Final Report

Harrison Sandstrom

I. Problem Statement and Motivation

The concept of gravity and spacetime curvature is commonly visualized in media through the use of an elastic sheet with a heavy object like a bowling ball in the center and smaller orbs revolving around it. While not a perfect representation of general relativity, they can be a good analogy for illustrating to the average person the concept of relativity and in a way how gravity itself works. While this can be something easily assembled in the classroom, it also lends itself to being simulated through real world testing.

II. Background and Literature Review

The use of a rubber sheet is derived from the theory of relativity. It is used in schools to teach general relativity to students, teach them about orbits, gravitational waves, and the relation of geodesics to this concept. In the concept of general relativity, the path of a freely falling object is a geodesic path through spacetime. This spacetime is curved by massive objects such as planets and stars. In a similar manner, we can represent the curvature of spacetime through the use of an elastic sheet that is deformed by a large mass[1], [3], [4].

Research that has been done on this analogy and modeling it in the real world has been done through either the use of trampolines or stretched latex. Batlle et al. analyzed the movement of a small ball orbiting a heavy mass placed in the center of a trampoline (a bowling ball with additional weights added). They modeled the trampoline shape and both a point particle model and a rolling sphere model to account for torque and moment of inertia. The experimental data found that the rolling sphere model significantly outperformed the simpler model in accurately representing the trajectories. Critically, though, they found that in representing the true model of relativity, the model failed giving apsidal precession values that were negative compared to the positive values predicted by general relativity. In this case Apsidal precision is the rotation of the elliptical orbit so that its apoapsis and periapsis (peak and valley) process around the body being orbited[1].

One of several methods evaluated for the validation of a ball on sphere model is the use of a Kinematic Match method which was specifically developed for the modelling of an elastic surface undergoing deformation under the impact of a sphere. This method uses the matching of kinematic constraints to the coinciding surfaces, matching velocities and imposing a tangency condition at the boundary of the contact. It allows for a detailed method of calculating the sphere's trajectory. This method however is not a nodal method and more complicated than would be desired in the universe model where new objects are not appearing in spacetime[2].

Another method that is better applied to the nodal contact solution is the use of implicit material point method (MPM) with the contact dynamics (CD) method to simulate soft particle interactions between the surface and the orbs. This method allows for the use of arbitrary particle shapes and assumes that deformations are localized to the contact points. It is able to create frictional contacts without the use of artificial damping or regularization. The use of friction is crucial in this method as the friction strongly impacts the deformations in the model due to the formation of stress chains which are greatly increased in strength[5].

III. Approach and Contributions

The project will have three main aspects.

(1). Implementation of a simple rolling sphere model over a deformed membrane. The membrane will be modeled as a discretized elastic element with a mesh of points and springs used, treated as an elastic shell [9], [10]. This is similar to how the mesh of points were made by Batlle et al.

(2). Implementation of contact dynamics. This will rely on the use of implicit material methods with the elastic being a large deformable surface that is deformed by what will be assumed to be perfectly hard spheres. When the spheres contact the surface they will use an iterative process to calculate their applied contact area at each step so that the tangency error will be minimized during the tracking.

(3). The last and final step is the simulation of the gravitational analogy. This will be taking the applied steps and putting them together to simulate the motion of the orbs across the surface.

A. Contributions 1 & 2

In order to model the mesh as a mesh of discretized points connected by springs, a grid of $N \times N$ points was created over a $1m \times 1m$ area as seen in figure 1.

The springs were given a stretching stiffness given by the equation

$$k_{k^{th}Edge} = \frac{\sqrt{3}}{2} Y h l_{kreflen} \tag{1}$$

where the Young's modulus $Y = 0.01GPa$ and thickness $h = 0.001m$.

The implementation of the contact mechanics using an implicit contact model (IMC) relates the distance between the center of a ball and each node.[7], [9]

$$\mathbf{r} = \mathbf{q_{ball}} - \mathbf{q_{node}} \tag{2}$$

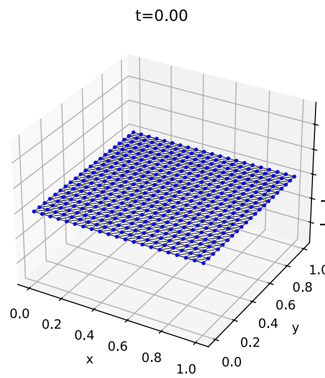

Fig. 1. Node Based flat sheet mesh

$$d = ||\mathbf{r}|| \tag{3}$$

and if the distance between the two nodes is $d < R_{ball}$ then it turns the contact on. We can then implement the contact as the force applied (F_{normal}) is

$$F_{normal} = K_{contact}(R_{ball} - d)\mathbf{n} \tag{4}$$

where $\mathbf{n}$ is the normal vector between the ball and the surface and $K_{contact}$ is the stiffness of the surface. Based on how stiff you make the surface you determine how hard the repelling force is on the objects. It effectively is describing the squishiness of the objects when they make contact. The normal force that was calculated is the force against the nodes and the balls with opposite directions as seen here.

