



ISSN 1823-626X

Malaysian Journal of Fundamental and Applied Sciences

available online at <http://mjfas.ibnusina.utm.my>



Effect of Time Complexities and Variation of Mass Spring Model Parameters on Surgical Simulation

Salina Sulaiman^{1*}, Tan Sing Yee² and Abdullah Bade¹

¹Real-time Computer Graphics and Visualization Group (GRAVS), School of Science and Technology, UMS, 88400 Kota Kinabalu, Sabah, Malaysia

²Department of Electrical and Computer Engineering, School of Engineering and Science, Curtin University, CDT 250, 98009, Miri, Sarawak, Malaysia

Received 8 January 2013, Revised 30 March 2013, Accepted 16 April 2013, Available online 20 May 2013

ABSTRACT

Physically based models assimilate organ-specific material properties, thus they are suitable in developing a surgical simulation. This study uses mass spring model (MSM) to represent the human liver because MSM is a discrete model that is potentially more realistic than the finite element model (FEM). For a high-end computer aided medical technology such as the surgical simulator, the most important issues are to fulfil the basic requirement of a surgical simulator. Novice and experienced surgeons use surgical simulator for surgery training and planning. Therefore, surgical simulation must provide a realistic and fast responding virtual environment. This study focuses on fulfilling the time complexity and realistic of the surgical simulator. In order to have a fast responding simulation, the choice of numerical integration method is crucial. This study shows that MATLAB ode45 is the fastest method compared to 2nd ordered Euler, MATLAB ode113, MATLAB ode23s and MATLAB ode23t. However, the major issue is human liver consists of soft tissues. In modelling a soft tissue model, we need to understand the mechanical response of soft tissues to surgical manipulation. Any interaction between haptic device and the liver model may causes large deformation and topology change in the soft tissue model. Thus, this study investigates and presents the effect of varying mass, damping, stiffness coefficient on the nonlinear liver mass spring model. MATLAB performs and shows simulation results for each of the experiment. Additionally, the observed optimal dataset of liver behaviour is applied in SOFA (Simulation Open Framework Architecture) to visualize the major effect.

| Surgical Simulation | Mass-Spring Model | Numerical Integration Methods | Simulation Open Framework Architecture | Time Complexities |

© 2013 Ibnu Sina Institute. All rights reserved.

<http://dx.doi.org/10.11113/mjfas.v9n4.105>

1. INTRODUCTION

Surgical simulation is proposed for the needs to complement the traditional surgical training methods. With the advances in computer technology, virtual environment (VE) is introduced into entertainment, experiments, flight simulators, and health care. VE is crucial in surgical simulation as it combines a convincing representation of soft tissues where interaction between the end user and the virtual objects are allowed. VE increases the immersive of the end user by the generating realistic virtual objects. By applying virtual reality into the simulators, training in surgical operation will decrease the pressure of trainees as they are allowed to make mistakes. This realistic learning environment enhances the interest of the student learning enthusiasm. Meanwhile, this training system is much cheaper in terms of cost and time. However, surgical simulation is challenging [8]. Although the surgical simulation training is proven to be more useful than traditional surgical training, the accuracy assessment of the simulated surgical outcome is still difficult because soft tissues are complex, nonlinear, viscous-elastic, anisotropic and time-dependent [16]. Moreover, Bianchi et al [2] stated that typically to represent an organ geometrically, at least 1500 nodes or roughly 3000 elements are needed. Human tissue modelling is one of the most demanding among the

potential applications of deformable objects due to the complexity exhibited by the soft tissues [8, 12]. Therefore, a suitable deformable model is essential to represent the exact properties of the soft tissues.

Surgical simulation is a physically-based simulation which involves the deformation of the modelled object. Therefore, physically-based modelling is suitable to represent the deformable objects by using techniques such as the Finite Element Model (FEM), Mass-Spring Model (MSM), Finite Difference Model (FDM), Boundary Element Model (BEM) and others. In this study, the human liver is chosen to model by nonlinear MSM. MSM is a discrete model, known as the particle system with special interaction forces. There are four steps toward the simulation based on MSM. Firstly, spatial discretization where the deformable object is sampled in mass points. Secondly, identify the forces interacting between the mass points in the deformable objects. The forces are represented by the Newton 2nd Law and the Hooke's Law of the springs. Thirdly, govern a mathematical model which represents the dynamics in the deformation. Lastly, solve the temporal discretization in a system of ordinary differential equations (ODE) using the numerical integration solvers. There are several ODEs to solve the discretized system. However, it is crucial to solve the system with a fast converging ODE

*Corresponding author. E-mail: salina@ums.edu.my

solver in order to fulfil the requirement of realistic and real-time interaction for a surgical simulation.

In modelling the human liver structure in MSM, there are several important properties of the liver that we are concerned with. Human liver consists of certain properties such as absorption, scattering, anisotropy, penetration depth, nonlinear and viscous-elasticity. However, only two properties will be considered in this study. They are the nonlinearity and viscous-elasticity.

This study focuses on time complexity against the CPU runtime of different MATLAB ODE solvers onto the specified material properties of the human liver which is modelled by MSM and performance of a MSM under various conditions. Furthermore, observation from different damping and stiffness coefficients which will be manually tuned in SOFA to visualize the effect caused by varying the damping and stiffness coefficients will be recorded.

This paper is organized as follows; methodology of this study is presented in Section II. The described model is then applied to findings in Section III. Finally, conclusions, limitation and future work are discussed in Section IV.

