Indian Institute of Technology Bombay

Bachelor's Thesis

Submitted To:
Salil Kulkarni
Asst. Professor
Mechanical Engineering
Department

Submitted By :
Ashwin Khadke
120100028

Contents

1 Introduction

Enabling robots to handle highly deformable objects for domestic purposes (cooking etc.) requires an accurate description of the object's material properties. Additionally, being able to simulate the behavior of such objects in reasonable time is another requirement. Handling metallic or wooden objects is comparatively easy, as the rigid body assumption isn't quite off. The idea was to come up with a simulation framework that can handle the real-time constraint and simulate objects that undergo large deformations, such as rubber bands or pizza dough.

Smoothed Particle Hydrodynamics[8] and Material Point method[5] are presented. The SPH formulation was implemented and the results are discussed. Additionally, some commonly used models to simulate Hyperelastic materials are presented and discussed.

2 Smoothed Particle Hydrodynamics

- **Model** [4]
  The object is assumed to be made up of particles. At every time instant $'t'$ the state of the object is defined by it's rest ($\mathbf{R}^t = (p_1^T, p_2^T ... p_n^T)^T$) and a deformed ($\mathbf{D}^t = (q_1^T, q_2^T ... q_n^T)^T$) configuration, where $\mathbf{p}_i$ denotes the resting position of the i^{th} particle and $\mathbf{q}_i$ denotes it's deformed position. The configuration of each particle is it's relative position with respect to it's neighbours. Neighbourhood of a particle is denoted by $\mathbf{N}_i$, defined by the set of particles enclosed in a euclidean ball around the point. $\mathbf{P}_i$ be the rest configuration of the i^{th} particle and $\mathbf{Q}_i$ be the deformed configuration

$$\mathbf{P}_i = [(p_1 - p_i), (p_2 - p_i), ..., (p_r - p_i)] \tag{1}$$

$$\mathbf{Q}_i = [(q_1 - q_i), (q_2 - q_i), ..., (q_r - q_i)] \tag{2}$$

$$\mathbf{W}_i = Diag(w_{i,1}, w_{i,2} ... w_{i,r}) \tag{3}$$

$$w_{i,j} = \begin{cases} c(\cos\frac{(r_{i,j}+h)\pi}{2h}) + c & : r_{i,j} \leq h \\ 0 & : r_{i,j} > h \end{cases}$$

$$c = \frac{1}{4h^3(\frac{\pi}{3} - \frac{8}{\pi} + \frac{16}{\pi^2})}$$

Here $\mathbf{r}$ is the number of particles in the neighbourhood of $\mathbf{i}$, $r_{i,j}$ is the distance between the j^{th} neighbour of i and i, and $w_{i,j}$ denotes the weight[7] assigned to

the $i's$ j^{th} neighbour. $\mathbf{F}_i$ denotes the affine transformation that 'best relates' $\mathbf{P}_i$ and $\mathbf{Q}_i$

$$\mathbf{F}_i = \min_F \sum_{j \epsilon N_i} w_{i,j} \|F(p_i - p_j) - (q_i - q_j)\| \tag{4}$$

$$\mathbf{F}_i = \mathbf{Q}_i \mathbf{W}_i \mathbf{P}_i^T (\mathbf{P}_i \mathbf{W}_i \mathbf{P}_i^T)^{-1} \tag{5}$$

$$\mathbf{F}_i = \mathbf{U}_i \widehat{\mathbf{F}}_i \mathbf{V}_i^T$$

Cauchy strain is defined as $\epsilon_i = \frac{(\widehat{F}_i + \widehat{F}_i^T)}{2} - I = \widehat{F}_i - I$. Strain energy density $U_i = \frac{(\epsilon_i \sigma_i)}{2}$, where stress $\sigma_i = C\epsilon_i$. ϵ_i can be written as a 6-element vector. C is a 6x6 material matrix.

$$C = \begin{bmatrix} 2G+\lambda & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & 2G+\lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & 2G+\lambda & 0 & 0 & 0 \\ 0 & 0 & 0 & G & 0 & 0 \\ 0 & 0 & 0 & 0 & G & 0 \\ 0 & 0 & 0 & 0 & 0 & G \end{bmatrix}$$

$$G = \frac{E}{2(1+\nu)}, \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$$

