SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Mechanics

Journal homepage:
https://semarakilmu.com.my/journals/index.php/appl_mech/index
ISSN: 2289-7895

A Comparative Study of Numerical Modelling and Analysis for Large Articulated Pendulums

Siti Fatimah Azzahra Ahmad Noh[1,*], Mohamad Ezral Baharudin[1], Mohd Zakimi Zakaria[1], Mohd Sazli Saad[1], Azuwir Mohd Noor[1]

[1] Faculty of Mechanical Engineering Technology, Pauh Putra Campus, Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia

ARTICLE INFO

Article history:
Received 20 October 2024
Received in revised form 21 November 2024
Accepted 28 November 2024
Available online 30 December 2024

Keywords:
Augmented Lagrangian; fully recursive; multibody dynamics

ABSTRACT

In this article, we present a large system of multiple pendulums, also articulated pendulums, with twenty pendulums as a multibody model. The main objective of the study is to compare the computational time efficiency of two multibody formulations: the augmented Lagrangian and the recursive method for each articulated system. The equations of motion were derived for each formulation and the fourth- and fifth-order Runge-Kutta methods were utilised to solve for the equations by representing the kinematics and dynamics of the systems numerically. The computational times that corresponded to the manipulated step size and tolerance were compared for both formulations. The results showed that the augmented Lagrangian formulation had a significant divergence towards the negative y-axis at tolerance 0.1s for all modified step sizes. The animations also demonstrated elongation for specific pendulums based on the step size selection at a tolerance 0.1s. The recursive method, on the other hand, produced the best-fit plots and stable results for all xy-position and velocity-time plots for each adjusted step size and tolerance. Therefore, the recursive method is concluded to be more efficient than the augmented Lagrangian formulation in solving large open-loop multibody systems.

1. Introduction

A simple pendulum is usually made up of a small weight or bob suspended from a fixed point by a thin string [1]. Displacing the pendulum's position from its equilibrium will cause it to sway back and forth about the point. Recent studies by Bondada *et al.,* [2] mentioned that the literature on mechanical pendulums has been studied for years, from a simple linear pendulum to complex nonlinear pendulum systems such as multiple pendulums [3] and inverted pendulums [4]. The dynamics of the pendulum systems have been applied in various applications such as cranes [5,6], humanoid robots [4,7], energy harvester [8-10] and many more [11].

For this work, we extended the pendulum system into twenty articulated pendulums to represent a multibody model [12]. In general, a multibody system is a study of interconnected rigid or flexible

* *Corresponding author.*
E-mail address: sfazzahra@studentmail.unimap.edu.my

https://doi.org/10.37934/aram.129.1.105121

bodies through joints [13-15]. According to Mahbouba *et al.,* [16]. The pendulum is a very useful tool for expressing body motion and understanding its equation and finding the solution was therefore required to predict their movement over time and depend on the various initial conditions. As shown in Figure 1, a number of articulated pendulums, n are linked together through joints, with a length of the pendulums, *l*. The joints in articulated bodies have numerous degrees of freedom (DOF) and constraints length or angle [17]. A minimum of two active DOFs are usually included in articulated bodies to attain spatial capabilities [18].

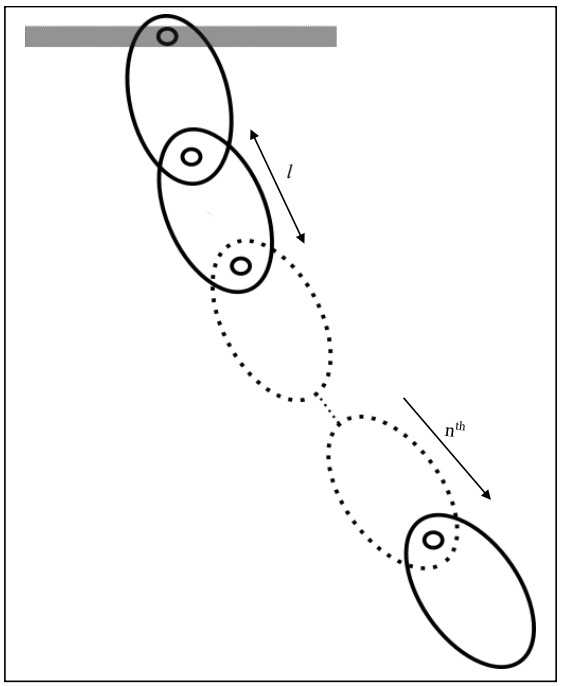

Fig. 1. Illustration of nth articulated pendulums

By assuming an equal distribution of mass, *m* along the articulated pendulum's length, *l*, each body's centre of mass is positioned at its midpoint, with a moment of inertia, I= $ml^2/12$. The angle, ϑ_n between each arm and the vertical axis is taken as generalized coordinates to describe the system's configuration as shown in Figure 2.

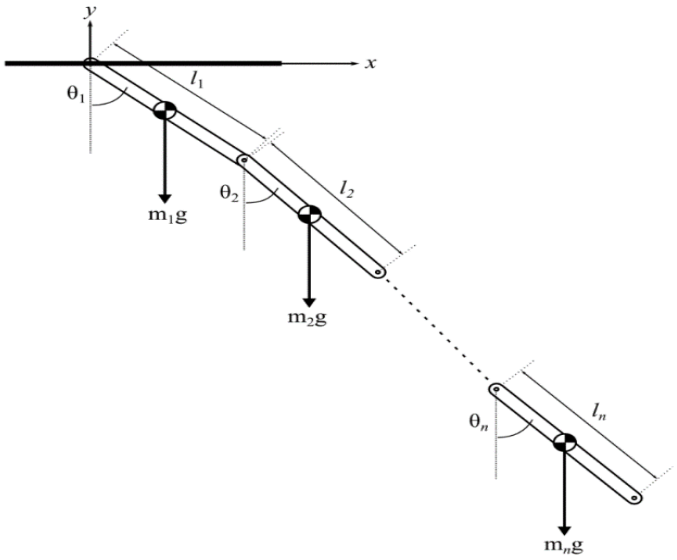

Fig. 2. System configuration of nth articulated pendulums

The suspension point of the first pendulum serves as the origin of the coordinate system. By assuming the position of the first pendulum's centre of mass such that:

$$x_1 = \frac{l_1}{2} cos\theta_1 \tag{1}$$

$$y_1 = \frac{l_1}{2} sin\theta_1 \tag{2}$$

The positions in the horizontal and vertical directions are represented by the variables *x* and *y*, respectively. Subsequently, the position of the second pendulum's centre of mass can further be defined as:

$$x_2 = \frac{l_2}{2} cos\theta_2 + l_1 cos\theta_1 \tag{3}$$

$$y_2 = \frac{l_2}{2} sin\theta_2 + l_1 sin\theta_1 \tag{4}$$

The remaining equations for n=3 until n=20 in the articulated system can further be determined by the same mathematical patterns. Since the system is constrained, the equations of motion were derived based on two well-known multibody formulations, that are the augmented Lagrangian and fully recursive method to compare their computational efficiency in the large open-loop system. The efficiency comparison between these two formulations may provide valuable insights into their computational capabilities and relative effectiveness for further research in multibody dynamics system [15,19] .

2. Multibody Formulations

In the study of multibody system, the kinematics of the system need to be determined before the equations of motion can be developed. There are two methods to describe the kinematics of the multibody system; the global formulation and the topological method. The coordinates of the bodies are described with respect to the global frame of reference in the global formulation and the position of the bodies will be based on the preceding body frame of reference in the topological method [20].