2. METHODOLOGY

2.1 Mass Spring Model (MSM)

MSM is the simplest model. It begins with a discrete model instead of with a system of partial differential equation that is complicated. This model consists of meshes of mass, spring and damper elements [11]. Each of the mass point links together by a network of zero mass springs. Since MSM is a discrete model, it is fast and computationally efficient. Therefore, it is useful in representing the nonlinear liver model and simulate in interactive speeds. MSM is one of the earliest models used in computer graphics. Waters [9] used a static MSM for facial modelling. Meanwhile, when real-time deformable model was first introduced into virtual surgical simulation, Cover et al [5] applied a simple mass spring model to simulate deformation of a gall bladder that is related to liver. However, the earlier works on MSM are mostly limited to 2D modelling and 3D rigid object modelling. Phannurat et al [13] presented a method of constraining physically based deformable objects where an object can be defined locally in terms of kinetic and dynamic (mass, position, velocity), and physical parameters (compressibility and elasticity). Christensen [4] on the other hand, presented the application of MSM for automatic motion synthesis, where there are no “flexion springs” considered for bending. A flexion spring is a spring that is in a state of being flexed as of a joint, a bodily function. Various researches efforts focused on improving the computational performance and accuracy of the MSM. Maciel et al [10] had used diagonal springs in applying the mass spring system for the computational method to simulate soft tissues. It is the generalized MSM, named molecular model which consists of mass points, and elastic forces established between the molecules by a spring-like connection. Diagonal springs were to avoid

tetrahedral meshes. Even though it offers the desired results but it creates difficulty in terms of computing spring constants. Zhang et al [18] optimized the traditional mass-spring system by adding a curvature force that intended to control the degree of bending and twisting of soft tissues. The angular springs produced curvature force. Meanwhile, Phannurat et al [13] applied MSM to the triangular meshes that represent the soft tissue model and each mass point linked to its neighbours by mass springs of non-zero natural length. This method benefits the real-time computation and deformation.

2.2 Numerical Integration Method

The two families of integration methods are explicit and implicit. They can be divided into multistep and unistep. In the calculation of multistep, it needs two or more previous values to compute the next system state. In this paper, the MATLAB ode113 is the only multistep explicit method. MATLAB ode45 and 2nd Ordered Euler are both unistep explicit. While MATLAB ode23s and MATLAB ode23t are both unistep implicit. Explicit integration is simple and straight forwarded. It computes the system state from the previous system state but demands on small step size thus makes it inefficient for stiff system. Meanwhile, implicit integration is more complicated than explicit integration. The next system state is calculated from the current system state and next system state itself. It allows large time steps therefore increases the stability of the system. However, it is not computationally efficient.

The studies of the numerical integration methods for the deformable objects simulation has begun since the 90's when Terzopoulos et al [15] developed new ideas in the usage of elasticity properties to model deformable objects. He has used the implicit integration method to solve the resultant system of equations. Later, due to the ease of implementation and reduced computation per simulation time steps of the explicit integration method. Many researchers have selected explicit integration methods to create real-time simulations. Provot [14] in the research on deformation constraints of a mass-spring model to describe rigid cloth behaviour chose the explicit Euler integration methods. However, after the work of Baraff and Witkin [1] the popularity of implicit integration method increased again. The implementation of implicit integration for the mass-spring models in cloth simulation has avoided the instability of explicit integration method. Nevertheless, there exists contradiction in implicit integration method. That is, although implicit integration methods are unconditionally stable, at the same time, it is computationally expensive too. The issue of this implicit method is that the time steps have increased, accompanied with the requirement of larger storage space. In order to overcome this drawback of implicit integration method, implicit-explicit (IMEX) integration method is being developed by Eberhardt et al [6]. The specialty of this method is that the stiff parts of the system of equations are calculated by using implicit method while the non-stiff part is done by explicit method. Furthermore, Volino and Magnenat [17] compared different numerical integration

methods in cloth simulation. The main idea to compare the numerical integration methods is to enable user to choose the adequate method with complete knowledge of the advantages and drawbacks of the main techniques. Concurrently, with the demanding needs in surgical simulation, there have been many research works related to medical field. Hauth et al [7] introduced a stable integration method that combines the advantages of explicit methods with the enhanced stability of implicit methods. Meanwhile, Celine et al [3] adopted an explicit integration scheme in their model. Zhu et al [19] used the fourth-order Runge-Kutta method as the numerical integrator of the mass-spring model. Apart from that, Liang et al [8], proposed a novel mixed numerical integral algorithm for the deformation simulation of soft tissues in virtual surgery system. A combination of Euler’s method and Runge-Kutta method are proposed in order to improve the performance of the simulation by balancing the complexity and accuracy.

2.3 Modelling Nonlinear Liver MSM System

In modelling the human liver, we assume that the MSM is free of damping. However, in actuality, all the deformation is damped to some degree by friction forces. These forces can be caused by fluid friction or internal friction between the molecules. The equation of motion for viscously damped free is as below:

$$m\ddot{x} + c\dot{x} + k_1x = 0 \dots (2.1)$$

m is the mass of the human liver, c is the damping coefficient, k_1 is the internal stiffness of the spring, x , \dot{x} and \ddot{x} are the position, velocity and acceleration of the corresponding mass point. Equation 1 is an unforced linear MSM. This model will be used as a case study 1.

