

3D Soft Body Simulation Using Mass-spring System with Internal Pressure Force and Simplified Implicit Integration

Jaruwan Mesit, Ratan Guha and Shafaq Chaudhry

School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida 32816

Email: {jmesit, guha, shafaq}@eecs.ucf.edu

Abstract— In this paper, we propose a method to simulate soft bodies by using gravitational force, spring and damping forces between surface points, and internal molecular pressure forces. We consider a 3D soft body model composed of mesh points that define the body's surface such that the points are connected by springs and influenced by internal molecular pressure forces. These pressure forces have been modeled on gaseous molecular interactions. Simulation of soft body with internal pressure forces is known to become unstable when high constants are used and is averted using an implicit integration method. We propose an approximation to this implicit integration method that considerably reduces the number of computations in the algorithm. Our results show that the proposed method realistically simulates soft bodies and improves performance of the implicit integration method.

Index Terms— Soft body simulation, implicit integration method, internal pressure force, mass-spring system

I. INTRODUCTION

Soft body models simulate real-world objects that deform as they interact with the environment, e.g., clothes, hair, sand, and liquids [12, 15, 16, 17, 18, 22, 29, 30]. A body object is modeled by a group of points, each of which has properties such as position, velocity, force, etc. During each animation time step, the surface points get rearranged into new positions and their properties change, due to external interaction with the environment, and internal characteristics of the model itself.

Extensive research has been conducted on modeling and simulating soft bodies. Complex physics-based models have been developed for accurate and realistic soft body deformations that involve a high number of computations. These include finite element method (FEM) [2, 6, 7, 11], finite volume method (FVM) [1, 2], and long element method (LEM) [8, 9, 10]. On the other hand, there are models based on mass-spring system [20, 21, 28] that provide approximate deformation behavior, yet, are intuitively simple and execute much faster than accurate models. There are, however, some drawbacks in mass-spring systems [13]. The spring connections between points are not easily derived from measured material properties. Moreover, a mass-spring system with explicit integration cannot handle increased stiffness under large spring constants. Stiffness in a system causes

poor stability and requires the numerical integrator to take small time steps, even when the results of motion occur over much larger time intervals. These problems have been addressed by Desbrun *et al.* [14] and Kang *et al.* [19] who have proposed efficient and stable algorithms for animating a mass-spring system with the latter offering a faster calculation. In general, the implicit methods require more memory resources and high number of computations since they involve unknown values.

In this paper, we propose and present an implicit integration method for animating soft bodies. We model surface of the soft body by using a mass-spring system and simulate internal characteristics of the soft body using pressure forces. To show that our proposed method solves the stiffness problem, we compare it with the explicit integration method. Moreover, we provide a performance comparison between the explicit integration method, an adapted version of the implicit integration method presented in [19], and our proposed method. The results show that motion of soft bodies can be realistically rendered with the proposed implicit method, even with large spring constants.

The paper proceeds as follows: Section II provides some background on soft body simulation, and Section III details our approximated implicit method. In Section IV, we present the explicit and implicit methods with which we compare our proposed simplified implicit method. Section V presents the simulation experiments and comparison results. In Section VI, we cite applications that can be implemented by the proposed method. Finally, we conclude in Section VII, and discuss possible further work.

II. BACKGROUND

This section covers background information about the three components of our proposed architecture: mass-spring system for soft body simulation; internal pressure forces for realistic 3D deformation; and Newton's second law of motion for animation.

A. Mass-spring System to Model a Soft Body

Soft bodies using mass-spring systems can be classified into two different categories: modeling a 3D object as a 2D grid structure e.g. cloth, and modeling a 3D object as a 3D structure e.g. a bouncing ball. In these models, a soft body is represented as a triangular, rectangular, or tetrahedral mesh where each point has its own properties such as mass, velocity, force, and position. The force exerted at each point is cumulative of the forces of its neighbors and is represented by a differential equation which can be evaluated using numerical integration.

For 2D grid modeling of a 3D object, Terzopoulos *et al.* [16] proposed deformable objects with elastic properties which have successfully been used in soft body cloth simulation [14, 19]. Provot [28] used a mass-spring system to present the structure of cloth as a mesh of $m \times n$ mass points which are relocated at each time step. In Provot's method, the internal spring force, $\mathbf{F}_{int}(P_{i,j})$, acting on a particular surface point $P_{i,j}$, is calculated by spring forces that connect the point to its neighbors and is given by:

$$\mathbf{F}_{int}(P_{i,j}) = -\sum_{(k,l) \in R} K_{i,j,k,l} \left[l_{i,j,k,l} - l_{i,j,k,l}^0 \frac{l_{i,j,k,l}}{|l_{i,j,k,l}|} \right],$$

where R is the set of all points (k, l) linked to $P_{i,j}$ by a spring; $l_{i,j,k,l}$ is the vector $\overrightarrow{P_{i,j}P_{k,l}}$; $l_{i,j,k,l}^0$ is the rest-length of the spring that links $P_{i,j}$ and $P_{k,l}$; and $K_{i,j,k,l}$ is the stiffness of the spring $(P_{i,j}, P_{k,l})$. The damping force at point $P_{i,j}$ is given by:

$$\mathbf{F}_{dis}(P_{i,j}) = -C_{dis} \mathbf{v}_{i,j},$$

where $-C_{dis}$ is the damping coefficient, and $\mathbf{v}_{i,j}$ is the velocity of point $P_{i,j}$. To avoid unrealistic deformation of a hanging cloth, he proposed to increase stiffness of the spring. Fuhrmann *et al.* [23] use collision detection to simulate interactive animation of cloth and use their results to simulate virtual garments. Since cloth simulation collapses in explicit integration methods for large spring constants, several researchers propose implicit Euler integration method to change position, velocity, and force of a point in each time step [14, 19, 24, 26].