The net internal forces developed at the i^{th} point are given by,

$$f_i = -\sum_{j \epsilon N(i)} \frac{\partial U_j}{\partial \mathbf{q}_i} = -\sum_{j \epsilon N(i)} \frac{\partial \widehat{F}_j}{\partial \mathbf{q}_i} (\widehat{F}_j - I) \tag{6}$$

Assuming U_i and V_i are constants in an iteration. The derivatives can be evaluated as follows

$$\frac{\partial F_j}{\partial q_{i,d}} = \begin{cases} w_{i,j} U_j^T e_d (p_i - p_j)^T (P_j W_j P_j^T)^{-1} V_j & (if i \neq j) \\ -U_i^T e_d \sum_{k \neq i} w_{i,k} (p_k - p_i)^T (P_i W_i P_i^T)^{-1} V_i & (if i = j) \end{cases} \tag{7}$$

Only the symmetric matrix $\frac{1}{2}((\frac{\partial F}{\partial q_{i,d}})^T + (\frac{\partial F}{\partial q_{i,d}}))$ is stored in the form of a 6 element vector. The Laplacian matrix is defined as follows

$$L_{j,i} = \begin{bmatrix} \frac{\partial F_j}{\partial q_{i,1}} & \frac{\partial F_j}{\partial q_{i,2}} & \frac{\partial F_j}{\partial q_{i,3}} \end{bmatrix} \tag{8}$$

$L_{*,i}$ be the i^{th} column of L. The internal forces can thus be written as

$$f_i^t = -(L_{*,i})^T C L D^t + \sum_j (L_{j,i})^T C I \tag{9}$$

- **Plasticity**[3]
  If the stresses developed in the object exceed the yield stress, it undergoes some permanent deformations. The plastic deformations are non-linear with respect to particle positions. It is assumed here that in each time-step the object undergoes only elastic deformations and at the end of the time-step the plastic deformations are separated. A multiplicative plasticity model is assumed.

$$\mathbf{F}_i = \mathbf{F}_{e,i}\mathbf{F}_{p,i} \quad \mathbf{F}_{e,i} = \mathbf{U}_i\widehat{\mathbf{F}}_{e,i}\mathbf{V}_i^T \quad \mathbf{F}_{p,i} = \mathbf{V}_i\widehat{\mathbf{F}}_{p,i}\mathbf{V}_i^T$$

  Thus the following result,

$$\widehat{\mathbf{F}}_i = \widehat{\mathbf{F}}_{e,i}\widehat{\mathbf{F}}_{p,i}$$

  To evaluate $\widehat{\mathbf{F}}_{p,i}$ the following heuristic is applied.

$$\widehat{\mathbf{F}}_i^* = (det(\widehat{\mathbf{F}}_i))^{\frac{-1}{3}}\widehat{\mathbf{F}}_i$$

$$\widehat{\mathbf{F}}_{p,i} = (\widehat{\mathbf{F}}_i^*)^\gamma$$

$$\gamma = \min(\mu(\frac{\|\sigma\| - P_y - K\alpha}{\|\sigma\|}), 1) \ ^1$$

- **Explicit Integration**
  If explicit integration is used, the update rule looks as follows,

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \Delta t(\mathbf{f}_{ext,i}^t + \mathbf{f}_i^t)$$

$$\mathbf{q}_i^{t+1} = \mathbf{q}_i^t + \Delta t(\mathbf{v}_i^t) + \frac{(\Delta t)^2}{2\mathbf{m}_i}(\mathbf{f}_{ext,i}^t + \mathbf{f}_i^t)$$

  Thus $\mathbf{D}^{t+1}$ is obtained. To evaluate the internal forces in the next time step, evaluate deformation matrix $\mathbf{F}$ and it's SVD ($\mathbf{U}$, $\widehat{\mathbf{F}}$ and $\mathbf{V}$) using $\mathbf{P}_i^t$ and $\mathbf{Q}_i^{t+1}$. Evaluate the Laplacian matrix using equations (15) and (16). Internal forces are obtained by applying equation (17)

$$\mathbf{f}^{t+1} = -\mathbf{L}^T\mathbf{CLD}_{t+1} + \sum_j(\mathbf{L}_{j,*})^T\mathbf{CI} \tag{10}$$