$$\text{node force} = -\mathbf{F_{normal}} \tag{5}$$

$$\text{ball force} = +\mathbf{F_{normal}} \tag{6}$$

lastly we get the hessian of the contact which is

$$H = -K(\mathbf{n}\mathbf{n}^\top) - \frac{F_n}{d}(I - \mathbf{n}\mathbf{n}^\top) \tag{7}$$

The Jacobian for those points can be returned as

$$\mathbf{J}_c = \begin{bmatrix} \frac{\partial \mathbf{F}_{\text{node}}}{\partial \mathbf{q}_{\text{node}}} & \frac{\partial \mathbf{F}_{\text{node}}}{\partial \mathbf{q}_{\text{ball}}} \\ \frac{\partial \mathbf{F}_{\text{ball}}}{\partial \mathbf{q}_{\text{node}}} & \frac{\partial \mathbf{F}_{\text{ball}}}{\partial \mathbf{q}_{\text{ball}}} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & -\mathbf{H} \\ -\mathbf{H} & \mathbf{H} \end{bmatrix} \tag{8}$$

Once we have done this we can just get the equations for the net force and jacobians in the objective function as, with the F_c and J_c representing the contact forces being added.

$$\mathbf{J} = \mathbf{J_b} + \mathbf{J_s} + \mathbf{J_c} + \mathbf{J_v} \tag{9}$$

$$\mathbf{F} = \mathbf{F_b} + \mathbf{F_s} + \mathbf{F_c} + \mathbf{F_v} \tag{10}$$

B. *Simulation*

The simulation operates on the ability to create a mesh of given dimension and parameters. Then a ball is spawned to act as the planetary body and the simulation begins to cycle. At a certain point a secondary smaller ball is added ($t = 5s$) with some initial velocity to watch it orbit around the first ball and see its trajectory. For reference a simulation of $15s$ total time and two balls with a mesh 25×25 nodes in dimension takes approximately 2.1 hours to run with optimizations.

In order to optimize the code to run this fast the capabilities of a just in time compiler was used[7] to convert the numpy based functions into compiled code that runs significantly faster than before.

IV. Results

Figures 2-6 contain some snapshots of the 3d projection of a 3 kg ball being introduced onto the mesh before a second ball is introduced at $t = 5s$. In figures 7 and 8 we can see that the first ball settles to a z of -0.2m fairly quickly while the second ball, when introduced, very quickly skirts around the edge of the divot when introduced, but quickly loses its horizontal energy around the center and descends into the divot. The reason for the delay is that the position of the second ball is treated as being at (0,0) until it is introduced, so once it is introduced it quickly jumps to its starting position before following its normal trajectory.

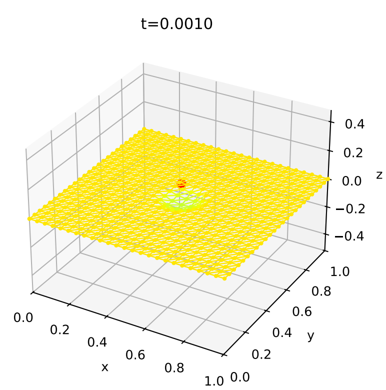

Fig. 2. Single Sphere at 0.001s into calculation

V. Commentary and Future Work

As can be seen from the plots and trajectories, this method currently has an issue with lost energy leading to the oscillations to attenuate and the spheres to quickly move to their lowest energy state at the basin they create from their weight in the elastic sheet. There are two main causes of this.

One of the causes is the addition of a viscous factor to increase the rate of convergence to allow the simulations to complete faster there is an inherent loss of energy in the system which then causes the balls to lose energy as they