A lot of work has been done on 3D modeling of 3D structures like simulation of surgery, fluid-based soft body, and bouncing ball. A layer mass-spring system for facial animation was first presented in [29]. In craniofacial surgery simulation [30, 31], the patient's skin, muscle, and bone layers are modeled using mass-spring systems where the deformation of each layer is based on its mechanical spring constant. Padilla *et al.*

[27] present the Transurethral Resection of the Prostate (TURP) method to remove inner prostate tissue using a mass-spring system in 3D structure. The mass-spring system in this model calculates the internal force, \mathbf{g}_i , acting on point i by:

$$\mathbf{g}_i = \sum_{j \in N(i)} \mu_{i,j} \frac{(|x_i - x_j| - l_{i,j}^0)(x_i - x_j)}{|x_i - x_j|},$$

where $\mu_{i,j}$ is the stiffness coefficient of the spring connecting point i to any point j in the neighborhood $N(i)$ of i ; x_i is the current position of point i ; x_j is the current position of point j ; and $l_{i,j}^0$ is the spring length at rest position.

Nixon and Lobb [3] present mass-spring system that uses classical fluid mechanic's Navier-stokes differential equations to simulate soft body surfaces. The force interacting with the surface at one particular point is generated from the fluid force exerted by neighboring points. The spring forces, \mathbf{F}_{ij} , between two points, i and j , at positions, s_i and s_j , with velocities, \mathbf{v}_i and \mathbf{v}_j , is given by:

$$\mathbf{F}_{ij} = \left[k_s (|s_j - s_i| - r_{ij}) + k_d \frac{(\mathbf{v}_j - \mathbf{v}_i) \cdot (s_j - s_i)}{|s_j - s_i|} \right] \frac{s_j - s_i}{|s_j - s_i|},$$

where k_s and k_d are the spring and damping constants respectively, and r_{ij} is the spring's rest-length. The force \mathbf{F}_{ji} is equal and opposite to \mathbf{F}_{ij} , i.e., $\mathbf{F}_{ji} = -\mathbf{F}_{ij}$.

Matyka and Ollila [12] present a pressure model for soft body simulation in which pressure force is applied within a 3D mesh structure by using a mass-spring system. The authors use ideal gas approximation to calculate the pressure force being exerted on surface points. Since pressure based simulation is simpler and faster as compared to the fluid based simulation, we apply it into our proposed implicit integration method to simulate a soft body. This model is discussed next.

B. Internal Pressure Force for 3D Modeling

This section explains the modeling of soft-body behavior with a pressure force. The model in this paper is based on the classic soft body method presented in [12]. The pressure force vector \mathbf{F} , acting on a point on the surface, is given by:

$$\mathbf{F} = P \hat{\mathbf{n}} A, \tag{1}$$

where P is the pressure value, $\hat{\mathbf{n}}$ is the normal vector to the surface on which the point resides, and A is the surface area. Note that, a larger force might generate a smaller pressure if it is distributed over a wider surface area. In contrast, a smaller force can create a larger pressure if the area is smaller.

The value of P can be obtained using ideal gas approximation [5]. In the Clausius-Clapeyron equation, we have:

$$PV = nRT \quad (2)$$

where V is volume of the body, n is the gas mol number, R is the ideal gas constant, and T is the temperature of the body. Then, the pressure within the soft body can be calculated by:

$$P = V^{-1}nRT \quad (3)$$

In our simulation, the temperature is assumed to be constant, while the volume is recomputed at every simulation time step.

C. Newton's Second Law of Motion for Soft Body Animation

To animate the soft body, we consider a mesh of points drawn on its surface and use Newton second law of motion to animate those points. The second law of motion states that the time rate of change of a body's momentum is equal to the vector sum of the external forces acting on it [25]. This equation can be expressed by using the following first order differential equation:

$$\frac{d\mathbf{v}(t)}{dt} = \frac{\mathbf{F}(\mathbf{v}(t), t)}{m}, \quad (4)$$

where \mathbf{v} is the velocity of an object; \mathbf{F} is the cumulative force that applies on the object and is a continuous function of velocity, \mathbf{v} , and time, t ; and m is the mass of the object assumed to be constant in our simulation. By a basic theorem of calculus, integrating (4) over the interval $[t, t+h]$ yields

$$\mathbf{v}(t+h) = \mathbf{v}(t) + \int_t^{t+h} \frac{\mathbf{F}(\mathbf{v}(t), t)}{m} dt, \quad (5)$$

where h is the interval time step.

This equation expresses that the velocity can be computed at time $t+h$ if we know (a) the initial velocity of the object at time t and (b) the net force acting on the object at every instant between t and $t+h$. Since the net force is a function of \mathbf{v} and velocity is not known over the interval of integration, the integral in (5) cannot be exactly calculated. This requires approximating the value of \mathbf{F} across the interval of integration by using either the explicit, implicit, or trapezoid Euler methods. In the next section, we discuss the explicit and implicit Euler methods.

Explicit Euler Integration

Explicit Euler integration is a straight-forward approach for soft body simulation with pressure forces where velocity, point, and force are related by the following set of equations [14, 19]:

$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + \mathbf{F}_i^t \frac{h}{m_i} \quad (6)$$

$$x_i^{t+h} = x_i^t + \mathbf{v}_i^{t+h} h \quad (7)$$

Here, \mathbf{v}_i^t is the velocity of point i at time t , \mathbf{F}_i^t is the force acting on point i at time t , x_i^t is the position of point i at time t , and h is the time interval between simulation steps. The position of point i at time $t+h$ can be easily evaluated by the current values of x_i^t , \mathbf{v}_i^t , and \mathbf{F}_i^t . Equation (6) is the result of evaluating (5) at the lower limit of integration.

In the explicit Euler method, the velocity at time $t+h$ is evaluated from force at time t and stability is achieved only when the time steps are small [35]. In fact, for stability, the step size must be inversely proportional to the square root of the stiffness [34]. Otherwise, the simulation will fail.

Implicit Euler Integration

The advantage of implicit Euler integration over explicit Euler integration method is that it solves the stability problem. A large step can be applied to implicit Euler integration without simulation failure. To find the value of variables in the subsequent time step, implicit Euler integration method replaces \mathbf{F}_i^t by \mathbf{F}_i^{t+h} as follows [14, 19]:

$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + \mathbf{F}_i^{t+h} \frac{h}{m_i} \quad (8)$$

$$x_i^{t+h} = x_i^t + \mathbf{v}_i^{t+h} h \quad (9)$$

This small change has been proven to solve unstable conditions in the explicit Euler method [36, 37]. Equation (8) is the result of evaluating (5) at the upper limit of integration. Since implicit Euler method is more computationally expensive in space and time, we present a simple approximation in our proposed implicit integration method that reduces the amount of computation.