2.1 Kinematics

In multibody systems, the bodies connected by joints are governed by constraints on the particles. These constraints are often easiest to describe when they are set in the coordinate system of the body frame of reference. In Figure 3, the global frame of reference coordinates is positioned at the same point as body A, with both sets of axes parallel to each other. As the particle P is attached to the body A, the body's frame of reference coordinates, ($\overline{\mathbf{X}}_\mathbf{1}$, $\overline{\mathbf{X}}_\mathbf{2}$) are also fixed to the body and move with it. The components $\overline{\boldsymbol{u}}_\mathbf{1}$ and $\overline{\boldsymbol{u}}_\mathbf{2}$, which are parallel to the body frame of the reference coordinate axis, make up the vector $\overline{\mathbf{u}}$ which reflects the position of particle P in the body reference coordinates. The vector $\mathbf{r}_P$, on the other hand, specifies where the particle P is located in the global frame of reference coordinates.

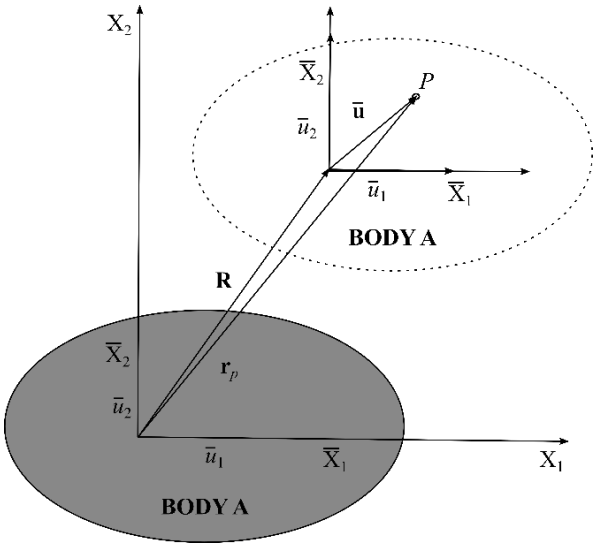

Fig. 3. The body and global frame of reference coordinates of A

By referring to the global frame of reference coordinates, the following expression can be used to describe the particle's position:

$$\mathbf{r}_P = \bar{\mathbf{u}} = \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} \tag{5}$$

Following this, a vector **R** was used to move the body frame of reference coordinates, which changed the particle P's position vector $\mathbf{r}_P$ to what is shown in the diagram:

$$\mathbf{r}_P = \mathbf{R} + \bar{\mathbf{u}} = \begin{bmatrix} R_1 \\ R_2 \end{bmatrix} + \begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix} \tag{6}$$

The equation $\boldsymbol{r}_P = \mathbf{R} + \bar{\mathbf{u}}$ is no longer adequate to represent the location of particle P if the body frame of reference coordinates is both moved by the vector **R** and rotated by an angle θ. As a result, the vector $\bar{\mathbf{u}}$ must be defined in terms of the global frame of reference coordinates to determine the position of particle P. As shown in Figure 4, the component $\bar{u}_1$ has a_1 projection on the global frame of reference axis X_1, while the component $\bar{u}_2$ has a_2 projection on the global frame of reference axis X_1.

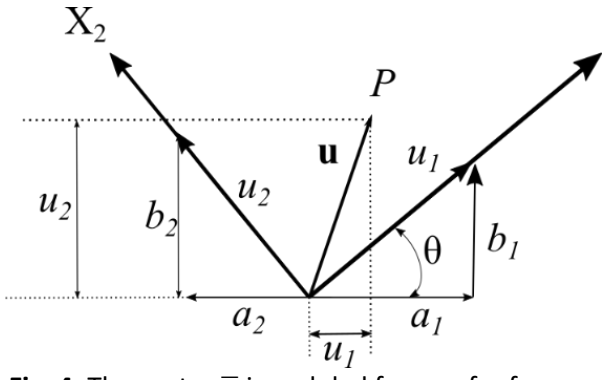

Fig. 4. The vector $\bar{\boldsymbol{u}}$ in a global frame of reference

The vector **u** is expressed such that:

$$\underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\mathbf{u}} = \begin{bmatrix} a_1 - a_2 \\ b_1 + b_2 \end{bmatrix} = \begin{bmatrix} \bar{u}_1 \cos(\theta) - \bar{u}_2 \sin(\theta) \\ \bar{u}_1 \sin(\theta) + \bar{u}_2 \cos(\theta) \end{bmatrix} = \underbrace{\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \bar{u}_1 \\ \bar{u}_2 \end{bmatrix}}_{\bar{\mathbf{u}}} \tag{7}$$

The rotation matrix is referred to as matrix **A**. The position of particle P can be defined by the following expression by including the translation of the body frame of reference coordinates **R**:

$$\underbrace{\begin{bmatrix} r_1 \\ r_2 \end{bmatrix}}_{\mathbf{r}} = \underbrace{\begin{bmatrix} R_1 \\ R_2 \end{bmatrix}}_{\mathbf{R}} + \underbrace{\begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \overline{u_1} \\ \overline{u_2} \end{bmatrix}}_{\bar{\mathbf{u}}} = \underbrace{\begin{bmatrix} R_1 \\ R_2 \end{bmatrix}}_{\mathbf{R}} + \underbrace{\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}}_{\mathbf{u}} \tag{8}$$

2.1.1. Equations of motion

The principle of virtual work is frequently used in the development of equations of motion. The dynamic equilibrium can be reached in an unconstrained system by:

$$\delta W_{iner} = \delta W_{ext} \tag{9}$$

whereby δW_{iner} is the virtual work of the inertial forces and δW_{ext} is the external applied forces. This comprises the accumulation of externally imposed forces as well as the inertia's virtual work.

$$\delta W_{iner} = \delta \boldsymbol{q} \cdot (\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_{\boldsymbol{v}}) \tag{10}$$

$$\delta W_{ext} = \delta \boldsymbol{q} \cdot \mathbf{Q}_{\boldsymbol{e}} \tag{11}$$

From above, the mass matrix is denoted as **M**, the generalized coordinates as $\ddot{\mathbf{q}}$, the vector of quadratic velocity vector as $\mathbf{Q}_{\boldsymbol{v}}$ and the vector of the generalized force of the multibody system as $\mathbf{Q}_{\boldsymbol{e}}$. Equalizing the Eq. (10) and Eq. (11), the equations became:

$$\delta \boldsymbol{q} \cdot (\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_{\boldsymbol{v}} - \mathbf{Q}_{\boldsymbol{e}}) = \mathbf{0} \tag{12}$$

Since the system's constraints are ignored, $(\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_{\boldsymbol{v}} - \mathbf{Q}_{\boldsymbol{e}})$ cannot be equal to zero. Thus, the equations of motion must contain the constraints to reach for the dynamic equilibrium in the use of multibody.