In order to represent the human liver which is nonlinear and viscous-elasticity, we need to form a nonlinear equation of motion. The equation is as below:

$$m\ddot{x} + c\dot{x} + k_1x + k_3x^3 = 0 \dots (2.2)$$

Here, the nonlinear MSM consist of two kinds of stiffness, which are the internal stiffness, k_1 and k_3 external stiffness. In this equation of motion, there is no force exerted. Hence, it is an unforced nonlinear liver model. This model will be used as a case study 2.

To represent a dynamic forced nonlinear liver model, a periodic force P of magnitude P is added to the right hand side (RHS) of equation 2.2.

$$m\ddot{x} + c\dot{x} + k_1x + k_3x^3 = P \cos(\omega t) \dots (2.3)$$

where m is the mass of the human liver, c is the damping coefficient, k_1 and k_3 are the internal and external stiffness of the spring, x , \dot{x} and \ddot{x} are the position, velocity and acceleration of the corresponding mass point while $P \cos(\omega t)$ is the periodic force P . This model is the full

nonlinear liver model where the springs and damping are all in nonlinear form.

2.4 Linearization of Dynamic Equation of Nonlinear Liver Model

There are five methods selected to compare and show that ode45 is the fastest method in solving the ordinary differential equation. However, we first have to introduce linearization for the second order differential equation above in order for us to implement the chosen ode solvers using the pair of derived first order differential equation.

Let equation 2.3 be the equation of motion that represents the nonlinear viscous-elasticity liver model.

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + k_1x + k_3x^3 = P \cos(\omega t) \dots (2.4)$$

Set the initial value for velocity equals to 1 and displacement equals to 0.5, with fixed values for damping coefficient of 2, internal stiffness of 0.7, external stiffness of 0.8, and Periodic force of 0.5 with the shearing angle of 0.5, respectively. In addition, average mass of the human liver is 1.45kg. Substitute these initial values into equation 2.4, we have:

$$1.45 \frac{d^2x}{dt^2} + 2 \frac{dx}{dt} + 0.7x + 0.8x^3 = 0.5 \cos(0.5t) \dots (2.5)$$

To linearize the second order ODE, let $\frac{d^2x}{dt^2}$ be the acceleration of x , $\frac{dx}{dt}$ be the velocity of x and $x(t)$ be the displacement of x , then solve for $\frac{d^2x}{dt^2}$, then we have:

$$\frac{dx}{dt} = v(t) \dots (2.6)$$

$$\frac{d^2x}{dt^2} = \frac{-2 \frac{dx}{dt} - 0.7x - 0.8x^3 + 0.5 \cos(0.5t)}{1.45} \dots (2.7)$$

Since $\frac{d^2x}{dt^2} = \frac{dv}{dt}$, thus by substituting this into equation 2.7, we will have a pair of first order ODE as below:

$$\frac{dx}{dt} = v(t) \dots (2.6)$$

$$\frac{dv}{dt} = \frac{-2 \frac{dx}{dt} - 0.7x - 0.8x^3 + 0.5 \cos(0.5t)}{1.45} \dots (2.8)$$

Since there is only first order Euler method available in MATLAB ode solvers, therefore in order to implement the nonlinear liver model by using second order Euler

Method, the equation of motion is discretized analytically as below:

Let $\ddot{x} = f(t, x, \dot{x}); x(t_0) = x_0 = 0; \dot{x}(t_0) = u_0 = 0$. The first order ODE would be:

$$\begin{aligned} \dot{x} &= u, \\ \dot{u} &= f(t, x, u), \end{aligned}$$

with initial conditions of $x(t_0) = x_0 = 0$ and $\dot{x}(t_0) = u_0 = 0$, therefore the discretized form would as shown below:

$$\begin{aligned} t_{i+1} &= t_i + \Delta t, \\ x_{i+1} &= x_i + \Delta t(u_i), \\ u_{i+1} &= u_i + \Delta t * f(t_i, x_i, u_i) \dots (2.9) \end{aligned}$$

Linearization of Second Order Euler Method is crucial as without linearization, the computational time for it would be higher and unstable. Thus, linearization is necessary.

The pseudo code in MATLAB for this method would be:

$$\begin{aligned} f(n) &= f_euler(t(n), x(n), u(n)); \\ t(n+1) &= t_i(t(n), h); \\ x(n+1) &= x_i(x(n), h, u(n)); \\ u(n+1) &= u_i(u(n), h, f(n)); \\ n &= n + 1; \end{aligned}$$

As mentioned earlier, ode45 is actually similar to Runge-Kutta method of order 4 (RK4). Thus it is an explicit integration method which has medium accuracy. In order to fulfil the first objective, we just need to implement the nonlinear liver model into MATLAB by keying-in this function, such that:

$$tode45(n+1) = cputime;$$

Meanwhile, MATLAB ode23s is based on Rosenbrock formula of order 2. It is an implicit integration method which has low accuracy but computationally more efficient than the Euler Method. The coding is as follows:

$$tode23s(n+1) = cputime;$$

MATLAB ode23t is also an implicit integration method that implements the trapezoidal rule. This method is chosen because it can be used to solve the moderately stiff problem. The coding goes as follows:

$$tode23t(n+1) = cputime;$$

Lastly, MATLAB ode113 is an explicit integration method which is very accurate since it is the thirteenth order. Ode113 is also known as the variable order Adams-