III. PROPOSED METHOD

In our proposed implicit integration method, soft bodies experience external gravitational force, surface spring and damping forces, and internal pressure force, all of which allow realistic deformation for 3D modeling. To animate points on the surface of the soft body, Newton's second law of motion with implicit integration method is employed.

A. Modeling of the Soft Bodies

In this paper, a triangular mesh of connected points models a soft body as shown in Fig. 1 [12]. Each point i

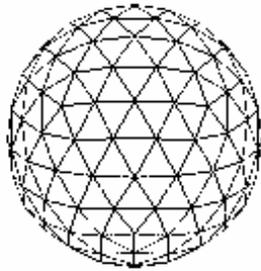


Figure 1. Structure of soft body.

on the surface has six neighboring points where pairs of neighboring points form six triangular faces around point i . Gravitational force, spring and damping forces, and internal pressure forces act on point i and are used to simulate behavior of the soft body. Calculation of these forces is described next.

A1. Gravitational Force

Gravitational force emanates from the earth, moon, or other massively large objects. The force of gravity experienced by an object on earth is always equal to weight of the object experiencing the gravitational pull. Now, gravitational force experienced by a point on the surface of a soft body is computed by:

$$\mathbf{F}_{gi}^t = m_i \mathbf{g}, \quad (10)$$

where \mathbf{F}_{gi}^t is the gravitational force at time t , acting on point i of mass m_i , and \mathbf{g} is acceleration due to Earth's gravity.

A2. Spring and Damping Forces

To simulate elasticity of the soft body, we consider a mass-spring system, in which any two points on the surface of the soft body are connected through springs. A force on one point results in an equal and opposite force on the connected point. A stretched or compressed spring obeys Hooke's law which states that the force exerted by a coiled spring is linearly proportional to the difference between the stretched or compressed length and rest-length of the spring.

Now, to simulate the surface points on a soft body using this mass-spring system, we consider a point i connected to a set of neighboring points. Thus, point i experiences spring forces from all j that belong to i 's connected neighborhood. The cumulative spring force acting on i is given by [19]:

$$\mathbf{F}_{si}^t = \sum_{\forall j|(i,j) \in E} k_{ij} (|x_j - x_i| - l_{ij}^0) \frac{(x_j - x_i)}{|x_j - x_i|}, \quad (11)$$

where \mathbf{F}_{si}^t is the net internal spring force exerted on point i for every point j in the neighborhood E ; k_{ij} is

the spring constant of the spring connecting points i and j ; x_i and x_j are the positions of points i and j respectively; l_{ij}^0 is the rest-length of the spring between points i and j ; and t is current time step. Since the spring applies equal and opposite forces to each connected point pair, it exerts \mathbf{F}_{si}^t on one point and $-\mathbf{F}_{si}^t$ on the connected point.

Along with springs, dampers are typically used in the numerical simulation of soft bodies. A damper connected between two points works against velocity, like viscous drag, to slow down the relative velocity of the two points. For all points j , belonging to the connected neighborhood of point i on the surface of a soft body, the cumulative damping force, \mathbf{F}_{di}^t , experienced by point i is given by [14]:

$$\mathbf{F}_{di}^t = \sum_{\forall j|(i,j) \in E} k_{ij} h(\mathbf{v}_i - \mathbf{v}_j), \quad (12)$$

where \mathbf{F}_{di} is the damping force, k_{ij} is the damping constant of the damper between points i and j , \mathbf{v}_i is the velocity at point i , \mathbf{v}_j is the velocity at point j , t is current time step, and h is time elapsed [14].

A3. Internal Pressure Force

We approximate the internal pressure forces by assuming that there are gas molecules inside the soft body. Since pressure force is the vector sum of the pressure times the area, (1) for point i can be modified to consider all surface areas of its neighboring connected points as follows:

$$\mathbf{F}_{pi}^t = \sum_{\forall jk|(i,j,k) \in E} a_{ijk} \hat{\mathbf{n}} \frac{1}{V} nRT, \quad (13)$$

where \mathbf{F}_{pi}^t is the net pressure force experienced by point i at time t , a_{ijk} is surface area of the face connecting point i to all point pairs (i,k) in the neighborhood of i , $\hat{\mathbf{n}}$ is a normal vector to the surface where the pressure force is acting, V is volume of the body, n is the gas mol number, R is the ideal gas constant, and T is the temperature of the body. We use the Axis Aligned Bounding Box (AABB) [32, 33] to determine volume of the soft body.

B. Combination of All Forces

In this section, we describe how to combine the forces acting at one particular surface point of the soft body. From the previous section, we have all forces, i.e., gravitational force, spring and damping forces and internal pressure forces as follows:

$$\mathbf{F}_{gi}^t = m_i \mathbf{g} \quad (10)$$

$$\mathbf{F}_{si}^t = \sum_{\forall j|(i,j) \in E} k_{ij} (|x_j - x_i| - l_{ij}^0) \frac{(x_j - x_i)}{|x_j - x_i|} \quad (11)$$

$$\mathbf{F}_{di}^t = \sum_{\forall j|(i,j) \in E} k_{ij} h(\mathbf{v}_j^t - \mathbf{v}_i^t) \quad (12)$$

$$\mathbf{F}_{pi}^t = \sum_{\forall j|(i,j) \in E} a_{ij} \hat{\mathbf{n}} \frac{1}{V} nRT \quad (13)$$

Then, the net force \mathbf{F}_i^t at point i is given by:

$$\mathbf{F}_i^t = \mathbf{F}_{si}^t + \mathbf{F}_{di}^t + \mathbf{F}_{pi}^t + \mathbf{F}_{gi}^t \quad (14)$$

In the next section, we present how to animate soft bodies after calculating the net force at any particular point.