2.1.2 Lagrange multiplier method

The constraint vector, $\boldsymbol{\Phi}$ which contains the constraint equations and can be integrated into the equations of motion in a multibody system. The number of constraint equations, n_c in the vector of constraints is less or equal to the number of generalized coordinates, n_g. Hence, the following equation must be satisfied to satisfy the relationships between the constraint equations and generalized coordinates [21]:

$$\boldsymbol{\Phi}\,(\mathbf{q}, t) = \mathbf{0} \tag{13}$$

where t is time. Nonetheless, the Lagrange multiplier approach entails a set of multipliers, $\boldsymbol{\lambda}$ in the constraints of the equations of motion [22]. The reaction forces due to the constraints are connected to the vector of Lagrange multipliers, such that:

$$\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q}_e - \mathbf{Q}_v + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{0} \tag{14}$$

The Jacobian matrix of the constraints is symbolized by $\boldsymbol{\Phi}_{\mathbf{q}}$. Hence, the equations of motion can be transformed into a matrix from Eq. (13) and Eq. (14) as follows:

$$\begin{bmatrix} \mathbf{M} & \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \boldsymbol{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_e + \mathbf{Q}_v \\ \mathbf{Q}_c \end{bmatrix} \tag{15}$$

The constraints are expressed in terms of acceleration constraints in Eq. (15), where the acceleration vectors $\ddot{\mathbf{q}}$ and Lagrange multipliers $\boldsymbol{\lambda}$ are still unknown. To get more understanding of Lagrange multipliers, Figure 5 illustrates the values of four Lagrange multipliers representing constraints applied at two joints of a double pendulum in both the x and y axes. The x-axis shows the time steps of the simulation, while the y-axis indicates the values of the Lagrange multipliers. Different coloured lines correspond to the multipliers for each constraint. Notable peaks and troughs indicate significant constraint forces at specific times, suggesting critical moments in the pendulum's motion, such as changes in direction or collisions.

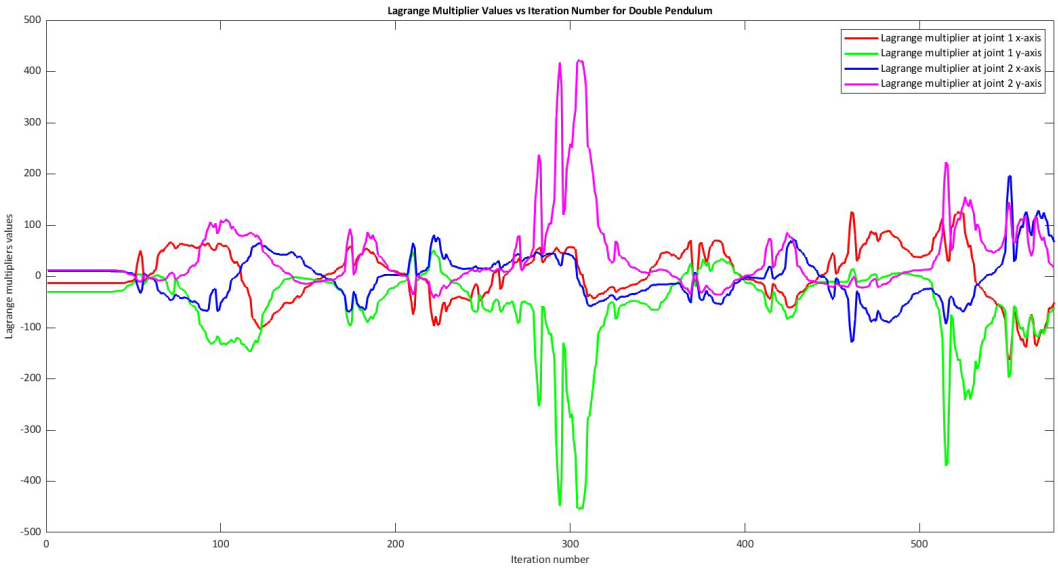

Fig. 5. Illustrative example of Lagrange multiplier values for double pendulums

Lagrange multipliers are essential in multibody dynamics for enforcing constraints without altering the equations of motion directly. In a double pendulum system, these multipliers ensure the lengths of the pendulum rods remain constant and the joints move correctly. By incorporating these constraints systematically, Lagrange multipliers maintain the physical accuracy and numerical stability of the simulation. This method allows for precise adherence to the system's constraints, leading to more reliable and accurate modelling of complex mechanical systems like the double pendulum.

The constant components of the constraint equations will vanish when they are differentiated twice over time. Therefore, to fulfil the constraint equations at a certain time, this value must be

identified as described by Eq. (13). A constraint stabilization method, such as penalty and augmented Lagrangian formulation, can be used to resolve this problem.

2.1.3 Penalty formulation

The penalty method introduces penalty terms into the equations of motion,

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_e + \mathbf{Q}_v - \boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\xi}\boldsymbol{\omega}_{\mathrm{n}}\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi}) \tag{16}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\omega}_n$ and $\boldsymbol{\xi}$ are diagonal matrices containing the values of penalty factor, natural frequencies and damping ratios, respectively, at each constraint condition. When penalty value $\boldsymbol{\alpha}$ is set to large, it will ensure the constraints within tight tolerances but might lead to numerical problems. The value selection for $\boldsymbol{\xi}$ and $\boldsymbol{\omega}_n$ can be varied as no general procedures are available to determine their values. Therefore, this approach may produce inconsequential results in certain cases.

2.1.4 Augmented Lagrangian relaxation method

The augmented Lagrangian method was developed to overcome the drawbacks of the penalty method and the Lagrange multiplier method [19]. As mentioned earlier, large penalty values will converge the constraints within a tight tolerance while it may lead to numerical illness and round-off error. In the augmented Lagrangian method, an iterative procedure was introduced to account for the constraints in the equations of motion [23]. In this method, Eq. (16) may turn to augmented by adding the Lagrange multipliers. The equations of motion can be written as:

$$\mathbf{M}\ddot{\mathbf{q}} + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\lambda} = \mathbf{Q}_e + \mathbf{Q}_v - \boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi}) \tag{17}$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier governing the new stable equation. As the Lagrange multipliers are steady enough to impose the constraints in the equations, the numerical value of the penalty coefficient does not need to be large. In this method, Lagrange multipliers are not assumed as unknowns but computed through an iterative process as:

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \boldsymbol{\alpha}(\ddot{\boldsymbol{\Phi}} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi})_{k+1} \tag{18}$$
$$k = 0,1,2,\ldots$$

where subscript k is the iteration number and the initial $\boldsymbol{\lambda}_0 = 0$. As the iterative solution solves the Lagrange multipliers, the augmented Lagrangian formulation leads to a system of ordinary differential equations without additional unknowns. The system of equations of motion, including the iterative scheme, can be written in the form of index-1 as follows:

$$(\mathbf{M} + \boldsymbol{\alpha}\boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\Phi}_{\mathbf{q}})\ddot{\mathbf{q}}_{k+1} = \mathbf{M}\ddot{\mathbf{q}}_k + \boldsymbol{\Phi}_{\mathbf{q}}^{\mathrm{T}}\boldsymbol{\alpha}(\boldsymbol{\Phi}_{\mathbf{q}t}\dot{\mathbf{q}} + \boldsymbol{\Phi}_{tt} + 2\boldsymbol{\xi}\boldsymbol{\omega}_n\dot{\boldsymbol{\Phi}} + \boldsymbol{\omega}_n^2\boldsymbol{\Phi}) \tag{19}$$

where for the initial iteration $\mathbf{M}\ddot{\mathbf{q}}_{k=0} = \mathbf{Q}_e + \mathbf{Q}_v$. The iterative process is repeated until the setting tolerance ε meets the form

$$\|\ddot{\mathbf{q}}_{i+1} - \ddot{\mathbf{q}}_i\| = \varepsilon \tag{20}$$

As this procedure involves an iteration loop, it may lead to extra computing time. The procedure is normally implemented by employing Newton-Raphson iteration method. A major advantage of using this approach is that the penalty factor values are no longer critical because they are treated by the Lagrange multipliers.