Bashforth-Moulton PECE solver. It might be more accurate than ode45. However, due to its characteristics as a multistep solver, where it normally needs the solutions at several preceding time points to compute the current solution, ode113 would be less computationally efficient than the ode45. The coding is as follows:

$$tode113(n+1) = cputime;$$

2.5 Case Studies under Various Conditions

Case Study 1:

A second order viscously linear equation (2.1) will be tested under various conditions. There are three conditions, stated as below:

- Condition 1: varying the value of mass, m
- Condition 2: varying the value of internal stiffness, k
- Condition 3: varying the value of damping, c

Case Study 2:

A second order fully nonlinear equation without forces (2.2) will be tested under various conditions. There are two conditions, stated as below:

- Condition 1: varying the value of internal stiffness and external stiffness
- Condition 2: varying the value of damping, c

Case Study 3:

A second order fully nonlinear equation with force (2.3) will be tested under various conditions. There are two conditions, stated as below:

- Condition 1: varying the value of internal and external stiffness
- Condition 2: varying the value of damping, c

There will be a benchmark ideal liver model based on a set of parameters taken from the literature review of Colombo & Littlewood [20]. The set of parameters are given in Table 1.0. This benchmark liver model will be compared in which there will have force exerted onto the model and the other one without force. Each of the simulation results will be plotted in graph where there will be acceleration, velocity and phase plane plot. The phase plane plot is to check the stability of the model. The range of the human liver is standard and taken from Kerdok [21].

The experiment to compare the ODE solvers is conducted in three dimensional rigid bodies. The procedure to find the fastest CPU runtime against a range of time step ($0.00005 < h < 2$) is performed onto a nonlinear model. The MSM is loaded with five specific ODE solvers with damping constant of -2, internal stiffness of 0.8 and exerted force of 0.5N. The procedure starts with simulation time step of 2, 0.6, 0.2, 0.15, 0.05, 0.01, 0.005, 0.0005 and 0.00005. The goal of this experiment is to find the optimal and fastest numerical integration method in solving nonlinear second order equation.

Table 1. Parameters of the benchmark liver model

Gender	Stiffness, k (kPa)			Damping, c (N/m ²)			Mass (kg)
	Slim	Normal	Fat	Slim	Normal	Fat	
Male	4.6	5.3	6.7	5	6	8	1.4 – 1.5
Female	4.3	5.1	6.6	4.5	6	8	1.2 – 1.4

3. RESULTS & DISCUSSION

3.1 Time Execution

The total CPU runtime used in simulation for the numerical integration methods from minimum to maximum were ode45, ode23s, ode113, 2nd order Euler and ode23t as shown in Figure 1.0 and Table 2.0. Ode45 and ode 113 are the built-in explicit integration methods of MATLAB. Ode23s and ode23t on the other hands are the built-in implicit integration methods of MATLAB. The second order Euler in this experiment is derived in Section 2 as the built-in ode15s of MATLAB is not suitable to solve second order of the nonlinear equation. Through this experiment, in Figure 1.0, ode45 (red line) is shown to be the fastest integration method in solving the nonlinear liver model. Ode45 is also known as the explicit fourth-order Runge-Kutta. The total CPU runtime remain almost constant through the range of time step of $0.0005 < h < 2$. When size of time step decreases to 0.00005, total CPU runtime per second increases drastically. Therefore, ode45 is said to have a medium accuracy as ode45 converges at large time steps ($h > 2$) but become inaccurate as the time step decreases ($h < 0.0005$).

Ode23s is the second fastest among the five methods. It is also known as Rosenbrock method which is used to solve stiff differential equations. The graph of ode23s is quite similar to ode45 and ode113. Within the range of time step $0.0005 < h < 2$, the total CPU runtime remains almost constant and increases as the time step decreases at 0.00005. However, ode23s is less accurate than the ode45 as it takes more computational time.

Ode113 has the average speed among the five. It is known as the Adams-Bashforth-Moulton PECE method which is used to solve non-stiff differential equations. The graph of ode113 is quite similar to ode45 and ode23s. Within the range of time step $0.0005 < h < 2$, the total CPU runtime remains almost constant and increases as the time step decreases at 0.00005. Ode113 is used to solve multistep intensive problems. It is very accurate compares to ode45 and ode23s. However, high computational time will cause the simulation fail to response in real-time.

Second order Euler is the most constant and stable among the five. However, the computational time is higher than the aforementioned methods. Therefore, Second order Euler is not applicable in solving the nonlinear liver model as there is a trade-off between computational efficiency and realistic visualization.

Ode23t is the slowest among the five. It is basically an implementation of the trapezoidal rule. The total CPU runtime remains constant within the time step of

$0.0005 < h < 2$. However, it slightly increases at time step of 0.00005. Although in Figure 1.0, ode23t seems to be stable, the computational time per seconds is too high and not applicable in surgical simulation.

Based on the findings and analyses on the graph, it is obvious that ode45 is the most suitable to be implemented into the nonlinear liver simulation because the size of time step is directly proportional to the number of iterations. As the number of iterations increase, the time steps will increase. Aforementioned, ode45 has the medium accuracy but the total CPU runtime per seconds is the lowest among the five methods. Therefore, ode45 suit the best to overcome the trade-off between computational efficiency and accuracy.