C. Animation with Implicit Integration Method

Our proposed integration method is inspired from [14] and [19] to model 3D soft objects as a 2D grid e.g. cloth. Ref. [19] offers a fast and stable calculation method as compared to [14]. We extend this method to model 3D objects as 3D structures and include an approximation for implicit integration to animate the soft bodies. Rather than computing velocity at each neighboring point, we use an approximation of the forces exerted by neighboring points to generate the velocity at the considered point. Our proposed simplified approximated implicit method bears less computation as compared to the method in [19]. The derived method is described below.

As mentioned earlier, implicit Euler method equations are given by:

$$\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + \mathbf{F}_i^{t+h} \frac{h}{m_i} \quad (8)$$

$$\mathbf{x}_i^{t+h} = \mathbf{x}_i^t + \mathbf{v}_i^{t+h} h \quad (9)$$

\mathbf{F}_i^{t+h} is not easily evaluated at the current value of \mathbf{x}_i^t , \mathbf{v}_i^t , and \mathbf{F}_i^t , but can be approximated by a first-order derivative as [14, 19]:

$$\mathbf{F}_i^{t+h} = \mathbf{F}_i^t + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \Delta \mathbf{x}^{t+h} \quad (15)$$

where \mathbf{F}^t is the net force at point i , and is given by $\mathbf{F}^t = [\mathbf{F}_1^t, \mathbf{F}_2^t, \dots, \mathbf{F}_r^t]^T$, where r is number of surface points defining the soft body. The difference of positions, $\Delta \mathbf{x}^{t+h}$, at times t and $t+h$, is given by:

$$\Delta \mathbf{x}^{t+h} = \mathbf{x}^{t+h} - \mathbf{x}^t \quad (16)$$

It is also written as:

$$\Delta \mathbf{x}^{t+h} = (\mathbf{v}^t + \Delta \mathbf{v}^{t+h}) h \quad (17)$$

where $\Delta \mathbf{v}^{t+h}$ is the difference of the velocities at times t and $t+h$.

To calculate \mathbf{F}^{t+h} , Desbran *et al.* [14] use the negated Hessian matrix $H = \frac{\partial \mathbf{F}}{\partial \mathbf{x}}$ of the mass-spring system to solve implicit method integration in (15). Substituting (17) in (15), the equation becomes:

$$\mathbf{F}^{t+h} = \mathbf{F}^t + H(\mathbf{v}^t + \Delta \mathbf{v}^{t+h}) h \quad (18)$$

Then, substituting (18) in (8) and simplifying, we obtain [14]

$$\left(I - \frac{h^2}{m} H\right) \Delta \mathbf{v}^{t+h} = (\mathbf{F}^t + H \mathbf{v}^t h) \frac{h}{m} \quad (19)$$

Now, the approximated Hessian matrix is given as [14]:

$$H_{ij} = k_{ij}, \text{ if } i \neq j; \text{ and}$$

$$H_{ii} = -\sum_{j \neq i} k_{ij}, \text{ otherwise,}$$

where k is the spring constant.

The calculation in (19) becomes expensive as there is an $n \times n$ matrix H involved, where n denotes the number of surface points on the soft body. Since H_{ij} is 0 if points i and j are not linked, the velocity change of point i can be updated by considering only the linked points. This approximation leads to [19]:

$$\left(1 - \frac{h^2 H_{ii}}{m_i}\right) \Delta \mathbf{v}_i - \frac{h^2}{m_i} \sum_{\forall j|(i,j) \in E} (H_{ij} \Delta \mathbf{v}_j) = \frac{\mathbf{F}_i^t h}{m_i} \quad (20)$$

If k is constant for all springs, then H_{ii} and H_{ij} can be approximated as $-kn_i$ and k , respectively, to get [19]:

$$\frac{(m_i + h^2 kn_i)}{m_i} \Delta \mathbf{v}_i^{t+h} = \frac{\mathbf{F}_i^t h}{m_i} + \frac{h^2 k \sum_{(i,j) \in E} \Delta \mathbf{v}_j^{t+h}}{m_i}$$

Simplifying for $\Delta \mathbf{v}_i^{t+h}$ and substituting the value

$$\Delta \mathbf{v}_j^{t+h} = |\Delta \mathbf{v}_j^t| \frac{\mathbf{F}_j^t}{|\mathbf{F}_j^t|} \text{ in (20), we get:}$$

$$\Delta \mathbf{v}_i^{t+h} = \frac{(\mathbf{F}_i^t h + kh^2 \sum_{(i,j) \in E} |\Delta \mathbf{v}_j^t| \hat{\mathbf{F}}_j^t)}{m_i + kh^2 n_i} \quad (21)$$

where $\hat{\mathbf{F}}_j^t$ is the normalized force at point j , evaluated by $\mathbf{F}_j^t / |\mathbf{F}_j^t|$.

We simplify this equation by considering the velocity difference at times t and $t+h$. In our approximation, the value of

$$\sum_{(i,j) \in E} |\Delta \mathbf{v}_j^t| \hat{\mathbf{F}}_j^t \quad (22)$$

is approximated by:

$$n_i |\Delta \mathbf{v}_i^t| \hat{\mathbf{F}}_i^t, \quad (23)$$

where n_i is the number of neighbors of point i and $\hat{\mathbf{F}}_i^t$ is the normalized force at point i , evaluated by $\mathbf{F}_i^t / |\mathbf{F}_i^t|$. Equation 22 computes the summation from the number of neighboring surface points that are connected to point i , all of which are assumed to be of identical mass and form part of a symmetrical mesh that defines the surface of the soft body. Since this is a mass-spring system, force computed at point j is equal and opposite to the force exerted at the connected point i . Thus, we have $\mathbf{F}_j^t = -\mathbf{F}_i^t$. Now, force is used to compute velocity, as shown in (8). Since we assume the same parameters, i.e. mass, spring and damping constants, for each surface point and each spring, we may assume $|\Delta \mathbf{v}_j^t| = -|\Delta \mathbf{v}_i^t|$ without loss of generality. Then, the summation of (22) is approximated by using the number of neighboring points n_i multiplied by the magnitude of velocity difference and normalized force at point i . Using the approximation of (23) in (21), we get:

$$\Delta \mathbf{v}_i^{t+h} = \left(\frac{\mathbf{F}_i^t h + kh^2 (n_i |\Delta \mathbf{v}_i^t| \hat{\mathbf{F}}_i^t)}{m_i + kh^2 n_i} \right).$$