2.1.5 Sparsity in the equations of motion

Matrix **W**, vector **x** and vector **b** are the three main vector-matrix structures that are typically used to solve the linear equations of motion. The matrix representation of the equation of motion using Lagrange multipliers is as follows:

$$\underbrace{\begin{bmatrix} \mathbf{M} & \mathbf{\Phi}_{\mathbf{q}}^{\mathrm{T}} \\ \mathbf{\Phi}_{\mathbf{q}} & \mathbf{0} \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{Q}_e - \mathbf{Q}_v \\ \mathbf{Q}_c \end{bmatrix}}_{\mathbf{b}} \tag{21}$$

It is essential to account for the sparsity of matrix **W** in real-time when solving for $\ddot{\mathbf{q}}$ and $\boldsymbol{\lambda}$. The mass matrix **M** and the Jacobian matrix of constraints $\mathbf{\Phi}_{\mathbf{q}}$ typically have sparse qualities, as in Eq. (21). The dimension of matrix **M** may differ significantly between a planar and a spatial model. The total mass matrix for n bodies in a spatial model, where the mass matrix is diagonally oriented, is as follows:

$$\mathbf{M}_i = \begin{bmatrix} m_i\mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_i \end{bmatrix}_{6\times 6}, \quad \mathbf{M} = \mathrm{diag}[\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_{\mathrm{n}}]_{6n\times 6n} \tag{22}$$

It is possible to observe the sparsity of the Jacobian matrix of constraints $\mathbf{\Phi}_{\mathbf{q}} = \mathbb{R}^{n_c \times n_g}$. Moreover, two interconnected bodies are used to express the constrained joint equation. The number of constraint equations will rise to $5\mathrm{n}_r$ as the number of revolute joints n_r in the system rises. A large-scale restricted system will consequently have a sparse and asymmetric Jacobian matrix. The augmented matrix, **W**, will end up being incredibly sparse and symmetric despite the connection of the mass matrix, **M**, with the Jacobian matrix, $\mathbf{\Phi}_{\mathbf{q}}$ and their arrangement in matrix form as given in Eq. (21).

2.2 Recursive Formulation

For the relative coordinates between articulated bodies connected by joints, recursive formulation can create kinematic attributes such as position, virtual displacement, velocity and acceleration. The decreased dimensionality and degree of freedom of the system are often referred to as the dynamics of the recursive formulation. Figure 6 shows the technique of backward and forward motion in the recursive formulation in order to provide required elements for the equations of motion.

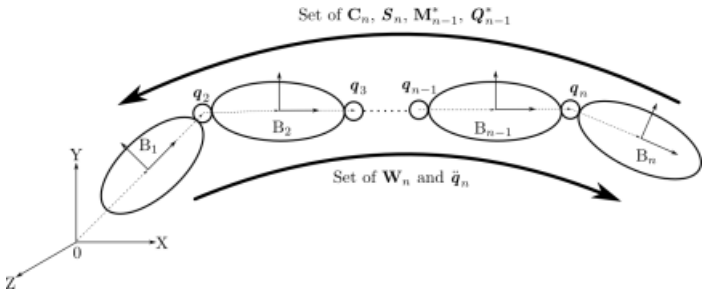

Fig. 6. Backward and forward movement approach

2.2.1 Kinematics

When building the system matrix and solving the equations of motion in the kinematics of the recursive formulation, the relative motion between constraints and nearby bodies is needed [24]. As in Figure 7, the basic arrangement of two interconnected bodies is indicated by a joint.

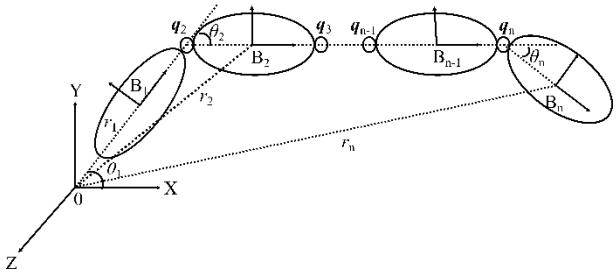

Fig. 7. Relationship of contiguous bodies

The body reference frame on the body B_0 is sequentially translated to the body B_n in order to acquire the orientation depicted in Figure 7 above. With point *0* acting as the global reference frame, the collection of bodies can be visualized as an open-loop chain. Below is the description of each body's relative kinematic position in a global reference frame:

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} + \mathbf{A}_{n-1}\overline{\boldsymbol{u}}_n \tag{23}$$

where $\mathbf{A}_{n-1}$ is the three-dimensional absolute rotation matrix from the previous body, B_{n-1} and $\boldsymbol{u}_n = [x_n y_n z_n]^T$ is the body, B_n's relative position vector with respect to the previous body, B_{n-1} describing $\boldsymbol{r}_n$ relative to point *0*. The absolute rotation matrix $\mathbf{A}_{n-1}$ is recognized as a multiplication of an n-body series whereby

$$\mathbf{A}_{n-1}{}^{*} = \underbrace{\begin{bmatrix} \cos(\theta_{n-1}) & -\sin(\theta_{n-1}) & 0 \\ \sin(\theta_{n-1}) & \cos(\theta_{n-1}) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{z-axis} \tag{24}$$

The time derivative of Eq. (23) can be used to calculate joint velocity such that

$$v_n = \dot{\boldsymbol{r}}_n \tag{25}$$

The relative angular velocity of the body B_n with regard to the body B_{n-1} is given by angular velocity, ω_n. The function of joint relative coordinate q_n and joint relative velocity, $\dot{\boldsymbol{q}}_n$, denoted by ω_n can be expressed as follows:

$$\omega_n = \omega_{n-1} + \boldsymbol{B}_n \dot{\boldsymbol{q}}_n \tag{26}$$

$\boldsymbol{B}_n$ is the unit vector of the joint rotational axis whereas $\boldsymbol{B}_n = [0\ 0\ 1]^T$ as shown in Figure 5. By merging v_n and ω_n into one form, the velocity relationship between the bodies can be calculated.

$$\begin{Bmatrix} \dot{\boldsymbol{r}}_n \\ \omega_n \end{Bmatrix} = \begin{bmatrix} \mathrm{I} & -\widetilde{\boldsymbol{u}}_n \\ 0 & \mathrm{I} \end{bmatrix} \begin{Bmatrix} \dot{\boldsymbol{r}}_{n-1} \\ \omega_{n-1} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{H}_n \\ \boldsymbol{B}_n \end{Bmatrix} \dot{\boldsymbol{q}}_n \tag{27}$$

To express the total velocity, Eq. (27) can be compressed to the extent such that,

$$\mathbf{V}_{\boldsymbol{n}} = \mathbf{C}_{\boldsymbol{n}} \mathbf{V}_{\boldsymbol{n-1}} + \boldsymbol{S}_{\boldsymbol{n}} \dot{\boldsymbol{q}}_{\boldsymbol{n}} \tag{28}$$

Also, the acceleration of each body can be formed by applying second time derivative of Eq. (28).

$$\mathbf{W}_n = \mathbf{C}_n \mathbf{W}_{n-1} + \boldsymbol{S}_n \ddot{\boldsymbol{q}} + \mathbf{D}_n \tag{29}$$

where $\mathbf{D}_n = \dot{\boldsymbol{S}}_n \dot{\boldsymbol{q}}_n$.