Even though ode23s is the second fastest method among the five, it is not the best for real-time surgical simulation because surgical simulation has to be fast and able to achieve a real-time interactive speed. Ode113 on the other hand is the most accurate method among the five. Unfortunately, a real-time surgical simulation requires for a fast interactive speed, thus, ode113 is not suitable for nonlinear liver simulation.

The second order Euler is the most stable among the five. However, the drawback of this method is that it requires higher computational time than the ode113. Ode23t on the other hand is the slowest among the five methods through the implementation of trapezoidal rule.

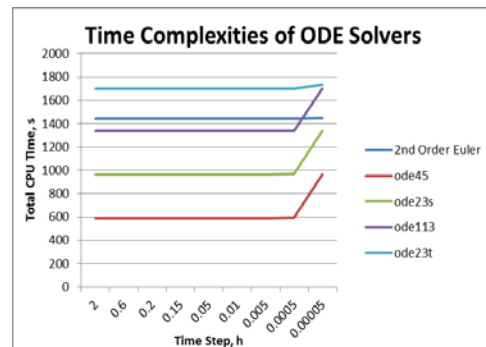


Fig. 1. Computation time versus time step where $h=2, 0.6, 0.2, 0.15, 0.05, 0.01, 0.005, 0.0005, \text{ and } 0.00005$

Therefore, ode45 is the most suitable method to be implemented in the next finding of this study because the total CPU runtime per second is low and ode45 allows for large time steps. In each case, there is a trade-off of accuracy versus speed. Choosing a computationally intensive solver, increasing the number of nonlinear iterations, or reducing the step size increases accuracy and reduces idle time, raising the risk that the simulation will

not run in real time. On the other hand, adjusting these settings in the opposite direction increases the amount of idle time but reduces accuracy. The challenge is to find

settings that provide accurate results while permitting real-time simulation.

Table 2. The execution time for each numerical integration method with varied time steps

Time Step, h	Execution time for each of the Numerical Integration Method				
	2 nd order Euler	ode45	ode23s	ode113	ode23t
2	1444.3	588.155	963.6	1335.6	1700
0.6	1444.3	588.1706	963.6	1335.6	1700.1
0.2	1444.3	588.1706	963.6	1335.6	1700.2
0.15	1444.3	588.1862	963.6	1335.7	1700.2
0.05	1444.3	588.2174	963.6	1335.7	1700.3
0.01	1444.3	588.311	963.8	1335.8	1700.5
0.005	1444.5	588.5294	964	1336	1700.7
0.0005	1444.9	591.8366	967.3	1339.4	1702.7
0.00005	1448.3	963.587	1335.6	1699.9	1738.1

3.2 Investigation of MSM Performance

The second purpose of this experiment is to investigate the performance of a MSM under various conditions. In modelling the human liver, we assume that the MSM is free of damping. However, in actuality, all deformation is damped to some degree of friction force especially for a liver where there exists the internal friction between the molecules. The investigations of this experiment are done as follows:

- **Case Study 1:**

A second order linear equation (3.1), tested by varying the value of mass, damping and internal stiffness of the springs.

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + k_1x = 0 \dots (3.1)$$

Where m is the mass, c is the damping coefficient and k as the stiffness of the spring.

- **Case Study 2:**

A second order nonlinear equation without force (3.2) where the damping and stiffness coefficients are also nonlinear. Here, the tested only vary the stiffness and damping coefficient where mass is fixed at 1.45kg, but there are two different stiffness coefficients involved, one is the internal and the other one is the external stiffness.

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + k_1x + k_3x^3 = 0 \dots (3.2)$$

- **Case Study 3:**

A second order nonlinear equation (3.3) which is similar with (3.2), with a friction force on the RHS of the equation.

$$m \frac{d^2x}{dt^2} + c \frac{dx}{dt} + k_1x + k_3x^3 = P \cos(wt) \dots (3.3)$$

$$f(t) = P \cos(wt)$$

- **Benchmark Ideal Model of A Liver:**

A second order nonlinear equation which is same as (3.3) but with an ideal set of parameter which is abstracted from Colombo [20]. The parameters are illustrated in Table 1.0.

These case studies are then input into and solved by MATLAB R2009a.

3.3 Simulation Results

A. Case Study 1

Figure 2.0 to Figure 4.0 show the simulation results of case study 1. From these graphs, it can be shown that for any linear MSM, no matter how large or small the value of the parameters, the system will always oscillate to a stabilized state. When the mass of the body is large, time taken to stop the oscillation is prolonged. Meanwhile, as the stiffness increases, the amplitude of the system decreases. Conversely, the larger the damping coefficient, the system will stop oscillating faster. As shown in Figure 4.0, the lowest damping coefficient (c=0.1) oscillated the most. The body seems to displace to-and-fro position and never reach a resting point.