We can simplify the equation as follows:

$$\begin{aligned} \Delta \mathbf{v}_i^{t+h} &= \frac{\mathbf{F}_i^t h}{m_i + kh^2 n_i} + \frac{kh^2 n_i |\Delta \mathbf{v}_i^t| \hat{\mathbf{F}}_i^t}{m_i + kh^2 n_i} \\ &= \frac{\mathbf{F}_i^t h}{m_i + kh^2 n_i} + \frac{kh^2 n_i |\Delta \mathbf{v}_i^t| |\mathbf{F}_i^t|}{m_i |\mathbf{F}_i^t| + |\mathbf{F}_i^t| kh^2 n_i} \\ &= \frac{\mathbf{F}_i^t h}{m_i + kh^2 n_i} + \frac{kh^2 n_i |\Delta \mathbf{v}_i^t| |\mathbf{F}_i^t|}{|\mathbf{F}_i^t| (m_i + kh^2 n_i)} \\ &= \frac{|\mathbf{F}_i^t| |\mathbf{F}_i^t| h + kh^2 n_i |\Delta \mathbf{v}_i^t| |\mathbf{F}_i^t|}{|\mathbf{F}_i^t| (m_i + kh^2 n_i)} \\ &= \frac{\mathbf{F}_i^t h (|\mathbf{F}_i^t| + kh n_i |\Delta \mathbf{v}_i^t|)}{|\mathbf{F}_i^t| (m_i + kh^2 n_i)} \end{aligned} \quad (24)$$

A summary of simulation steps is shown in Fig. 2. The proposed method takes into account gravitational, spring, damping, and internal pressure forces acting at each point

on the surface of the body. All external and internal forces are calculated and aggregated and then the velocity of point i at time $t+h$ is computed by using either of the explicit, implicit, or proposed implicit methods. The result is passed onto the next step and then a new position is calculated for point i . Finally, a display function renders the soft body. This loop repeats until the user terminates the simulation.

IV. METHODS USED FOR COMPARISON

To show that our proposed implicit scheme is stable, we will compare with an explicit integration method. Also, to verify the working of our proposed simplified implicit integration scheme under large spring constants and changing internal pressure forces, we will compare with the implicit Euler integration method of [19]. The simulation will be run under different values of the design parameters: spring constant and gas mole number. Performance of the three methods in terms of frames per second will be analyzed. Also, we will present results of the proposed scheme when multiple soft bodies collide.

Among the three methods, calculation of $\Delta \mathbf{v}_i^{t+h}$ varies. For explicit integration, $\Delta \mathbf{v}_i^{t+h}$ is given by:

$$\Delta \mathbf{v}_i^{t+h} = \mathbf{F}_i^t \frac{h}{m_i}. \quad (25)$$

If we use the implicit integration by considering the surrounding points, $\Delta \mathbf{v}_i^{t+h}$ is given by [19]:

```

do
  for all points  $i=1$  to  $n$ 
    compute  $\mathbf{F}_{gi}^t = m_i \mathbf{g}$ 
    compute  $\mathbf{F}_{si}^t$ 
     $= \sum_{\forall j|(i,j) \in E} k_{ij} (|x_j - x_i| - l_{ij}^0) \frac{(x_j - x_i)}{|x_j - x_i|}$ 
    compute  $\mathbf{F}_{di}^t = \sum_{\forall j|(i,j) \in E} k_{ij} h (\mathbf{v}_j^t - \mathbf{v}_i^t)$ 
    compute  $\mathbf{F}_{pi}^t = \sum_{\forall j|(i,j) \in E} a_{ij} \hat{\mathbf{n}} \frac{1}{V} nRT$ 
    calculate  $\mathbf{F}_i^t = \mathbf{F}_{gi}^t + \mathbf{F}_{si}^t + \mathbf{F}_{di}^t + \mathbf{F}_{pi}^t$ 
    calculate  $\Delta \mathbf{v}_i^{t+h}$  by explicit, implicit, or proposed
    method
    calculate  $\mathbf{v}_i^{t+h} = \mathbf{v}_i^t + \Delta \mathbf{v}_i^{t+h}$ 
    calculate  $\mathbf{x}_i^{t+h} = \mathbf{x}_i^t + \mathbf{v}_i^{t+h} h$ 
  end for all
  display
while user does not end simulation

```

Figure 2. Steps of simulation

$$\Delta \mathbf{v}_i^{t+h} = \frac{(\mathbf{F}_i^t h + kh^2 \sum_{(i,j) \in E} |\Delta \mathbf{v}_j^t| \hat{\mathbf{F}}_j^t)}{m_i + kh^2 n_i} \quad (21)$$

In our proposed implicit integration method by simplified approximation, $\Delta \mathbf{v}_i^{t+h}$ is given by:

$$\Delta \mathbf{v}_i^{t+h} = \frac{\mathbf{F}_i^t h (|\mathbf{F}_i^t| + kh n_i |\Delta \mathbf{v}_i^t|)}{|\mathbf{F}_i^t| (m_i + kh^2 n_i)}. \quad (24)$$

Equations (25), (21) and (24) have been used to compare the three methods in different experiments that have been detailed in the next section.

V. EXPERIMENTS

Experiments were conducted for animation and collision detection of soft bodies with 12K faces on a Pentium4 processor running at 3.2 GHz with 512MB NVIDIA GeForce Go 6800 GPU. The experiments provide insight into the performance of explicit, implicit and the proposed simplified implicit method under changing internal pressure forces and large spring constants.

Soft Body Rendering using the Three Methods

Fig. 3 compares the results between explicit integration, implicit integration [19], and the proposed simplified implicit integration method. The three simulation results show slightly different visual results.

Fig. 4 shows the effect of changing the spring constant of the material. The spring constant is initially set to 500; then, increased to 1000. Fig. 4a shows that the soft body simulation using explicit integration collapses and disappears while the implicit integration method in Fig. 4b and our proposed method in Fig. 4c still generate the soft body animation correctly.

For all simulations, damping constant is 2.0 and mass is 0.1 kg.

