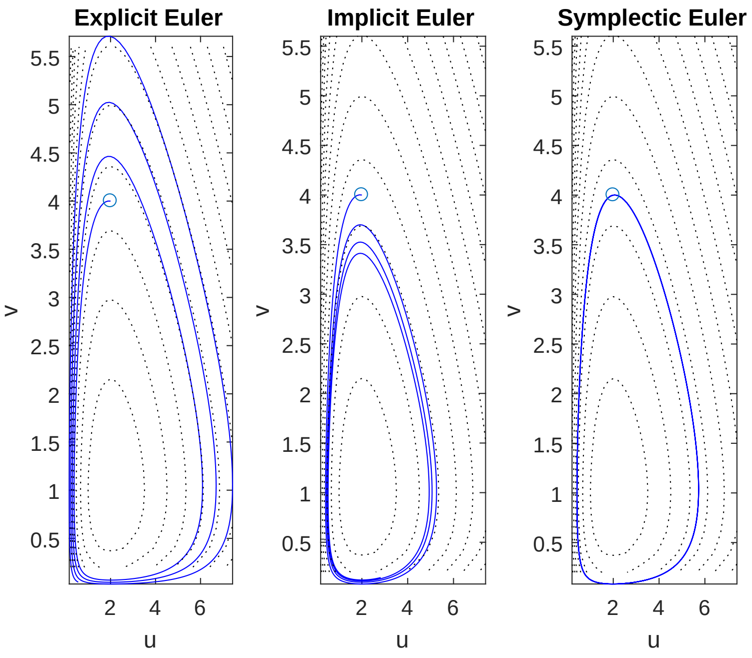


Figure 1.1: Solutions to the Lotka-Volterra Equation

The phase planes in Figure 1.1 convey that the solutions must be a closed loops and periodic - but only the solution using the Symplectic Euler scheme satisifies this criterion. The other two methods qualitatively incorrectly - the solution either spirals inwards or outwards.

Hence, it is important to study methods that preserve qualitative attributes of a system as weell. Since these attributes often occur in differential geometry (the study of differential equations in geometry), we call them geometric invariants - thus lending the name 'Geometric Numerical Integration' to their study.

In this paper, we will focus on a specific class of geometric invariants - non linear, highly oscillatory Hamiltonian problems. We will devote our attention to the problem of molecular dynamics - specifically a simulation of argon molecules as did by A. Rahman [4], and L. Verlet [5]. We will first investigate the preservation of geometric invariants in the aforementioned system by the industry standard, Velocity Verlet algorithm, followed by an investigation using the Newmark-Beta family of numerical methods (of which the Velocity Verlet algorithm is a special case).We will conclude with a section on improving the speed of the molecular dynamics simulation, and the effects of improving speed on the action of the above methods.

2 Newmark Beta Methods

In this section, we will look at a family of numerical methods used to solve second order ordinary differential equations. We will look at the general formula, the algorithm to implement them, and an experimental verification of convergence order for different values of the parameters.

2.1 Introduction to Newmark Beta Methods

The Newmark Beta methods were presented by Nathan M. Newmark in a paper [3] published in 1959. They were intended to be used for find solutions to problems in structural dynamics, *"capable of application to strucutres of any degree of complication, with any relationship between force and displacement"*, with considerations for *"any time of dynamic loading such as that due to shock or impact, vibration, earthquake motion, or blast from a nuclear weapon"*. However, they can also be used in less apocalyptical scenarios - with a special case known as the 'Velocity Verlet' algorithm being the standard for solving the equatoins of motion in molecular dynamics simulations. Writing $\mathbf{x}_i$, $\mathbf{v}_i$, and $\mathbf{a}_i$ for the displacement, velocity, and acceleration at timestep i respectively, the methods are of the form:

$$\begin{aligned} \mathbf{v}_{i+1} &= \mathbf{v}_i + h\left[(1-\gamma)\,\mathbf{a}_i + \gamma\mathbf{a}_{i+1}\right] \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + h\mathbf{v}_i + \frac{h^2}{2}\left[(1-2\beta)\,\mathbf{a}_i + 2\beta\mathbf{a}_{i+1}\right] \end{aligned} \tag{2.1}$$

where $\gamma \in [0,1]$ and $2\beta \in [0,1]$.

We have that the methods are second order accurate if and only if $\gamma = \frac{1}{2}$, and that they are conditionally stable if and only if $2\beta \geq \gamma \geq \frac{1}{2}$. The proof of these attributes is beyond the scope of this paper, and can be found in [2]. However, we will look at confirming the first statement through numerical simulations in Subsection **??**.