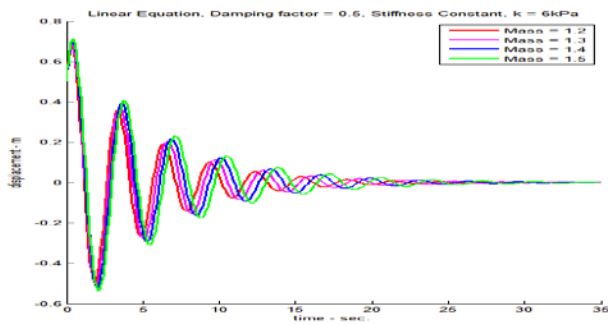


Fig. 2. The graph of displacement versus time when the value of the mass is varied at $m=1.2, 1.3, 1.4$ and 1.5

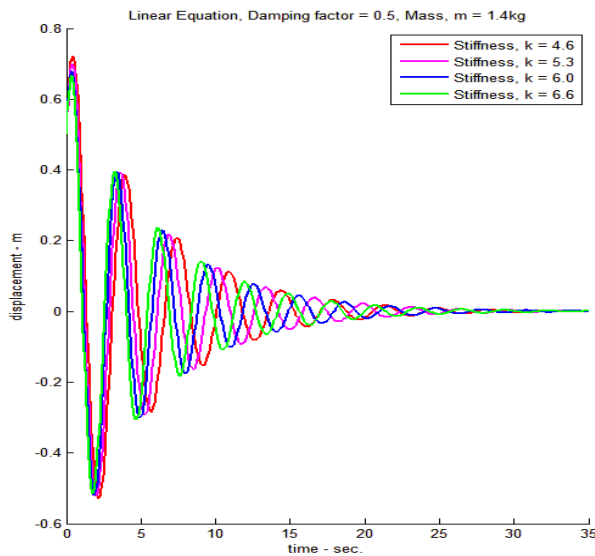


Fig. 3. The graph of displacement versus time when the value of stiffness is varied at $k= 4.6, 5.3, 6.0$ and 6.6

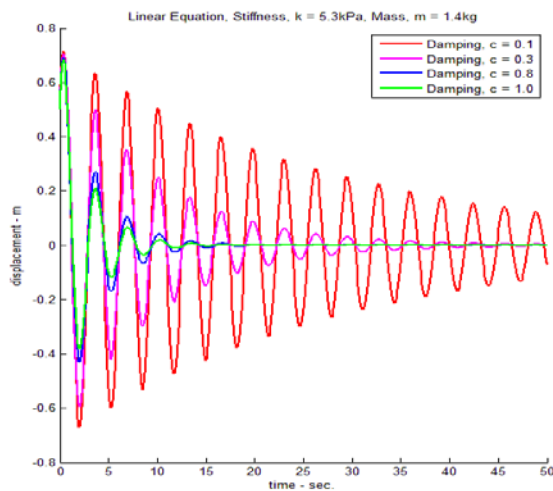


Fig. 4. The graph of displacement versus time when the value of damping is varied at $c= 0.1, 0.3, 0.8, 1.0$

B. Case Study 2

From Figure 5.0, even though the system is either critically damped (red) or overdamped (blue), the system is still stable because the direction field shown that the phase plot for both of the situation is converging to zero in a clockwise direction turning into the equilibrium point at 0.

Meanwhile, Figure 6(i) shows that when $c=-0.2$, the direction field is diverged from zero. Hence, the damping coefficient is less than zero ($c<0$), the system is instable. When $c=0$, as shown in Figure 6(ii) above, the system is unconditionally stable, however it is not pointing towards zero as there is no damping to stop the oscillation. In Figure 6(iii) where $c=2$, the system is stable as the direction field shows that it is converging to zero. However, in Figure 6(iv) where $c=10$, the system is conditionally stable as indicated by direction field where direction is converging onto the curve lying on the x-axis where x and y are equal to 0.

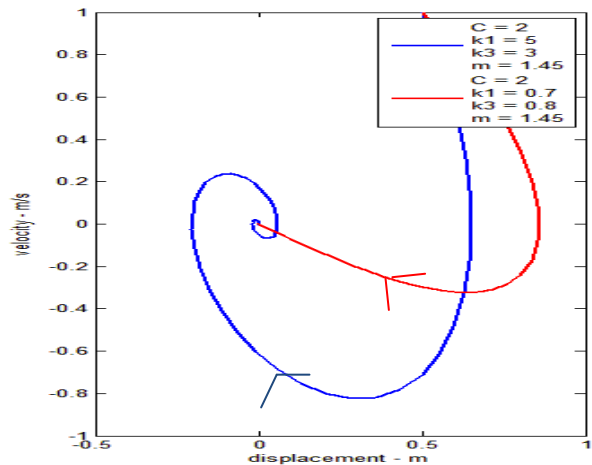


Fig. 5. The phase plane plot of velocity versus displacement which showed the stability when the value of stiffness is varied

C. Case Study 3

Figure 7.0 displays the simulation result of case study 3 where the value of force, P is varied. The damping coefficient is fixed at 2, internal force of 0.7, external force of 0.8, mass of liver weight 1.45kg and shearing angle at 0.5. When the external force of 0.1N (blue) is exerted onto the nonlinear liver, the system undergoes critical damping within 0-5 seconds. However, the body started to oscillate periodically after the critical damping. When the external force of 1.0N (red) is exerted onto the nonlinear liver, the system will undergoes critical damping and oscillated periodically again. As the exerted external force is large, the amplitude of the velocity and acceleration increases. Figure 8.0 indicates that forced nonlinear MSM is converging but it is conditionally stable.