2.2.2 Dynamics

Following to the kinematics of the elementary contiguous bodies, the equation of motion for rigid body system can be constructed. Since the total inertia and force in an open loop system can be expressed as

$$\sum_{i=0}^{n} \boldsymbol{M}_i \boldsymbol{W}_i - \boldsymbol{Q}_i + \cdots + (\boldsymbol{M}_{n-2} \boldsymbol{W}_{n-2} - \boldsymbol{Q}_{n-2}) + (\boldsymbol{M}^*_{n-1} \boldsymbol{W}_{n-1} - \boldsymbol{Q}^*_{n-1}) = 0 \tag{30}$$

whereby $\mathbf{M}^*_{n-1}$ is the sum of mass matrices of bodies $\mathbf{B}_{n-1}$ and $\mathbf{B}_n$,

$$\boldsymbol{M}^*_{n-1} = \boldsymbol{M}_{n-1} + {\boldsymbol{C}_n}^T \boldsymbol{M}_n \boldsymbol{C}_n - {\boldsymbol{C}_n}^T \boldsymbol{M}_n \boldsymbol{C}_n \boldsymbol{U}_n^{-1} {\boldsymbol{S}_n}^T \boldsymbol{M}_n \boldsymbol{C}_n \tag{31}$$

and $\mathbf{Q}^*_{n-1}$ is the total inertia force matrices at bodies $\mathbf{B}_{n-1}$ and $\mathbf{B}_n$,

$$\boldsymbol{Q}^*_{n-1} = \boldsymbol{Q}_{n-1} - {\boldsymbol{C}_n}^T (\boldsymbol{M}_n \big(\boldsymbol{S}_n \boldsymbol{U}_n^{-1} {\boldsymbol{S}_n}^T (\boldsymbol{Q}_n - \boldsymbol{M}_n \boldsymbol{D}_n) + \boldsymbol{D}_n \big) - \boldsymbol{Q}_n) \tag{32}$$

By repeating the above approach from body $\mathbf{B}_{n-1}$ to B_0, the base of the equation of motion can be determined such that,

$$\mathbf{M}_0 \mathbf{W}_0 - \mathbf{Q}_0 = 0 \tag{33}$$

However, if there is no constraint applied to the base body, B_0, therefore,

$$\mathbf{M}_0 \mathbf{W}_0 = -\mathbf{Q}_0 \tag{34}$$

2.2.3 Algorithm

For an open-loop system, a general description of the algorithm that produces the equation of motion is given by Figure 8 below:

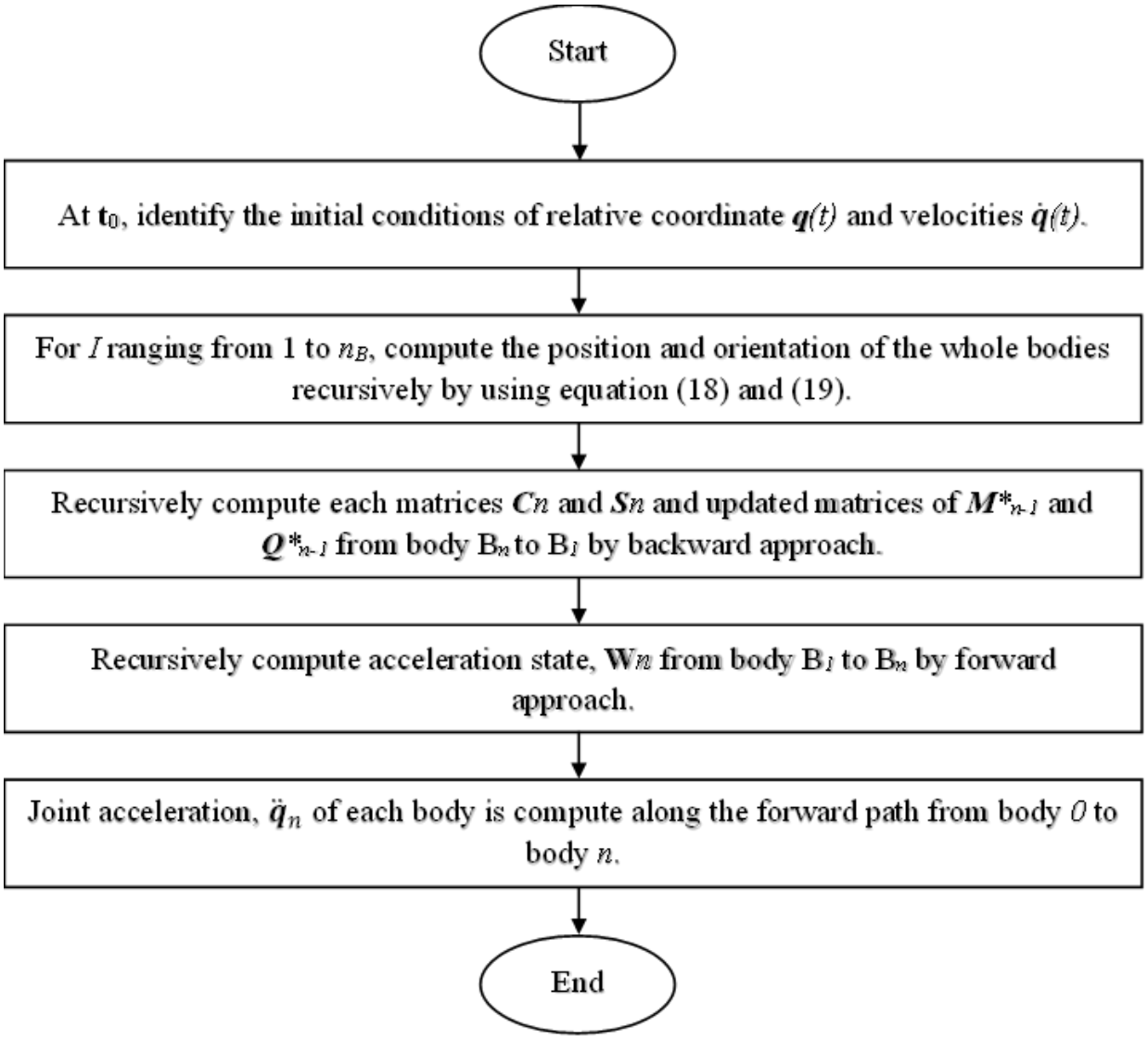

Fig. 8. General algorithm for equations of motion

Following to the last step, the time integration can be used to calculate the relative coordinates. From the position, $\boldsymbol{r}_{n-1}$ and velocity, $\boldsymbol{v}_{n-1}$ of the preceding adjacent body B_{n-1}, one may determine the position, $\boldsymbol{r}_n$ and velocity, $\boldsymbol{v}_n$ of the body, B_n. In summary, adopting the recursive formulation for dynamic analysis gives two substantial computing benefits: establishing a small set of differential equations and a simple numerical approach for solving the dynamic equations. Nevertheless, the recursive formulation has one disadvantage, though: the dynamic equations are represented in terms of a group of joint variables that are reliant on the multibody system's topological structure.

3. Numerical Solution

In this study, a MATLAB function based on an explicit Runge-Kutta of Eq. (4) and Eq. (5) specifically the Dormand-Prince pair known as ODE45 is used for solving ordinary differential equations (ODEs).

It mainly focuses on balancing accuracy with computational efficiency hence suitable for complex dynamic systems. In the case of numerical analysis of multibody dynamics like double pendulums with Lagrange multipliers and semi-recursive methods, ODE45 uses a set system of ODEs that include equations of motion and constraints and has initial conditions like positions, velocities and state variables. In general, the following equation can be handled by the ODE45 solver:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(t, \boldsymbol{x}),\ \boldsymbol{x}(\boldsymbol{t_0}) = \boldsymbol{x_0} \tag{35}$$

where t is the independent variable, $\boldsymbol{x}$ is a vector of dependent variables to be found and $\boldsymbol{f}(t, \boldsymbol{x})$ is a function of t and $\boldsymbol{x}$. The mathematical can be solved by setting the $\boldsymbol{f}(t, \boldsymbol{x})$ and the initial conditions, $\boldsymbol{x} = \boldsymbol{x}_0$ at time t_0 is known.