  The new rest state can be obtained by pre-multiplying $\mathbf{F}_{e,i}^{-1}$ to $\mathbf{D}^{t+1}$

$$\mathbf{R}^{t+1} = \mathbf{F}_{e,i}^{-1}\mathbf{D}^{t+1}$$

[1] $\gamma = \max(\mu(\frac{\|\sigma\| - P_y - K\alpha}{\|\sigma\|}), 0)$ is another heuristic

- **Implicit Integration**

The update rule for implicit integration is similar to the one in explicit except that the forces of next time-step are used,

$$\mathbf{q}_i^{t+1} = \mathbf{q}_i^t + \Delta t(\mathbf{v}_i^t) + \frac{(\Delta t)^2}{2\mathbf{m}_i}(\mathbf{f}_{ext,i}^{t+1} + \mathbf{f}_i^{t+1})$$

in vector form,

$$\mathbf{D}^{t+1} = \mathbf{D}^t + \Delta t(\mathbf{V}^t) + \frac{(\Delta t)^2}{2}\mathbf{M}^{-1}(\mathbf{f}_{ext}^{t+1} + \mathbf{f}^{t+1}) \tag{11}$$

where $\mathbf{M}$ is the mass matrix, $\mathbf{V} = (\mathbf{v}_1^T, \mathbf{v}_2^T ... \mathbf{v}_n^T)^T$ is the velocity vector. Substituting equation (18) in equation (19),

$$\left(\frac{(\Delta t)^2}{2}\mathbf{M}^{-1}\mathbf{L}^T\mathbf{C}\mathbf{L} + \mathbf{I}\right)\mathbf{D}^{t+1} = \mathbf{D}^t + (\Delta t)\mathbf{V}_t + \frac{(\Delta t)^2}{2}\mathbf{M}^{-1}\left(\mathbf{f}_{ext}^{t+1} + \sum_j \mathbf{L}_{j,*}^T\mathbf{C}\mathbf{I}\right) \tag{12}$$

Starting from the initial state of the input objects, the algorithm for implicit integration performs the following steps

1) Estimate $\mathbf{D}^{t+1}$ using equation (19) and assuming that the internal forces are same as in the previous time-step ($\mathbf{f}^{t+1} = \mathbf{f}^t$).
2) Estimate the rotation matrices $\mathbf{U}$ and $\mathbf{V}$ based on the estimated $\mathbf{D}^{t+1}$
3) Compute $\mathbf{L}$ and solve the linear system of equations in (20) to evaluate $\mathbf{D}^{t+1}$
4) Compute the velocities in the next time-step ($\mathbf{V}^{t+1} = (\mathbf{D}^{t+1} - \mathbf{D}^t)/\Delta t$)
5) Compute the rotation matrices and Laplacian using the computed $\mathbf{D}^{t+1}$ and evaluate internal forces using equation (18)
6) Separate the plastic and elastic deformations evaluate $\mathbf{R}^{t+1} = \mathbf{F}_{e,i}^{-1}\mathbf{D}^{t+1}$

- **Implementation details**

1) The simulation module was implemented on MATLAB. Time-step of 2ms were used in case of implicit and 0.1ms in case of explicit.
2) SVD and sparse system solver functions in MATLAB used for steps (2) and (3) respectively.
3) Approximately uniform point-sets (after refining and downsampling) of sizes 162, 1016 and 2025 were simulated. Instead of picking the k-nearest neighbours, all particles within a certain radius (τ(distance from closest neighbour), $\tau = 4.5$) were considered as neighbours.
4) 'h' factor used in weights was assigned to be 1.01(distance of the farthest

neighbour)

- **Simulations**
  Simulations including plasticity effects fail miserably for point-sets of sizes from 100 to 2000. The internal forces blow up in just the initial few time steps and distort the entire point set. Thus a part-by-part analysis of the simulation was performed. Only the elastic formulation was considered for simulating the transient response of the free end of a bar for some initial extension. The results are presented below.