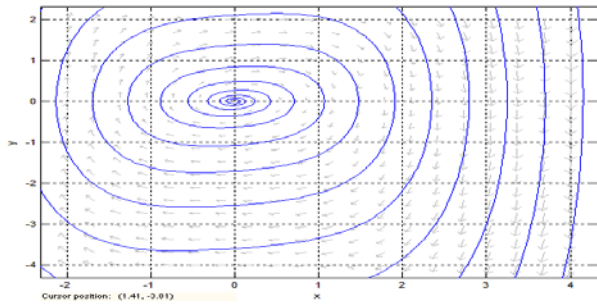


Fig. 6.(i) Phase plane plot when damping, $c=0.2$

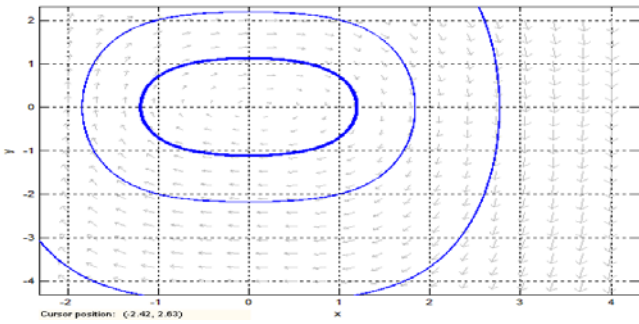


Fig. 6.(ii) Phase plane plot when damping, $c=0$

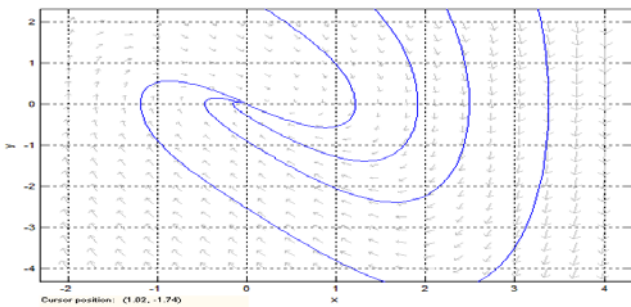


Fig. 6.(iii): Phase plane plot when damping, $c=2$

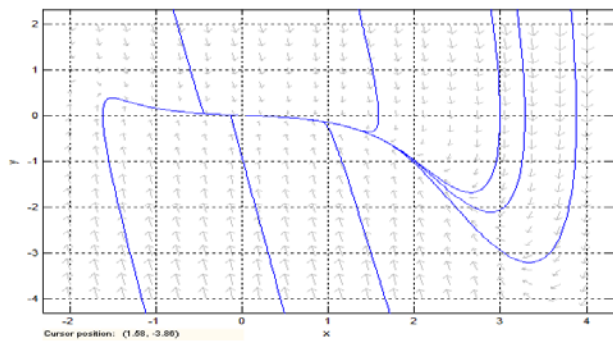


Fig. 6.(iv) Phase plane plot when damping, $c=10$

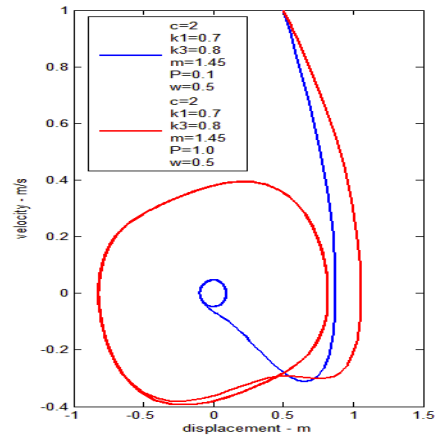


Fig. 7. Phase plane plot which show the stability of case study 3 when the friction force is varied

D. Benchmark Ideal Nonlinear Liver MSM

The simulation result of the benchmark liver model which is based on the parameters given in Table 1 is displayed in Figure 8.0. As shown in Figure 8.0, regardless on the existence of the forces, each of the simulation result of the system undergoes critical damping within 0-1 seconds. However, the unforced nonlinear liver model came to a rest state relatively fast. The forced nonlinear liver model on the other hand keep oscillate periodically. In Figure 8.0, the acceleration undergoes overdamped where unforced nonlinear liver model is more overdamped than the forced nonlinear liver model.

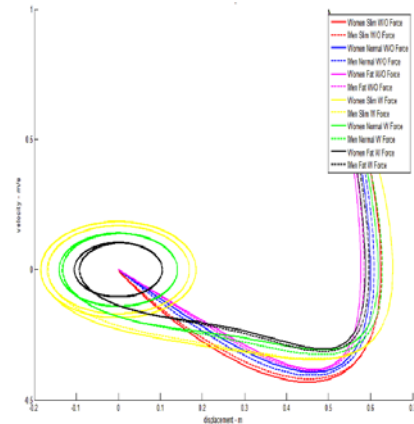


Fig. 8. Phase plane plot of the benchmark model

3.4 Simulation Result Implementation into SOFA

Meanwhile Figure 9.0 shows the results of visualization of the observed parameters such as damping and stiffness coefficients into SOFA (Simulation Open Framework Architecture). The visualization was performed on liver deformation.

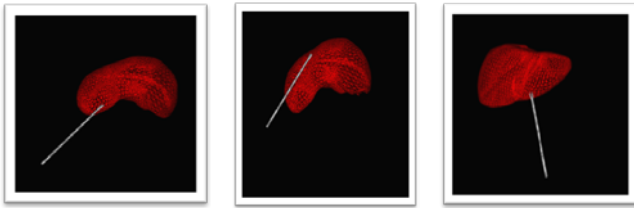


Fig. 9. The visualization of liver deformation implemented in SOFA

4. CONCLUSION

Simulating a simulation that is fast responsive with rapid interactive speed is easy. We can tune any parameters until we have the desired visualization. However, surgical simulation is not computer animation. It does not allow any mistake in the virtual surgical. Surgical simulation has to be concise and there exist a compromise of accuracy and efficiency. Therefore, an adequate of accurate biomechanical information is important in developing an efficient computation strategy. A lack of understanding of the mechanical response of soft tissues to surgical manipulations is not tolerable.