The iteration process in ODE45 starts by computing an initial step size which is based on local error estimates from the Dormand-Prince method. For every iteration, ODE45 does several sub-steps such as calculating intermediate values of dependent variables using Runge-Kutta coefficients, combining these intermediate values to estimate the solution at the end of the step and estimating local errors by comparing fourth-order solutions with fifth-order accurate solutions. This error estimation determines how much to adjust the step size. If this error exceeds or equals a given tolerance value then one reduces the step size while recalculating this current step. On the other side, if the error is much smaller, then step size increases resulting in better computational efficiency. This dynamic step control ensures that the solver maintains accuracy without performing any useless calculations hence makes it an efficient tool for handling stiff or complex systems.

The ODE45 solver occasionally generates results that are unnecessary or unexpected. Hence, it is essential to properly manage both relative and absolute error control tolerances in order to generate precise numerical results. Absolute tolerance (AbsTol) refers to the smallest value of the i^{th} solution component that is judged superfluous, whereas relative tolerance (RelTol) assesses the error in respect to the size of each component of the solution. RelTol guarantees that the entire component of the solution has a certain number of correct digits, except for those that are less accurate than the AbsTol thresholds, which by default are set at 0.001 or an accuracy of 0.1%. When the solution is close to zero, the accuracy is governed by the absolute tolerances.

The ODE45 iteratively solves the coupled equations of motion and constraints in the context of multibody dynamics with Lagrange multipliers. The adjustment of Lagrange multipliers by the solver ensures that constraints are satisfied at each time level. ODE45 achieves constraint enforcement without loss of numerical stability by updating variables with each timestep and repeating this process until the entire time interval is covered. For double pendulum type problems that have rapid changes in motion, adaptive step size control implemented by ODE45 provides a solution method capable of producing accurate and efficient simulations for such systems. The ODE45 solver is suitable as an alternative for any non-stiff equation as it can balance the accuracy and computational cost [25]. The fourth order of the Runge-Kutta method is widely used as it can be easily tolerated with changing step sizes, Δt [26].

4. Results and Discussion

In this work, the articulated pendulum made up of twenty pendulums was examined in order to compare for their mathematical formulations and numerical techniques. A desktop Asus PC with an Intel Core i5-4200U CPU running at 1.60 GHz and 2.30 GHz was used to conduct the whole analysis. A representation of the simulated twenty articulated pendulum is shown in Figure 9.

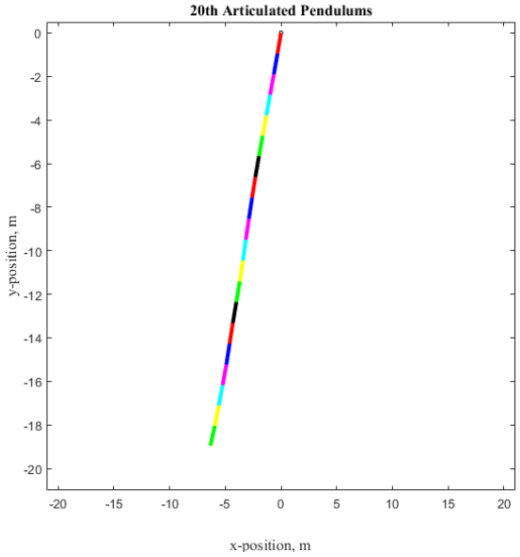

Fig. 9. Twenty articulated pendulums

The fourth and fifth-order Runge-Kutta method was used to integrate the equations of motion while regulating the step size and both RelTol and AbsTol tolerance. The numerical calculations were carried out using the ode45 solver. With a maximum solution time of 5 seconds, the computation times for both formulations were recorded. The results are shown in Table 1.

Table 1
Tolerance and step size of recursive and augmented Lagrangian comparison

Tolerance (s)	Step size (s)	Computing time (s)	
		Recursive	Augmented Lagrangian
0.1	0.1	4.74	6.65
	0.01	4.57	6.83
	0.001	4.67	7.34
	0.0001	5.96	10.07
0.01	0.1	7.02	6.83
	0.01	7.04	6.65
	0.001	7.06	7.11
	0.0001	9.13	10.16
0.001	0.1	7.19	6.42
	0.01	6.85	6.09
	0.001	6.84	6.68
	0.0001	9.05	9.75
0.0001	0.1	9.85	7.93
	0.01	10.59	7.57
	0.001	10.59	8.61
	0.0001	11.65	10.72

The recursive method outperforms the augmented Lagrangian method in terms of computing time when the tolerance is set at 0.1s, as in Figure 10. Both numerical techniques achieved significant findings at tolerances of 0.01s and 0.001s. The augmented method however computes faster than the recursive method when the tolerance approaches 0.0001s. The recursive formulation should be more computationally efficient since it needed less processing time for dynamic simulations [27]. This might be caused by the augmented formulation's separability of the mass matrix from the ode function makes the simulation compute faster than the recursive formulation.

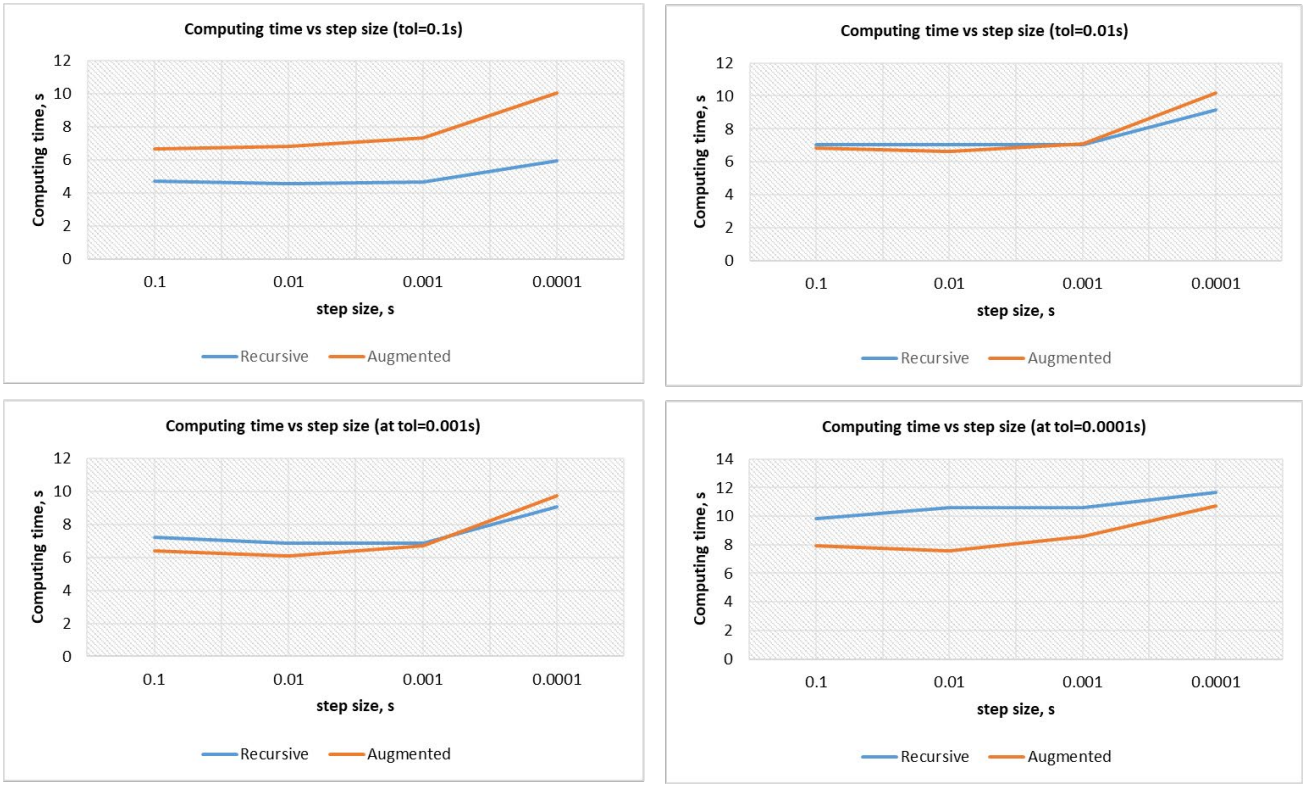

Fig. 10. Computing time vs step size for recursive and augmented Lagrangian method