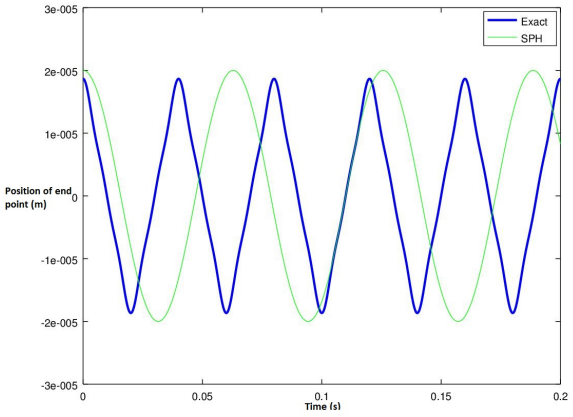

Figure 1: Trajectory of the free end of a bar with an extension of 0.002L

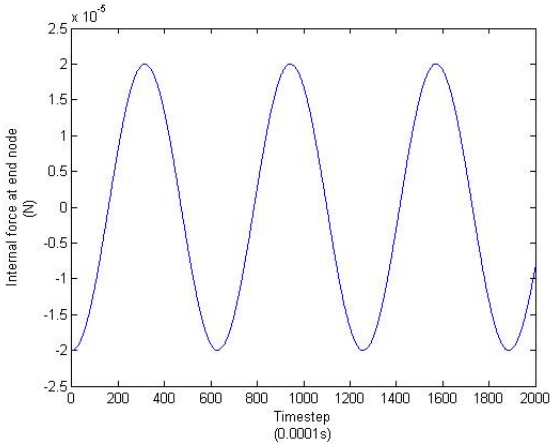

Figure 2: Internal force at a node on the free end of a bar with an extension of 0.002L

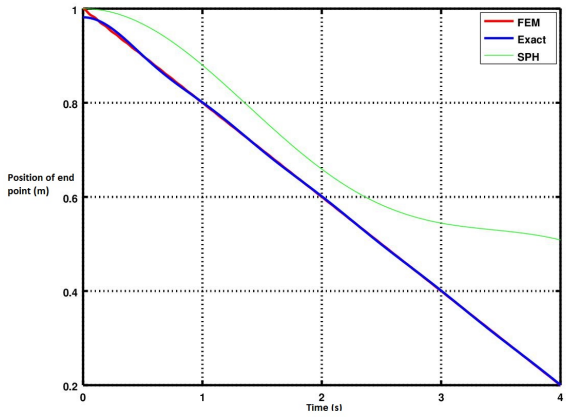

Figure 3: Trajectory of the free end of a bar with an extension of 0.2L

The above results were obtained for a bar 1cm long comprising of 8 nodes in the SPH simulation (8 vertices of a cube). The parameters used for simulation were, $E = 100N/m^2$, $\rho = 100N/m^3$ and $timestep = 0.0001sec$. The trajectories don't match the exact solution. Considering it as a linear elastic system, the net average force developed in the cross-section would be $EA\Delta x/L = 2 * 10^-5N$. But the simulations show the same amount of force on each node, i.e. four times the expected value in the cross-section! Another approach to model a small bar could be using a linear spring fixed at one end and a lumped mass attached to it at the other end. The fundamental frequency thus would be $\sqrt{k/m}$, where k is the equivalent spring stiffness (EA/L) and m is the lumped mass i.e. sum of masses of nodes at the free end. The frequencies do not match! If we consider a finer grid i.e. more number of cross-sections, the system is similar to n-masses attached to each other by springs. The FFT of the position of mass at the terminal end was computed and the principal frequencies compared with the analytical solution for $n = 2$. Even they don't match!

A finer grid of 4000 nodes was simulated. A bar of 5m length was subjected to 20N tensile force ($E = 100N/m^2$, $\rho = 100N/m^3$). It was released after steady state was achieved and the transient response simulated. The exact result is a saw-tooth like trajectory. Only the first 10 modes are considered to generate the profile. The SPH simulated response still does not match the exact solution.
The model is unstable for large deformations. Figure[4] depicts flipping of the tetrahedral elements (composed of a node and it's neighbours) when the

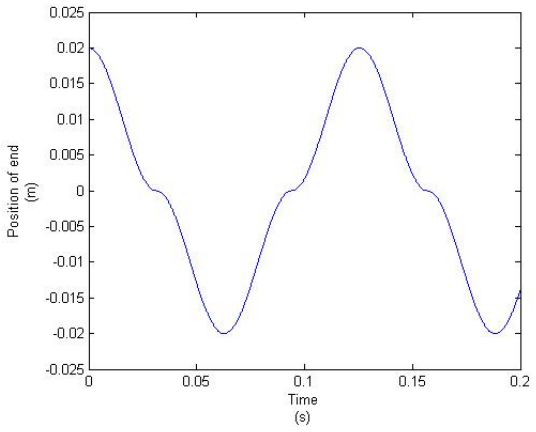

Figure 4: Trajectory of the free end of a bar with an extension of L

deformations are large.