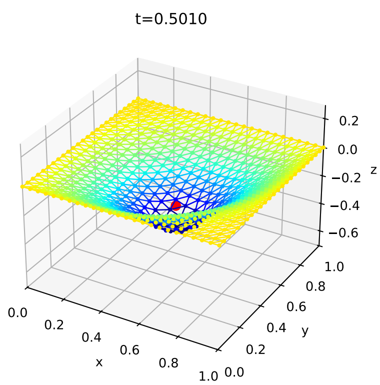

Fig. 3. Single Sphere at 0.501s into calculation

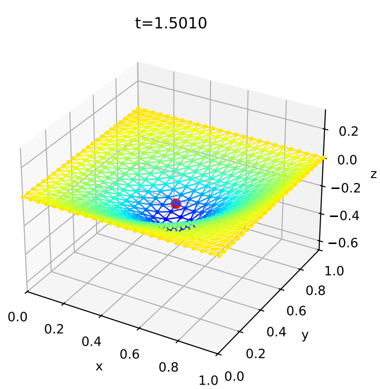

Fig. 4. Single Sphere at 1.501s into calculation

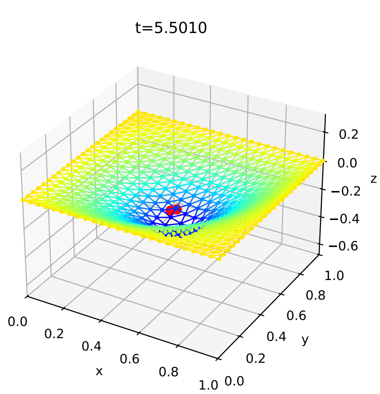

Fig. 5. Two Spheres at 5.501s into calculation

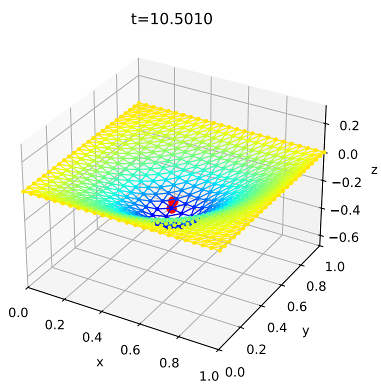

Fig. 6. Two Spheres at 10.501s into calculation

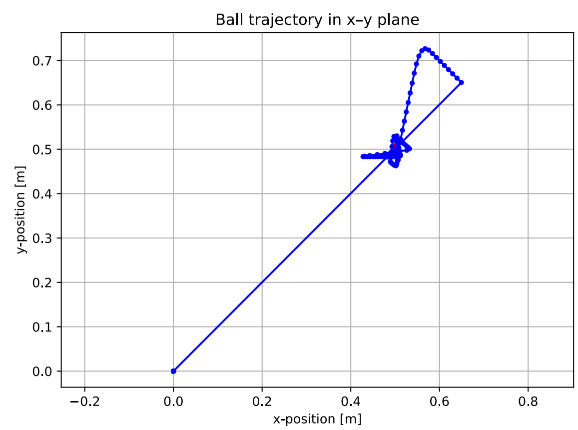

Fig. 7. Ball trajectories for both balls in x/y

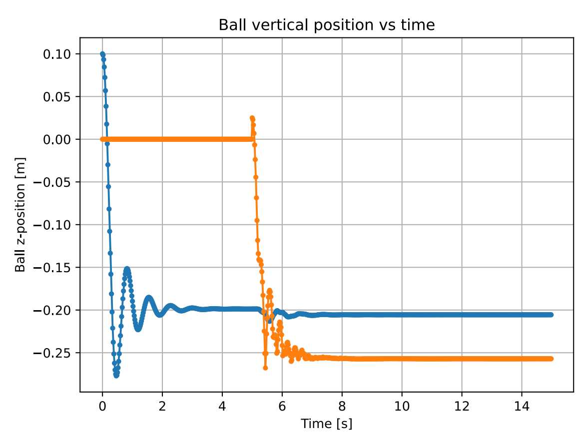

Fig. 8. Z position of the balls over time

travel around the surface and settle towards the bottom of the depression that they create.

The second cause is due to the use of a node based contact solution, the contact normal vector is between the ball and the node it is in contact with, so as the ball rolls around, it settles into the gaps between the nodes, which in effect causes a bumpy movement and leads to some lost energy.

In order to solve the problem of the viscous losses, we could use smaller time steps, which would make the simulation take longer but be more likely to converge each step. This, however would require further optimization of the code to become more feasible through either parallelization or offloading to the gpu for processing.

To solve the problem of the bouncing in between nodes, we could instead treat the contact as finding the nearest 3 nodes and finding their plane and using the normal vector of that plane in order to get the applied force to the ball. This would also prevent the ball from falling through any gaps in the mesh that might appear due to the stretching of the mesh leading to balls of small radius to slip through and not find any nodes to make contact with.

REFERENCES

[1] Batlle, P., Teixidó, A., Llobera, J., Medrano, I., & Pardo, L. C. (2019). Exploring the rubber sheet spacetime analogy by studying ball movement in a bent trampoline. *European Journal of Physics*, 40(4), 045005.

[2] Agüero, E. A., Alventosa, L., Harris, D. M., & Galeano-Rios, C. A. (2022). Impact of a rigid sphere onto an elastic membrane. *Proceedings of the Royal Society A*, 478(2266), 20220340.

[3] BOUBLIL, S., & BLAIR, D. Methods for addressing strengths and weaknesses of the rubber sheet analogy.

[4] Postiglione, A., & De Angelis, I. (2021). Experience gravity in the classroom using the rubber sheet: an educational proposal from the collaboration between University and School. *Physics Education*, 56(2), 025019.

[5] Nezamabadi, S., Radjai, F., Averseng, J., & Delenne, J. Y. (2015). Implicit frictional-contact model for soft particle systems. *Journal of the Mechanics and Physics of Solids*, 83, 72-87.

[6] Tong, D., Choi, A., Joo, J., & Jawed, M. K. (2023). A fully implicit method for robust frictional contact handling in elastic rods. *Extreme Mechanics* , 58, 101924.

[7] Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: a LLVM-based Python JIT compiler. In Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC (pp. 1–6).

[8] M. K. Jawed, Lecture_14_DiscreteShells_Simulation.ipynb, UCLA, 2025

[9] Kim, N.-H. (n.d.). Finite element analysis of contact problem (Chap. 5) [PDF]. University of Florida. https://web.mae.ufl.edu/nkim/egm6352/Chap5.pdf

[10] Jawed, M. K. (2025). Discrete simulation of slender structures [Lecture notes]. University of California, Los Angeles.