In Figure 11, the augmented Lagrangian method's xy-position shows a noticeable tilt towards the vertical axis at tolerance 0.1s, but the other tolerances show a stable result at a constant step size. On the other hand, the xy-position of the recursive method shows a significant pattern and results for all stepsizes and regulated tolerances. While the rest of the data show optimal animation at any stepsize, the animation of the xy-position of the augmented Lagrangian technique exhibits an unstable motion, with elongation occurring at a specific number of pendulums when the tolerance is set to 0.1s. The recursive technique, however, produces a clear and stable animation for any step size and tolerance.

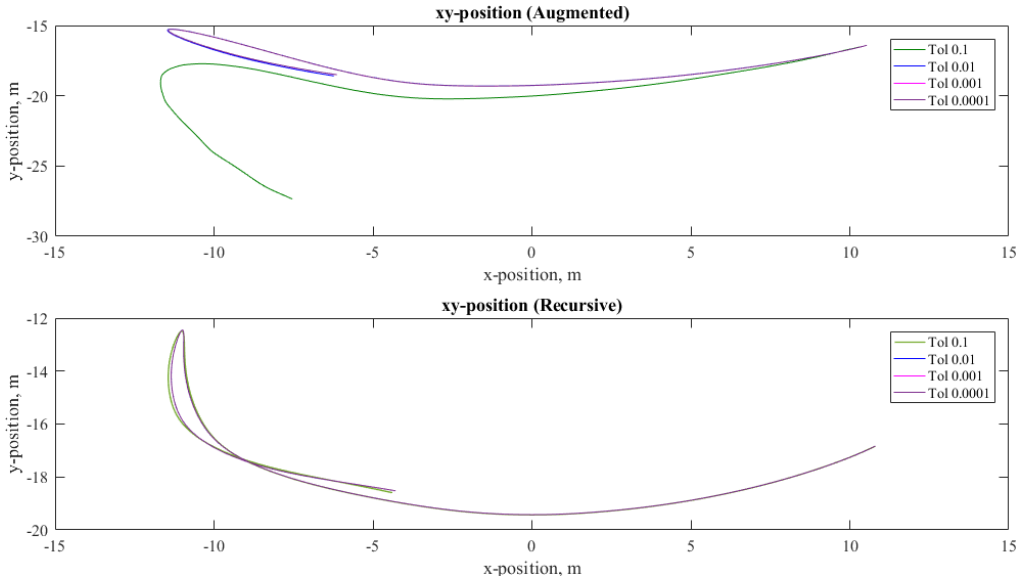

Fig. 11. Position of end tips of pendulum 20th at step size 0.1s

At all controlled step sizes of 0.1s, 0.01s, 0.001s and 0.0001s shown in Figure 12, the augmented Lagrangian showed unstable motions at a tolerance of 0.1s, in terms of velocity-time graph. The graph, however, started to resemble reliable result when the tolerance was lowered to 0.01s at any controlled step size. In contrast, the velocity-time graphs showed a little variation for recursive at a tolerance of 0.1s at any step size. However, as the tolerance tightened to 0.0001s, the patterns in the graphs gradually stabilized and significant.

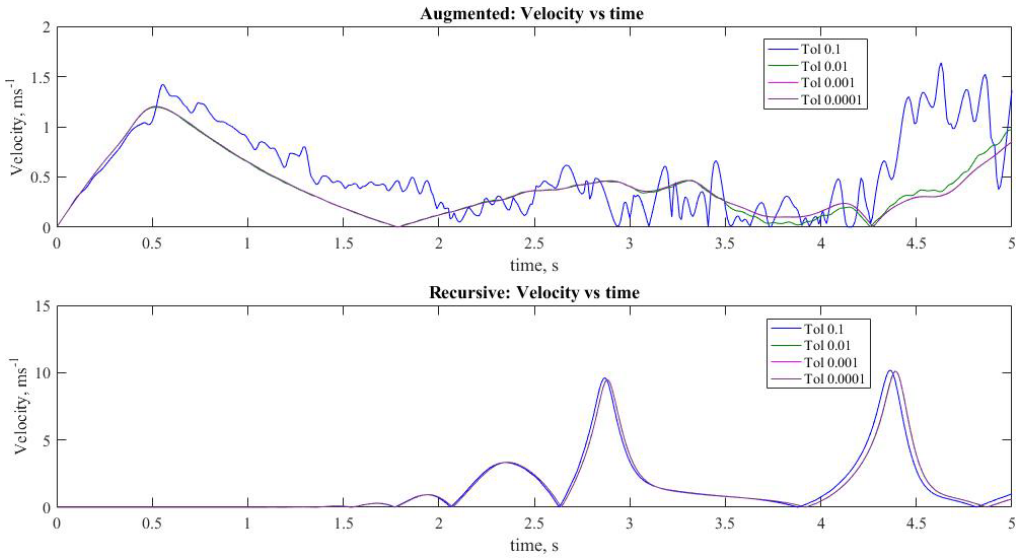

Fig. 12. Velocity-time graph at step size=0.01s

We have previously examined the computational time efficiency of the two formulations by varying the step size specified in Noh *et al.,* [28] over a number of pendulums, n = 2, 4, 8, 16, 20 and 40. The pendulums' performances were assessed with a maximum simulation time of 5 seconds and a constant tolerance of 0.01s. Taking the findings for step size 0.0001s, as shown in Table 2, the augmented Lagrangian computed faster than the recursive formulation for a lesser number of pendulums at n = 2, 4, 8 and 16. The recursive formulation, however, performed better in computing time than the augmented Lagrangian when we increased the number of pendulums to 20 and 40. This discovery validates and coincides with our present findings.

Table 2
Variability of pendulum's number vs formulations

No. of Pendulum	Formulations	
	Augmented Lagrangian	Recursive
2	2.4622	2.4636
4	2.6025	3.1051
8	3.5003	4.3666
16	7.0528	7.7205
20	10.0302	9.6861
40	32.6064	26.4762

5. Conclusion

The objective of the study was to examine the computation times of the augmented Lagrangian and recursive formulations based on the modified step size and tolerance in handling huge matrices. The step size and tolerance were controlled, as described in Table 1, to see for any differences in the

articulated systems for both formulations despite of the computing time results. The results showed that the augmented Lagrangian formulation presented a considerable negative y-axis deflection when the tolerance was set at 0.1s at any step size. Moreover, an elongation appeared at certain pendulums reflecting unusual and inconsistency in the animation patterns at step size 0.1s. The recursive formulation, however, showed a better fit for all xy-positions and velocity-time graphs. Consequently, it may be concluded that the recursive formulation is more effective than the augmented Lagrangian formulation in solving for large open-loop articulated systems.

Acknowledgment
This research was funded by a grant from Ministry of Education Malaysia (FRGS Grant FRGS/1/2018/TK03/UNIMAP/02/19).