Some of the common choices of parameters β and γ are:

- Average Acceleration Method $\left(\gamma = \frac{1}{2},\, \beta = \frac{1}{4}\right)$:

$$\begin{aligned} \mathbf{v}_{i+1} &= \mathbf{v}_i + h\left[\frac{1}{2}\mathbf{a}_i + \frac{1}{2}\mathbf{a}_{i+1}\right] \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + h\mathbf{v}_i + \frac{h^2}{2}\left[\frac{1}{2}\mathbf{a}_i + \frac{1}{2}\mathbf{a}_{i+1}\right] \end{aligned}$$

- Linear Acceleration Method $\left(\gamma = \frac{1}{2},\, \beta = \frac{1}{6}\right)$:

$$\begin{aligned} \mathbf{v}_{i+1} &= \mathbf{v}_i + h\left[\frac{1}{2}\mathbf{a}_i + \frac{1}{2}\mathbf{a}_{i+1}\right] \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + h\mathbf{v}_i + \frac{h^2}{2}\left[\frac{2}{3}\mathbf{a}_i + \frac{1}{3}\mathbf{a}_{i+1}\right] \end{aligned}$$

- Velocity Verlet Method $\left(\gamma = \frac{1}{2},\, \beta = 0\right)$:

$$\begin{aligned} \mathbf{v}_{i+1} &= \mathbf{v}_i + h\left[\frac{1}{2}\mathbf{a}_i + \frac{1}{2}\mathbf{a}_{i+1}\right] \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + h\mathbf{v}_i + \frac{h^2}{2}\mathbf{a}_i \end{aligned} \tag{2.2}$$

2.2 Computational Application of Newmark Beta Methods

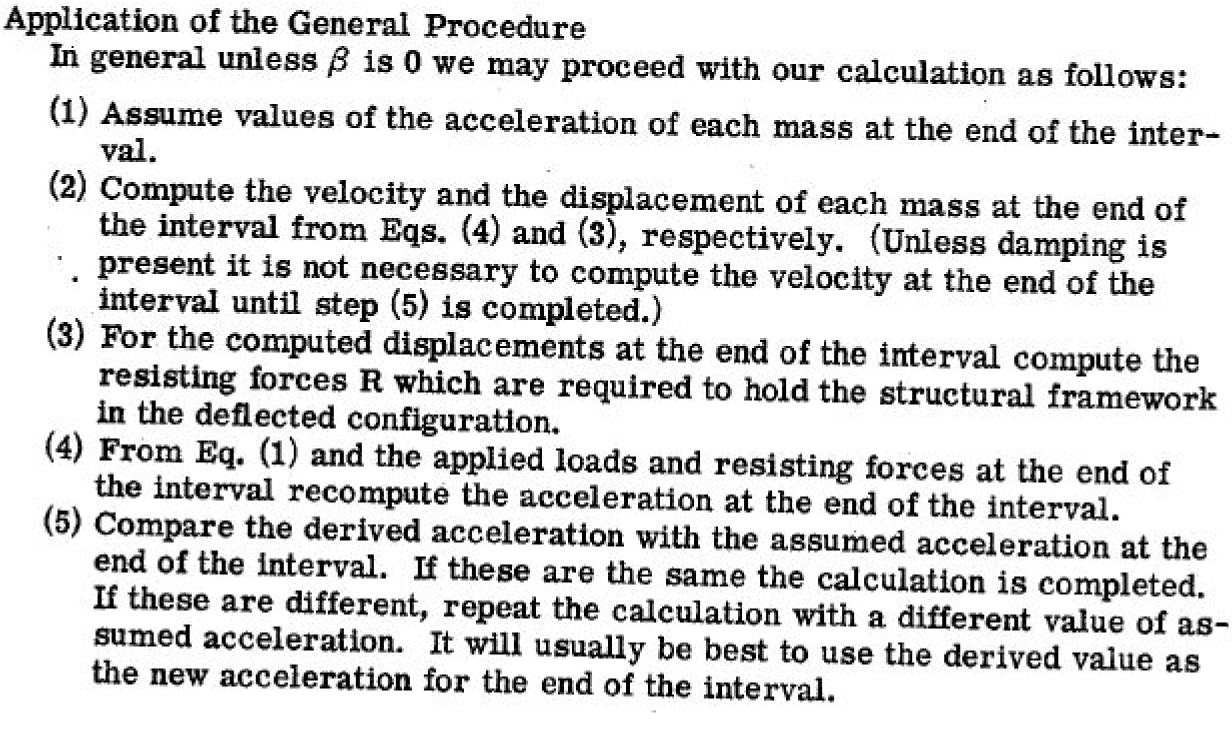
Figure 2.1 shows an excerpt from the original manuscript [3] where Newmark outlines an algorithm to calculate the next values. The method suggested belongs to a class of methods known as 'predictor-corrector' methods - an initial 'predicition' of the quantity at t_{i+1} is calculated using some function, and then this value is 'corrected' by using the prediction as the initial value for some other function to calculate the value of the quantity at the same point in time, t_{i+1}. The example below depicts the Heun method for an ODE of the form $y' = f(t, y)\,, \quad y(t_0) = y_0$.

$$\begin{aligned} \tilde{y}_{i+1} &= y_i + hf(t_i,\, y_i) && \text{prediciting using Forward Euler Method} \\ y_{i+1} &= y_i + \frac{h}{2}\left(f(t_i, y_i) + f(t_{i+1}, \tilde{y}_{i+1})\right). && \text{correcting using Trapezium Rule} \end{aligned}$$