Changing Internal Pressure Force in Simplified Implicit Method

This set of experiments was conducted to observe the effect of changing internal pressure forces on the soft body animation generated by our proposed simplified implicit integration method.

Fig. 5 shows results of the simulation when the spring constant is set to 1000 and the gas mol number n is switched from 700 to 100 at frame 500. The results show that the soft body simulation generates a realistic deformation when the internal pressure force is changed.

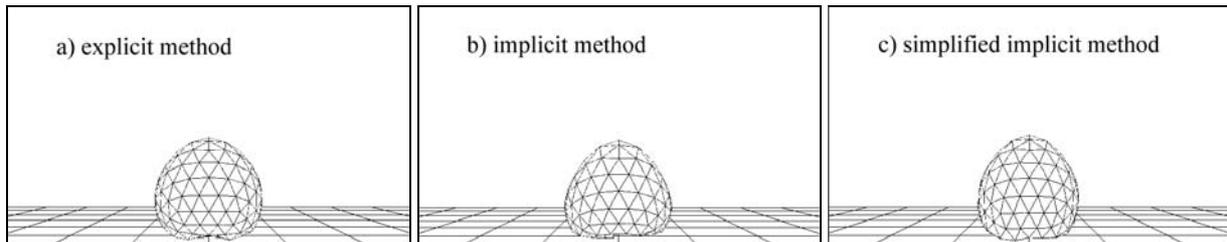


Figure 3. Simulations using spring constant 200, damping constant 2.0 and mass 0.1 kg show different visual rendering for the three methods.

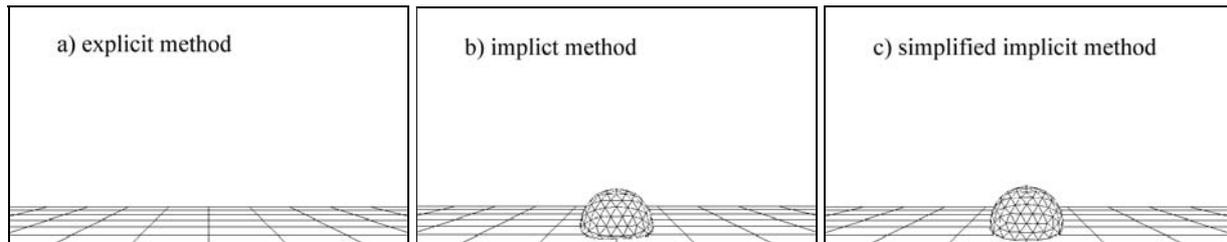


Figure 4. As the spring constant is changed from 500 to 1000, results show that the soft body disintegrates in the explicit integration method while both the implicit methods render the object correctly. Damping constant is set to 2.0 and mass is 0.1 kg.

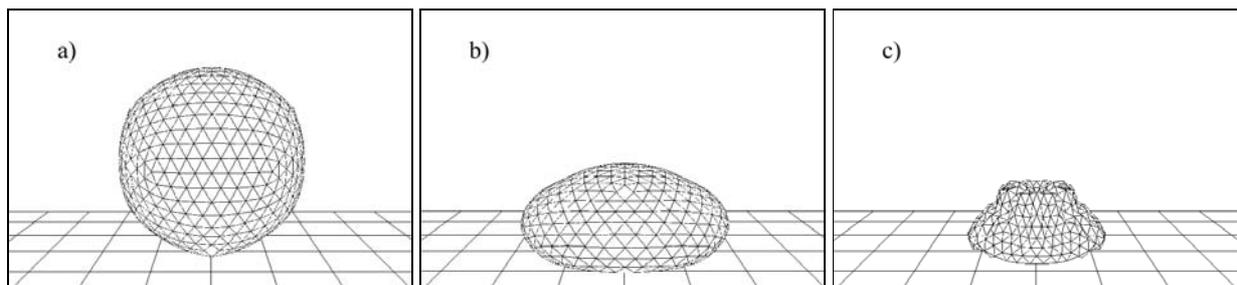


Figure 5. The sequence of soft body animation using the proposed simplified implicit method. The animation shown in captured at frames (a) 100, (b) 500, and (c) 1000 respectively. Gas mol number, n , has been changed from 700 to 100 at frame 500).

Performance Comparison of the Three Methods

Performance comparison between the explicit integration method, the implicit integration method, and our proposed method is shown in Fig. 6. The comparison is measured in terms of average frames per second (FPS). Spring and damping constants are set at 200 and 2.0, respectively.

The results show that both explicit and our proposed methods perform particularly well even when the number of surface points is increased. The explicit integration method gives better FPS as it requires fewer calculations; however, as mentioned earlier, it becomes unstable when the spring and damping constants are large.

On the other hand, the performance of the implicit integration method [19] drops because the algorithm is computationally expensive. Thus, our proposed method gives a stable, yet efficient simulation.

Collision Interaction using Simplified Implicit Method

Fig. 7 shows collision interaction between many soft body objects using the proposed approximated implicit method. The results show that our proposed method can generate realistic simulation of soft bodies' collision even when spring constants are large.

VI. APPLICATIONS

The proposed method is efficient and stable and can implement soft body models and props in real-time effectively. Applications include real-time gaming, non-real time graphics, and medical simulations. These applications have soft bodies like balls, bubbles, blob-like creatures, and skin for facial and speech animation. The method can also be used to animate soft objects for non-real time, pre-rendered movie or television special effects. Another application is surgical simulation where the layers of human tissue, such as the layer of muscles in human face, are simulated using mass-spring systems having different stiffness properties [29, 30]. Similarly, the method can be employed in simulating a human lung which expands and contracts with air based on properties and health of the lungs [38].

VII. CONCLUSION AND FUTURE WORK

We have presented a simplified implicit integration method for soft body 3D simulation using a mass-spring system with internal pressure forces.

Our soft body is a mesh of points connected by springs and affected by internal pressure forces that have been modeled after gaseous molecular interactions. An implicit Euler integration method has been used since the explicit approach makes soft body animation unstable under large spring constants with increased stiffness. Specifically, we have extended the 2D implicit integration method of [19] to 3D rendering of objects and propose a simplified approximation for implicit integration to animate our soft bodies. Instead of computing velocity at each neighboring point, we use an approximation of all forces exerted by neighboring points to generate the velocity at a given point. We present comparison with an explicit method and the implicit method of [19] and show that our proposed method produces stable and efficient 3D simulation of soft bodies.