References

[1] Fulton, E. L. and T. J. Gay. "Does a physical pendulum ever act like a simple pendulum?." *European Journal of Physics* 45, no. 2 (2024): 025001. https://doi.org/10.1088/1361-6404/ad1f95
[2] Bondada, Aditya and Vishnu G. Nair. "Dynamics of multiple pendulum system under a translating and tilting pivot." *Archive of Applied Mechanics* 93, no. 9 (2023): 3699-3740. https://doi.org/10.1007/s00419-023-02473-6
[3] Amer, T. S., Abdallah Galal and A. F. Abolila. "On the motion of a triple pendulum system under the influence of excitation force and torque." *Kuwait Journal of Science* 48, no. 4 (2021). https://doi.org/10.48129/kjs.v48i4.9915
[4] Zielinska, Teresa, Gabriel R. Rivera Coba and Weimin Ge. "Variable inverted pendulum applied to humanoid motion design." *Robotica* 39, no. 8 (2021): 1368-1389. https://doi.org/10.1017/S0263574720001228
[5] Jaafar, H. I., Z. Mohamed, M. A. Shamsudin, NA Mohd Subha, Liyana Ramli and A. M. Abdullahi. "Model reference command shaping for vibration control of multimode flexible systems with application to a double-pendulum overhead crane." *Mechanical Systems and Signal Processing* 115 (2019): 677-695. https://doi.org/10.1016/j.ymssp.2018.06.005
[6] Ouyang, Huimin, Zheng Tian, Lili Yu and Guangming Zhang. "Adaptive tracking controller design for double-pendulum tower cranes." *Mechanism and Machine Theory* 153 (2020): 103980. https://doi.org/10.1016/j.mechmachtheory.2020.103980
[7] Staab, Harald, Christoph Byner, Debora Clever and Bjoern Matthias. "A Pendulum Apparatus to Evaluate Unconstrained Human-Robot Contact." In *ISR 2020; 52th International Symposium on Robotics*, pp. 1-8. VDE, 2020.
[8] Wang, Tao. "Pendulum-based vibration energy harvesting: Mechanisms, transducer integration and applications." *Energy Conversion and Management* 276 (2023): 116469. https://doi.org/10.1016/j.enconman.2022.116469
[9] Ambrożkiewicz, Bartłomiej, Grzegorz Litak and Piotr Wolszczak. "Modelling of electromagnetic energy harvester with rotational pendulum using mechanical vibrations to scavenge electrical energy." *Applied Sciences* 10, no. 2 (2020): 671. https://doi.org/10.3390/app10020671
[10] Zaouali, Emine, Fehmi Najar, Najib Kacem and Emmanuel Foltete. "Pendulum-based embedded energy harvester for rotating systems." *Mechanical Systems and Signal Processing* 180 (2022): 109415. https://doi.org/10.1016/j.ymssp.2022.109415
[11] Hazem, Zied Ben and Zafer Bingül. "Comprehensive review of different pendulum structures in engineering applications." *IEEE Access* 11 (2023): 42862-42880. https://doi.org/10.1109/ACCESS.2023.3269580
[12] Rodrigues da Silva, Mariana, Filipe Marques, Miguel Tavares da Silva and Paulo Flores. "An improved methodology to restrict the range of motion of mechanical joints." *Nonlinear Dynamics* 112, no. 6 (2024): 4227-4256. https://doi.org/10.1007/s11071-023-09208-w
[13] Featherstone, Roy. *Rigid body dynamics algorithms*. Springer, 2014.
[14] Rui, Xiaoting, Jianshu Zhang, Xun Wang, Bao Rong, Bin He and Zhan Jin. "Multibody system transfer matrix method: the past, the present and the future." *International Journal of Mechanical System Dynamics* 2, no. 1 (2022): 3-26. https://doi.org/10.1002/msd2.12037
[15] Antonya, Cs and Razvan Gabriel Boboc. "Computational efficiency of multi-body systems dynamic models." *Robotica* 39, no. 12 (2021): 2333-2348. https://doi.org/10.1017/S0263574721000345
[16] Mahbouba, Yousif Jamal and Mohammad Qasim Abdullah. "Numerical Solution of Linear and Nonlinear Swinging Pendulum." Journal of Scientific and Engineering Research, 4 (5) (2017):102-9.

[17] Karim, Ahmad Abdul, Thibaut Gaudin, Alexandre Meyer, Axel Buendia and Saida Bouakaz. "Adding physical like reaction effects to skeleton-based animations using controllable pendulums." In *Transactions on Edutainment VI*, pp. 111-121. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. https://doi.org/10.1007/978-3-642-22639-7_12
[18] Saab, Wael, William S. Rone and Pinhas Ben-Tzvi. "Robotic tails: a state-of-the-art review." *Robotica* 36, no. 9 (2018): 1263-1277. https://doi.org/10.1017/S0263574718000425
[19] Malczyk, Paweł, Janusz Frączek, Francisco González and Javier Cuadrado. "Index-3 divide-and-conquer algorithm for efficient multibody system dynamics simulations: theory and parallel implementation." *Nonlinear Dynamics* 95, no. 1 (2019): 727-747. https://doi.org/10.1007/s11071-018-4593-3
[20] Yu, Xinxin andreas Zwölfer and Aki Mikkola. "An efficient, floating-frame-of-reference-based recursive formulation to model planar flexible multibody applications." *Journal of Sound and Vibration* 547 (2023): 117542. https://doi.org/10.1016/j.jsv.2022.117542
[21] Rahikainen, Jarkko, Francisco González, Miguel Ángel Naya, Jussi Sopanen and Aki Mikkola. "On the cosimulation of multibody systems and hydraulic dynamics." *Multibody System Dynamics* 50, no. 2 (2020): 143-167. https://doi.org/10.1007/s11044-020-09727-z
[22] Cheng, Peng, Jumat Sulaiman, Khadizah Ghazali, Majid Khan Majahar Ali and Ming Ming Xu. "Newton-SOR Iterative Method with Lagrangian Function for Large-Scale Nonlinear Constrained Optimization Problems." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 46, no. 2 (2025): 251-262. https://doi.org/10.37934/araset.46.2.251262
[23] Burman, Erik, Peter Hansbo and Mats G. Larson. "The augmented Lagrangian method as a framework for stabilised methods in computational mechanics." *Archives of Computational Methods in Engineering* 30, no. 4 (2023): 2579-2604. https://doi.org/10.1007/s11831-022-09878-6
[24] Jaiswal, Suraj, Jussi Sopanen and Aki Mikkola. "Efficiency comparison of various friction models of a hydraulic cylinder in the framework of multibody system dynamics." *Nonlinear Dynamics* 104, no. 4 (2021): 3497-3515. https://doi.org/10.1007/s11071-021-06526-9
[25] Emagbetere, E., O. Oluwole and T. A. O. Salau. "Comparative study of Matlab ODE solvers for the Korakianitis and Shi model." *Bulletin of Mathematical Sciences and Applications* 19 (2017): 31-44. https://doi.org/10.18052/www.scipress.com/BMSA.19.31
[26] Salau, T. A. O. and Kayode Moses Akin. "Simulation of Harmonically Excited Simple Linear Pendulum (HESLP) Model using Selected Versions of Fourth Order Runge-Kutta Schemes." *International Journal of Engineering Science* 14334 (2017).
[27] Slaats, Paul MA. "Recursive formulations in multibody dynamics." (1991).
[28] Noh, SFA Ahmad, Mohamad Ezral Baharudin, Azuwir Mohd Nor, Mohd Zakimi Zakaria and Mohd Sazli Saad. "Numerical Modeling Methods for Large Open Loop Multibody System."