In order to build a real-time surgical simulator, a trade-off between the computational speed and biomechanical accuracy has to be considered. The complexity of the soft tissue models is the limiting the computational resources. Calculation of internal forces, and inverting the mass and damping matrices, consume a large amount of memory. Thus a suitable numerical integration method is crucial so that the computational time will be shortened.

There is still a wide improvement needed in the development of surgical simulation, especially in liver surgical simulation since liver is one of the most important organs in human body. Therefore, it is important to explore more on the stiffness of the liver. There are many situations that caused the liver to become stiff. One of the reasons is cirrhosis disease. This disease is one of the world killer diseases where the dead cells will eat the living cells and thus causing the liver to become stiff. Future works on surgical simulation can be focused on improving the realism of the liver biomechanical by adapting the exact parameters for benchmarking the differences and creating a simulation with nonlinear viscous-elasticity liver model.

REFERENCES

[1] D. Baraff, and A. Witkin, Large Steps in Cloth Simulation, Siggraph '98: Proceedings of the 25th Annual conference on computer graphics and interactive techniques, 1998, pp. 43-54
 [2] G. Bianchi, B. Solenthaler, G. Szekeley, and M. Harders, Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2 (2004) 293-301.

[3] S. Céline, N. Luc, B. Thierry, C. Hélène, J. Marie-Pierre, D. Marlène, S. Göran, Z. Michel, and B. Catherina, *Plant Physiol.* 140 (2008) 349-364.
 [4] C. M. Christensen, *The Innovators Dilemma: When New Technologies Cause Great Firms to Fail*, Harvard Business School Press, Boston, Massachusetts, 1997.
 [5] S. A. Cover, N. F. Ezquerro, J. F. O'Brien, R. Rowe, T. Gadacz, and E. Palm, *IEEE Computer Graphics and Application*, 13 (1993) 68-75.
 [6] B. Eberhardt, O. Eitzmuss, and M. Hauth, *Implicit-Explicit Schemes for Fast Animation with Particle Systems*, Proceedings of the Eurographics workshop on Computer animation and Simulation, Springer-Verlag, (2000), 137-151.
 [7] M. Hauth, J. Gross, W. Strasser, and G. F. Buess, *Soft Tissue Simulation Based Measured Data*. R.E. Ellis and T.M. Peters (Eds.): MICCAI (2003), LNCS 2878, 262-270.
 [8] H. Liang, and M. Y. Shi, *Mixed Numerical Integral Algorithm for Deformation Simulation of Soft Tissues*, F. L. Wang et al. (Eds.): AICI (2010), Part II, LNAI 6320, 87-97.
 [9] Y. Lee, D. Terzopoulos, and K. Waters, *Realistic Modeling for Facial Animation*, 1995, 55-62.
 [10] A. Maciel, G. Sankaranarayanan, T. Halic, V. S. Arikatla, Z. Lu, and S. De, *International Journal of Computer Assisted Radiology and Surgery*, 6 (2010) 457-471.
 [11] A. Nealen, M. Muller, R. Keiser, E. Boxerman, and M. Carlson, *Computer Graphics Forum*, 25 (2005) 809-836.
 [12] C. Paloc, A. Farasi, and F. Bello, *IEEE Computer Graphics & Application*, 26(6) (2006) 24-34.
 [13] P. Phannurat, W. Tharanon, and C. Sinthanayothin, *Simulation of Elastic Soft Tissue Deformation in Orthodontics by Mass Spring System*. 3rd International Symposium on Biomedical Engineering (ISBME), 2008.
 [14] X. Provot, *Deformation Constraints in a Mass Spring Model to Describe Rigid Cloth Behavior*, Proceedings of Graphics Interface 95 (Toronto, 1995), Canadian Human-Computer Communication Society, (1987), 147-154.
 [15] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, *Elastically Deformable Models*, SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer graphics and Interactive Techniques, (1987), 205-214.
 [16] M. Teschner, S. Girod, and B. Girod, *Realistic Modeling of Elasto-Mechanical Properties of Soft Tissue and Its Evaluation*, Proc. CARS, (2001), 51-56.
 [17] P. Volino, and N. Magnenat-Thalmann, *Comparing Efficiency of Integration Methods for Cloth Simulation*, Computer Graphics International Conference, (2001), 0265.
 [18] S. Zhang, L. Gu, P. Huang, and J. Xu, *Real-Time Simulation of Deformable Soft Tissue Based on Mass-Spring and Medial Representation*, Y. Liu, T. Liang, and C. Zhang (Eds.): CVIBA, (2005), LNCS 3765, 419-426.
 [19] L. Zhu, X. Ye, J. Xi, Y. Gu, and S. Guo, *A Real-Time Deformation Modeling Scheme of Soft Tissue for Virtual Surgical*. Proceeding of the 2010 IEEE International Conference on Information and Automation, (2010), 771-775.
 [20] C. Colombo, and J. Littlewood, *J Cyst Fibros.* 10(2) (2011) 7 -15.
 [21] A. Kerdok, M. Ottensmeyer, R. Howe, *Journal of Biomechanics*, 39(2006) 222-2231.