
Title here

Matthew Lenathen -
40506678

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of
BSc (Hons) Games Development

School of Computing

January 24, 2024

Authorship Declaration

I, Matthew Lenathen, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work in this dissertation following the School's ethical guidelines.

Signed:

Date:

Matriculation no: 40506678

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below one of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

Abstract

Abstract here

Contents

1	Introduction	8
2	Literature Review	9
2.1	Introduction to Cloth Simulation	9
2.2	Brief History of Cloth Simulation	9
2.3	Underlying Physics	10
2.4	Offline Cloth Simulation	11
2.4.1	Fine Cloth Interactions	11
2.4.2	Precomputing Cloth Effects	12
2.5	Real Time Cloth Simulation	13
2.5.1	Mass-Spring Systems	14
2.5.2	The Continuum Model	15
2.5.3	Cloth Simulation with Camera Capture	16
2.5.4	Cloth Simulation with Neural Networks	17
2.5.5	Shape Matching	18
2.5.6	Position Based Dynamics	20
2.6	Future Trends	23
2.7	Conclusion	23
3	Rest of Diss	24
3.0.1	Overview Of Project Content and Milestones	24
4	Chapter 2	25
5	Additional Information / Knowledge Required	26
	References	27
	Appendices	30
A	Project Overview	30
A.A	Example sub appendices	30
B	Second Formal Review Output	31
C	Diary Sheets (or other project management evidence)	32
D	Appendix 4 and following	33

List of Tables

List of Figures

1	An overview of how this implementation works	18
---	--	----

Acknowledgements

Insert acknowledgements here

I would like to thank my cat, dog and family.

1 Introduction

2 Literature Review

2.1 Introduction to Cloth Simulation

Cloth is a major aspect of realistic simulations. Present in many different industries such as: film, games development and virtual reality, cloth simulation is a challenge that combines maths, physics and programming.

Cloth simulation is a subsection of Physics-based animation. PBA uses physics and code to create a plausible looking display of realistic effects, e.g. fluid simulations, soft-body dynamics etc. Cloth is defined as a flexible plane of some material or fabric that can move in unique ways, such as bending or stretching. This makes it a tricky and unique problem to simulate accurately.

The end goal of a cloth simulation can vary depending on the intended use. In computer graphics, physical accuracy is not the highest priority, the end product is mostly focused on how it looks. This is in contrast to cloth simulations for real-world engineering problems, their focus being primarily on physical accuracy, in order to help them predict real-world scenarios. Real time usage of cloth is also very important in the games industry, and will be discussed later.

As it is featured in such a wide range of fields, the literature relating to cloth is plentiful. In this section, many papers, books and articles will be discussed and compared in order to give context to the focus of this project, Extended position based dynamics (XPBD).

2.2 Brief History of Cloth Simulation

Cloth simulation is a very active field in research, and has been for a handful of decades. The first graphic models used to mimicking cloth was developed in 1986 by Jerry Weil. Before this, cloth objects were created by mapping textures onto rigid surfaces. His method represents cloth as a 2d grid of 3d coordinates, and focused on simulating cloth held at constraint points, like a draped fabric. (Weil, 1986). This method didn't allow for cloth movement but it piqued the interest of the graphics community regarding cloth simulation.

An important development was made in 1987 by Terzopoulos et al. Their method was a general solution for representing elastic deformable objects like

rubber or paper. This method is physically-based, meaning the object is active and can respond to external forces such as gravity or wind. Their paper presented various equations on how objects such as cloth or paper should deform when presented with forces, it also describes how they integrated these equations through time, which is a key topic in simulation. This research laid the groundwork for many more physically based models, and more cloth oriented methods. (Terzopoulos, Platt, Barr, & Fleischer, 1987)

As time went on, more ways to simulate cloth were developed, such as particle based systems by Breen et al. in 1992. In particle-based systems, the simulating focused on a series of connected particles, and how they react with each other. The way these particles interacted formed the basis for the cloth dynamics. (Breen, House, & Getto, 1992) In their paper they decided to move away from continuum based methods and switch towards a particle based model. In 1995, Provot developed a similar particle system but used point masses connected with springs to represent a cloth mesh. (Provot, 1995)

In 1998, Baraff and Witkin developed a method that used implicit integration combined with the cloth being represented as a mesh of triangles. This new proposed system allowed for bigger time steps to be taken and faster simulation times. This system is still used today as the foundation for new methods. (Baraff & Witkin, 1998)

Later, in 2007, Müller et al. developed and published a paper on a state of the art method for dynamic simulations called Position Based Dynamics. This technique, along with a newer extended version, is the main focus of this project and will be discussed in a later chapter. (Müller, Heidelberger, Hennix, & Ratcliff, 2007) and (Macklin, Müller, & Chentanez, 2016)

2.3 Underlying Physics

In order to discuss the advancements and methods within the current literature, it is important to delve into the underlying physics that allow cloth simulations to function. This section will go over important principles that are crucial to creating such simulations.