- **Remarks**
  1) The model is only valid for small deformations. The strain approximation $(F + F^T)/2 - I$ instead of $F^T F - I$ is valid for small strains only.
  2) The computation of internal forces using strain energy density (Eq. [6]) lacks a volume measure. Thus to evaluate the internal force on the i^{th} node due to the j^{th} node, a volume measure equivalent to $(m_i + m_j)/2\rho$ was used.
  3) The method computes appropriate deformation gradients (verified analytically for the above simple cases), but fails to compute correct internal forces.

3 Choosing a Constitutive model

A constitutive model describes the relation between generated stresses and observed deformations in a material. The stresses developed in a material are work conjugates to generated strains. Different strain measures thus generate different definitions of stress tensors. The general approach used to arrive at a constitutive relation is to start with an energy function that describes the deformation energy (per unit volume) as a function of strains or invariants of the strain tensor, then relating the derivatives of this function with respect to various strain measures to corresponding stresses. Stability of the model is necessary for successful simulations. Therefore it is important to consider the range of forces or deformations that would be encountered before selecting an energy function. Following are some energy functions commonly used to model hyperelastic materials.

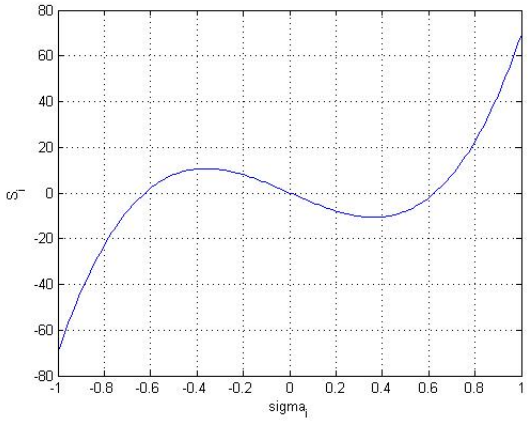

Figure 5: Principal stress as a function of the principal stretch
St. Venant Kirchoff model, $\mu = 100N/m^2$ and $\lambda = 27.78N/m^2$

- **Saint Venant Kirchoff's model**

Ψ represents the energy density function.
σ_i are the principal stretches, i.e. singular values of the deformation gradient

$$\Psi = \frac{\lambda}{2}\Big(\frac{\Sigma_i(\sigma_i^2 - 1)}{2}\Big)^2 + \mu\frac{\Sigma_i(\sigma_i^2 - 1)^2}{4} \tag{13}$$

$$\frac{\partial\psi}{\partial\sigma_i} = \frac{\lambda}{2}(\sigma_i(\Sigma_j\sigma_j^2)) - \frac{3\lambda}{2}\sigma_i + \mu(\sigma_i^3 - \sigma_i) \tag{14}$$

Shown below is the variation of principal stress with the principal stretch (with the other principal stretches remaining constant). The principal stretch remains 1 in the undeformed state. The element is in extension if $sigma_i > 1$ and it is in compression if $sigma_i < 1$. If $sigma_i < 0$ then the tetrahedral element (comprising of the node and it's neighbours) has flipped. As we can see in Figure 5, the principal stress starts decreasing beyond a certain extent of compression. Beyond this, due to absence of any restoring force, the element will get flipped. Thus St. Venant Kirchoff model works only for small strains in compression.

- **Compressible Neo Hookean model**

$$\Psi = \mu\Big(\frac{\Sigma_i(\sigma_i^2 - 1)}{2}\Big) - \mu\log J + \frac{\lambda}{2}(\log(J))^2 \tag{15}$$

Here, $J = |\Pi_i \sigma_i|$. For small deformations the above expression can be approximated as below [6] [2]

$$\Psi = \mu\Big(\frac{\Sigma_i(\sigma_i^2 - 1)}{2}\Big) - \mu \log J + \frac{\lambda}{2}(J-1)^2 \tag{16}$$