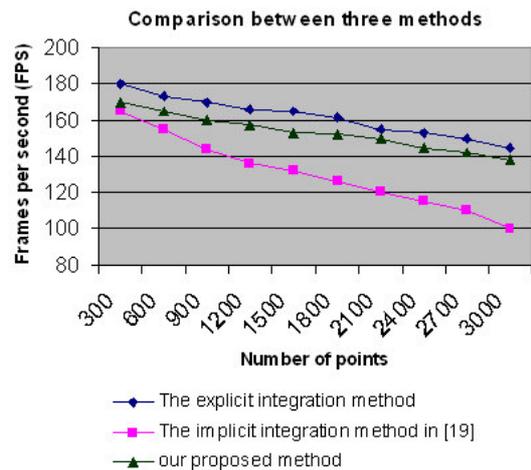


Figure 6. Performance comparison of the three methods: explicit, implicit and proposed implicit.

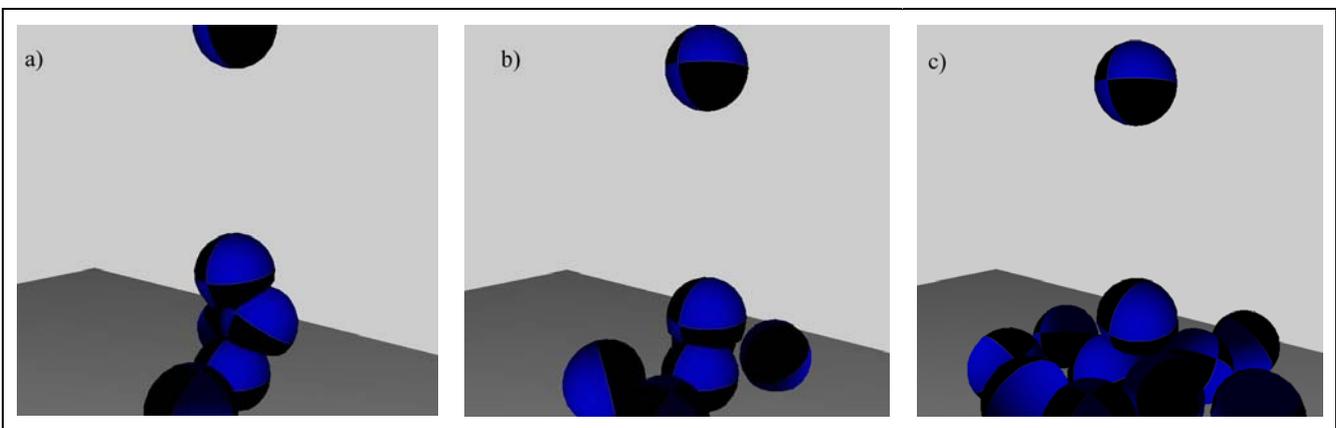


Figure 7. 12K faces of Soft bodies with pressure force are generated; the simulation is captured at frame 500, 1000, and 1500 shown in a), b), and c) respectively.

Future applications include extending the model to simulate soft tissue in medical problems, or to simulate molecular dynamics of different materials. By employing faster collision detection algorithms, the efficiency of the proposed method can be further improved. Additionally, a set of better libraries and bench marks would be designed to compare the algorithm with different models and scenarios.