However, this raises a question: how many times should the cycle of prediction and correction be run to get an accurate result? We can avoid this question by simplifying our assumptions. Since the goal of this paper is to apply the Newmark Beta methods to a molecular dynamics simulation, we can tune the method accordingly. In the absence of external forces, the molecular dynamics simulation is a conservative system. So the force (and by Newton's Second Law, the acceleration) is dependent only on the displacement of the particles. Thus, we can re-write Equation2.1 as:

$$\begin{aligned} \mathbf{v}_{i+1} &= \mathbf{v}_i + h\left[(1-\gamma)\,\mathbf{a}_i + \gamma\mathbf{a}_{i+1}(\mathbf{x}_{i+1})\right] \\ \mathbf{x}_{i+1} &= \mathbf{x}_i + h\mathbf{v}_i + \frac{h^2}{2}\left[(1-2\beta)\,\mathbf{a}_i + 2\beta\mathbf{a}_{i+1}(\mathbf{x}_{i+1})\right] \end{aligned}$$

as we will already have computed the values of $\mathbf{x}_i, \mathbf{v}_i,$ and $\mathbf{a}_i$.



Application of the General Procedure

In general unless β is 0 we may proceed with our calculation as follows:

(1) Assume values of the acceleration of each mass at the end of the interval.

(2) Compute the velocity and the displacement of each mass at the end of the interval from Eqs. (4) and (3), respectively. (Unless damping is present it is not necessary to compute the velocity at the end of the interval until step (5) is completed.)

(3) For the computed displacements at the end of the interval compute the resisting forces R which are required to hold the structural framework in the deflected configuration.

(4) From Eq. (1) and the applied loads and resisting forces at the end of the interval recompute the acceleration at the end of the interval.

(5) Compare the derived acceleration with the assumed acceleration at the end of the interval. If these are the same the calculation is completed. If these are different, repeat the calculation with a different value of assumed acceleration. It will usually be best to use the derived value as the new acceleration for the end of the interval.

Figure 2.1: Excerpt on applying the Newmark Beta methods computationally

Thus, we can see that the equation is implicit for $\mathbf{x}_{i+1}$. However, it is explicit for $\mathbf{v}_{i+1}$, as we can calculate $\mathbf{a}_{i+1}$ once we have the value of $\mathbf{x}_{i+1}$. This algorithm is depicted in Figure [CREATE FIGURE AND ADD REFERENCE]. We use Newton-Raphson Iteration to solve for $\mathbf{x}_{i+1}$:

$$\begin{aligned}\tilde{\mathbf{F}}\left(\mathbf{x}_{i+1}\right) &= \mathbf{x}_{i+1} - \mathbf{x}_i - h\left(\mathbf{v}_i + \frac{h}{2}\left(\left(1-2\beta\right)\mathbf{a}_i + 2\beta\mathbf{a}_{i+1}\left(\mathbf{x}_{i+1}\right)\right)\right) \\ D_{\mathbf{x}_{i+1}}\tilde{\mathbf{F}}\left(\mathbf{x}_{i+1}\right) &= \mathbf{I} - \beta h^2 \mathbf{J}_{\mathbf{a}_{i+1}} \\ \mathbf{x}_{i+1} &= \mathbf{x}_i - D\tilde{\mathbf{F}}\left(\mathbf{x}_i\right)^{-1}\tilde{\mathbf{F}}\left(\mathbf{x}_i\right) \qquad (2.3)\end{aligned}$$

provided that the inverse exists. Thus, we use the following algorithm to find the next set of points:

Algorithm 2.1: Newmark-Beta Method Algorithm

```
input  : $\mathbf{x}_i, \mathbf{v}_i, \mathbf{a}_i, h, \gamma, \beta$
output: $\mathbf{x}_{i+1}, \mathbf{v}_{i+1}, \mathbf{a}_{i+1}$

$\mathbf{v}_{i+1} \leftarrow \mathbf{v}_i + h\left(1-\gamma\right)\mathbf{a}_i$
Calculate $\mathbf{x}_{i+1}$ using the Newton-Raphson Method
$\mathbf{a}_{i+1} \leftarrow \frac{1}{m}\mathbf{F}\left(\mathbf{x}_{i+1}\right)$
$\mathbf{v}_{i+1} \leftarrow \mathbf{v}_{i+1} + \gamma h \mathbf{a}_{i+1}$ ;       // Splitting the update into two steps to reduce space
  usage
```

References

[1] Ernst Hairer, Gerhard Wanner, and Christian Lubich. *Examples and Numerical Experiments*, pages 1–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[2] Nathan M. Newmark. Computation of dynamic structural response in the range approaching failure. In *Proceedings of the Symposium on Earthquake and Blast Effects on Structures*, Los Angeles, 1952. Earthquake Engineering Research Institute.

[3] Nathan M. Newmark. A method of computation for structural dynamics. *American Society of Civil Engineers*, EM 3, 1959.

[4] Aneesur Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A):405–411, 1964.

[5] Loup Verlet. Computer experiments on classical fluids. i. thermodynamical properties of lennard -jones molecules. *Physical Review*, 159(1):98–103, 1967.