The variation is approximately linear for large principal stretches. In the

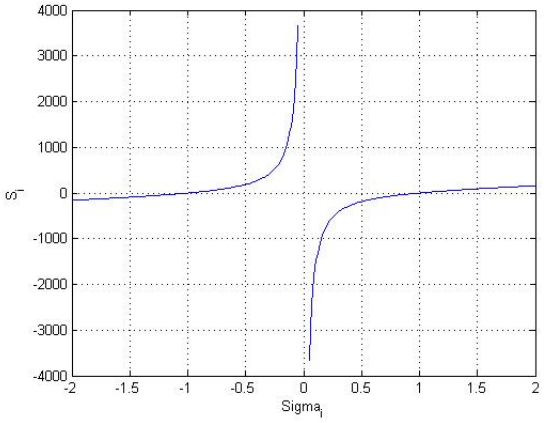

Figure 6: Principal stress as a function of the principal stretch
Compressible Neo-hookean model, $\mu = 100N/m^2$ and $\lambda = 27.78N/m^2$

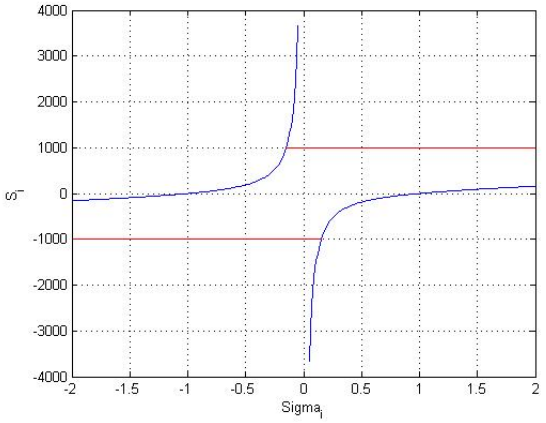

Figure 7: Principal stress as a function of the principal stretch
Bounded Compressible Neo-hookean model, $\mu = 100N/m^2$ and $\lambda = 27.78N/m^2$

compressive region the response stress goes on increasing indefinitely (Figure [6]). Thus it is impossible for elements to flip, but this is not practical as no

material can provide infinite response stress. Thus depending upon the material the stress values are bounded and the modified model can be used to simulate (Figure [7]).
Besides these, the Generalized Neo-Hookean model and the Generalized Mooney-Rivlin model are also used. Both are rubber elasticity models, with the former being used for rubbers with limited compressibility.

4 Material Point Method

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \nabla \rho . v = 0 \tag{17}$$

$$\rho \frac{Dv}{Dt} = \nabla . \sigma + \rho g \tag{18}$$

$$\sigma = \frac{1}{J} \frac{\partial \Psi}{\partial F} F^T \tag{19}$$

The basic idea behind the Material Point Method [5] is to use particles to track mass, momentum and deformation gradient. Each particle holds a position x_p, has some mass m_p and caries a velocity v_p. Momentum conservation and mass conservation equations can be easily discretized, but due to lack of an underlying mesh, computing stress derivatives for force evaluations is difficult. To tackle this problem a background grid is considered. Interpolating functions over the grid are used to discretize the $\nabla.\sigma$ terms. The grid functions used are as follows.

$$N_{\mathbf{i}}^h(\mathbf{x}_p) = N\Big(\frac{1}{h}(x_p - ih)\Big) N\Big(\frac{1}{h}(y_p - jh)\Big) N\Big(\frac{1}{h}(z_p - kh)\Big) \tag{20}$$

where $\mathbf{i} = (i, j, k)$ represents the grid index. $\mathbf{x}_p = (x_p, y_p, z_p)$, h represents the grid spacing and

$$N(x) = \begin{cases} \frac{1}{2}|x|^3 - x^2 + 2/3 & \text{if } 1 > |x| \geq 0 \\ -\frac{1}{6}|x|^3 + x^2 - 2|x| + 4/3 & \text{if } 2 > |x| \geq 1 \\ 0 & otherwise \end{cases} \tag{21}$$

For compact notation we use $w_{\mathbf{i}p} = N_{\mathbf{i}}^h(\mathbf{x}_p)$ and $\nabla w_{\mathbf{i}p} = \nabla N_{\mathbf{i}}^h(\mathbf{x}_p)$. Therefore, to update velocities at the grid nodes, mass and momentum are first transferred from the particles to the grid and the discretized stress derivatives are used to evaluate the new velocities. The transfer process is done using the interpolating weights $w_{\mathbf{i}p}$.