REFERENCES

- [1] J. Teran, S. Blemker, V. Ng Thow Hing, and R. Fedkiw, "Cloth & deformable bodies: Finite volume methods for the simulation of skeletal muscle," *Euro. Symp. on Comp. Anim. (SIGGRAPH Proc.)*, pp. 68–74, 2003
- [2] G. DeBunney, M. Desbrun, M.-P. Caniy, and A. Barr, "Adaptive simulation of soft bodies in real-time," *Comp. Anim.*, pp. 133–144, May 2000
- [3] D. Nixon, and R. Lobb, "A fluid-based soft-object model," *Comp. Graph. and App.*, IEEE, Vol. 22 Iss. 4, pp. 68–75, July-Aug. 2002
- [4] A. Witkin and D. Baraff, "An Introduction to Physically Based Modeling," *SIGGRAPH Course Notes*, 1993.
- [5] H. B. Callen, *Thermodynamics and an Introduction to Thermostatistics*, 2nd edition, John Wiley & Sons, New York, 1985.
- [6] D. L. James and D. K. Pai, "Multiresolution green's function methods for interactive simulation of large-scale elastostatic objects," *ACM Transactions on Graphics (TOG)*, Vol. 22 Iss. 1, Jan 2003.
- [7] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *Visualization and Computer Graphics, IEEE Transactions on*, Vol. 5 Iss. 1, pp. 62–73, Jan. -Mar 1999.
- [8] R. Balaniuk and K. Salisbury, "Dynamic simulation of deformable objects using the Long Elements Method," *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, pp. 58–65, Mar 2002.
- [9] I. F. Costa and R. Balaniuk, "LEM-an approach for real time physically based soft tissue simulation," *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. 3, pp. 2337–2343, 2001.
- [10] K. Sundaraj, C. Laugier, and I.F.Costa, "An approach to LEM modeling: construction, collision detection and dynamic simulation," *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, Vol. 4, 29 Oct. - 3 Nov. 2001
- [11] R. Rabaetje, "Real-time simulation of deformable objects for assembly simulations," *Proceedings of the Fourth Australian user interface conference on User interfaces*, Vol. 18, 2003
- [12] M. Matyka and M. Ollila, "Pressure Model of Soft Body Simulation," *SIGRAD2003*, Nov 20-21, 2003
- [13] S.F.F Gibson and B. Mirtich, "A Survey of Deformable Modeling in Computer Graphics," *Technical report, Mitsubishi electronic research*, 1997
- [14] M. Desbrun, P. Schröder, and A. Barr, "Interactive animation of structured deformable objects," *Proc. of Graphics Interface '99*, 1999.
- [15] M. Kass, "An introduction to continuum dynamics for computer graphics," In *SIGGRAPH Course Notes*. ACM SIGGRAPH, 1994.
- [16] D. Terzopoulos, J. Platt, and A. Barr, "Elastically deformable models," *Computer Graphics (Proc. of SIGGRAPH '87)*, pp. 205-214, 1987.
- [17] P. Volino, M. Courchesne, and N. Thalmann, "Versatile and efficient techniques for simulating cloth and other deformable objects," *Computer Graphics (Proc. of SIGGRAPH '95)*, pp. 137-144, 1995.
- [18] P. Volino, N. Thalmann, S. Jianhua, and D. Thalmann, "An evolving system for simulating clothes on virtual actors," *IEEE Computer Graphics and Applications*, Vol.16, Iss. 5, pp. 42-51, 1996.
- [19] Y. Kang, J. Choi, and H. Cho, "Fast and stable animation of cloth with an approximated implicit method," *Proc. Computer Graphics International*, 2000.
- [20] W. Recharde DeVaul, "Cloth dynamics simulation" master's thesis, Texas A&M University, Sep 3, 1997.
- [21] D. Pritchard, "Cloth parameter and motion capture" master's thesis, University of Waterloo, 2001.
- [22] P. Volino, F. Cordier and N. Magnenat-Thalmann, "From early virtual garment simulation to interactive fashion design," *Computer-Aid Design*, Vol 37, Iss. 6, pp. 593-608, May 2005.
- [23] A. Fuhrmann, V. Luckas, "Interactive animation of cloth including self collision detection," *Journal of WSCG*, Vol 11, Iss 1, pp. 203-208, 2003.
- [24] D. Baraff A. Witkin, "Large step of cloth simulation," *International Conference on Computer Graphics and Interactive Techniques*, Proceedings of the 25th annual conference on Computer graphics and interactive techniques, pp. 43 – 54, 1998
- [25] D.A. Nixon, "A Fluid-Based Soft Object Model" master's thesis, Univeristy of Auckland, New Zealand, Feb. 1999
- [26] M. Meyer, G. DeBunne, M. Desbrun, and A. H. Barr, "Interactive animation of cloth-like object for virtual reality," *The Journal of Visualization and Computer Animation*, Vol. 12, Iss. 1, pp. 1-12, May 2001.
- [27] M. A. Padilla Castañeda, F. A. Cosio, "Computer simulation of prostate resection for surgery training," *Proceedings of the 25th Annual International Conference of the IEEE*, Vol. 2, pp. 1152-1155, Sep 2003.
- [28] X. Provot, "Deformation Constraints in a Mass Spring Model to Describe Rigid Cloth Behavior," *Proceedings of Graphic Interface'95*, pp. 147-154, 2005.
- [29] K. Waters, "A Physical model of facial tissue and muscle articulation derived from computer tomography data," *Proc. SPIE*, Vol. 1808, pp. 574-583, 1992.
- [30] E. Keeve, S. Girod, B. Girod, "Craniofacial surgery simulation," *Proceedings of the 4th International Conference on Visualization in Biomedical Computing VBC'96*, Hamburg, pp. 541-546, 1996.
- [31] M. Teschner, S. Girod, and B. Girod, "Direct computation of nonlinear soft-tissue deformation," *Proc. Vision, Modeling, Visualization VMV'00*, pp. 383-390, 2000.
- [32] G. Van Den Bergen, "Efficient collision detection of complex deformable models using AABB trees," *Journal of Graphics Tools (USA)*, Vol. 2, pp. 1-13, 1997.
- [33] J. Mesit, E. Hastings, R. Guha, "Muti-level SB Collide: Collision and Self-Collision in Soft Bodies," *CGAMES 2006*.
- [34] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method*. McGraw-Hill, 1991.
- [35] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes in C*, second edition. Cambridge University Press, New York, USA, 1992.

- [36] M. Kass, An introduction to continuum dynamics for computer graphics. In *SIGGRAPH Course Notes*. ACM SIGGRAPH, 1994.
- [37] S. Nakamura, Initial value problems of ordinary differential equations. In *Applied Numerical Methods with Software*, pp. 289-350. Prentice-Hall, 1991.
- [38] A. P. Santhanam, S. N. Pattanaik, J. P. Rolland, and C. Imielinska, “ Physiologically-based Modeling and Visualization of Deformable Lungs”, In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications (PG’03)*, 2003

Jaurwan Mesit is a Ph.D. student at Department of Electrical Engineering and Computer Science at University of Central Florida. She received her M.Sc. in computer science and B.Sc. in computer science from NIDA (National of Development Administration) and Rajabhat Institute Phetchaburi Thailand in 1992 and 1999, respectively. She is working as a graduated research assistant at University of Central Florida. Her research area is computer graphics in collision detection for flexible objects such as cloths, soft body, and water.

Ratan Guha is a professor at University of Central Florida. He received his B.Sc. degree with honors in Mathematics and M.Sc. degree in Applied Mathematics from University of Calcutta and received the Ph.D. degree in Computer Science from the University of Texas at Austin in 1970. He has authored over 125 papers published in various computer journals, book chapters and conference proceedings. His research has been supported by grants from ARO, NSF, STRICOM, PM-TRADE, NASA, and the State of Florida. He has served as a member of the program committee of several conferences, as the general chair of CSMA’98 and CSMA’2000 and as the guest co-editor of a special issue of the Journal of Simulation Practice and Theory. He is a member of ACM, IEEE, and SCS and served as a member of the Board of Directors of SCS from 2004 to 2006. He is currently serving in the editorial board of two journals: International Journal of Internet Technology and Secured Transactions (IJITST) published by Inderscience Enterprises, and Modelling and Simulation in Engineering published by Hindawi Publishing Corporation.

Shafaq Chaudhry is a Ph.D. student of Computer Science at the University of Central Florida. She received her M.Sc. in Computer Science from UCF in Spring 2007 and B.Sc. in Computer Systems Engineering in 2001 from Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Pakistan. Her research interests include wireless ad hoc, sensor and mesh networks, WiMax, tactical networks, wireless communications, performance analysis and simulation.