- **Method**

  1) The first step is to transfer mass and velocities from particles to the grid. This is done using the interpolating functions. $m_{\mathbf{i}}^n = \Sigma_p m_p w_{\mathbf{i}p}^n$ and $v_{\mathbf{i}}^n = \Sigma_p v_p^n w_{\mathbf{i}p}/m_{\mathbf{i}}^n$

  2) The force discretization requires a notion of particle volume in the initial configuration. This is estimated as follows

$$\rho_p^0 = \Sigma_i \frac{m_{\mathbf{i}}^0 w_{\mathbf{i}p}^0}{h^3} \tag{22}$$

$$V_p^0 = \frac{m_p}{\rho_p^0} \tag{23}$$

  The internal forces are evaluated as follows

$$f_{\mathbf{i}}(x) = \Sigma_p V_p^n \sigma_p \nabla w_{\mathbf{i}p}^n \tag{24}$$

  Here $V_p^n = J_p^n V_p^0$, where J is the jacobian.

  3) Update velocities on the grid

$$v_{\mathbf{i}}^* = v_{\mathbf{i}}^n + \Delta t m_{\mathbf{i}}^{-1} f_{\mathbf{i}}^n \tag{25}$$

  4) Detect if projected grid nodes collide with any object, if they do update velocities accordingly. If explicit time integration scheme is used $v_{\mathbf{i}}^{n+1} = v_{\mathbf{i}}^*$. Note here that $v_{\mathbf{i}}^{n+1}$ is different from the grid velocity that would be evaluated in the next time step and the two should not be confused.

  5) Update deformation gradient for each particle as $F_p^{n+1} = (I + \Delta t \nabla v_p^{n+1}) F_p^n$ where $\nabla v_p^{n+1} = \Sigma_{\mathbf{i}} v_{\mathbf{i}}^{n+1} (\nabla w_{\mathbf{i}p}^n)^T$

  6) Update particle velocities as follows [1]

$$v_p^{n+1} = (1-\alpha) v_{PICp}^{n+1} + \alpha v_{FLIPp}^{n+1} \tag{26}$$

$$v_{PICp}^{n+1} = \Sigma_{\mathbf{i}} v_{\mathbf{i}}^{n+1} \nabla w_{\mathbf{i}p}^n \tag{27}$$

$$v_{FLIPp}^{n+1} = v_p^n + \Sigma_{\mathbf{i}} (v_{\mathbf{i}}^{n+1} - v_{\mathbf{i}}^n) \nabla w_{\mathbf{i}p}^n \tag{28}$$

  7) Update particle positions, $x_p^{n+1} = x_p^n + \Delta t v_p^{n+1}$

5 Conclusion

The SPH based simulations neither match the FEM nor the analytical results. The MPM, owing to it's combination of grid based and meshless approach can be a good alternative to handle large deformation problems. Thus MPM is being explored, but not only the method but also the choice of constitutive model is important. Thus different constitutive models and their formulations need to be considered, their stability and accuracy in our domain analyzed.

References

[1] The affine particle-in-cell method. Walt Disney Animation Studios, Chenfanfu Jiang, Craig Schroeder, Andrew Selle, Joseph Teran and Alexey Stomakhin.

[2] Energetically consistent invertible elasticity. SIGGRAPH 2012, Alexey Stomakhin, Russell Howes, Craig Schroeder and Joseph M. Teran.

[3] A finite element method for animating large viscoplastic flow. SIGGRAPH 2007, Adam W. Bargteil, Chris Wojtan, Jessica K. Hodgins and Greg Turk.

[4] Implicit integration for particle-based simulation of elasto-plastic solids. Pacific Graphics 2013, Yahan Zhou, Zhaoliang Lun, Evangelos Kalogerakis and Rui Wang.

[5] A material point method for snow simulation. Walt Disney Animation Studios, Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, Andrew Selle.

[6] Nonlinear continuum mechanics for finite element analysis. Bonet and Wood.

[7] A point-based method for animating elastoplastic solids. SIGGRAPH 2009, Dan Gerszewski, Haimasree Bhattacharya and Adam W. Bargteil.

[8] A unified particle model for fluid-solid interactions. Visualization and Multi-Media Lab, University of Zurich, Switzerland, B. Solenthaler, J. Schläfli and R. Pajarola.