Often, cloth is represented by particles. These particles have position p , and velocity v , and can be arranged in a grid structure to model a piece of cloth. In order to represent the actual geometry of these particles, they can be arranged into triangles that make up a bigger mesh. When these triangles

are small enough, that is when the simulated cloth can look realistic.

For simulations such as mass-spring systems, forces are the primary mechanism for making the particles move. In this case, spring forces between two particles are calculated using Hooke's law, and they say that if the spring is stretched or compressed more than the springs rest length, forces are applied to remedy that. Other common forces are gravity and wind and these will be applied to near all simulations regardless of if it is a force-based method or not.

As the main focus of this project is position-based dynamics, constraints must be discussed too. Constraints are pivotal to position-based dynamics because instead of relying on forces, PBD directly adjusts positions to satisfy constraints. Constraints can vary depending on the situation but usually there are some staples such as distance, bending and collision constraints. Each of these will ensure that the particles within the simulation act as realistic as possible.

2.4 Offline Cloth Simulation

Cloth simulation can generally be split into two categories: Offline and Real Time. Offline simulations are calculated, tweaked and edited before being rendered to the screen. Commonly used in film and animation, visual fidelity is of utmost importance. Offline calculations can also help in a real time scenario by precomputing effects.

2.4.1 Fine Cloth Interactions

As offline simulations are not limited by the time between each frame, they can afford to focus on realism. They can do this by scaling the number of triangles up and allowing for finer interactions like folding and wrinkling, these interactions being crucial to a convincing cloth simulation. A great example of such work is presented in a paper by Selle et al., they detail their method of creating a high resolution simulation with up to two million triangles, that can accurately capture the intricate details of cloth. (Selle, Su, Irving, & Fedkiw, 2009)

At the time of publication, cloth simulations were adding physically-based interactions like wrinkles through a separate modelling system. This can be seen in the work by Cutler et al., in which they created a system that al-

lowed the artist to add in wrinkles on top of an existing character's clothing (Cutler et al., 2005). They achieved this by splitting it into stages: wrinkle creation and wrinkle evaluation. In the wrinkle creation phase, the artists use a tool to create wrinkle patterns, which get turned into a set of forces on the surface. Next, in the wrinkle evaluation phase, the system generates these deformations based on character poses, and the artists can either choose to keep these generated wrinkles, or alter the pattern until satisfactory results are achieved.

In their results, they were able to use their system to create clothing wrinkles on hundreds of characters for a feature-length animated film. They noted that in production, close to zero hand-tweaking was needed for the wrinkles, proving its reliability.

When discussing their work, they did note that the visual results from a dynamic clothing simulation is much preferred to what they achieved with their kinematic wrinkle system. This is where the work of Selle et al., aims to improve upon.

In their paper, they aimed to create a system where fine details such as folds and wrinkles were created in a physically-based fashion, from the various forces acting on the cloth. Their setup included a mass-spring system with edge and bending spring, and a modified time integration scheme. One notable contribution from their work is how they handle collisions. They note that in order to achieve high resolution folds and wrinkles, the self collisions and interactions must be extremely robust, as they are what cause cloth to create these effects in the first place. In their paper, they created a hybrid repulsion/collision technique that works based on the last known collision. They also utilise parallelism to further improve computing times.

In their initial evaluation, the researchers put their system to the test with a twisted cloth model comprising 500,000 triangles. This example took an average of 30 minutes to calculate each frame. They found that their system was able to simulate an extremely detailed cloth with accurate folds and wrinkles, alongside realistic collision and repulsion. These advanced and realistic effects lend themselves to the offline nature of the simulation, as calculating all of these effects are extremely computationally expensive.

2.4.2 Precomputing Cloth Effects

In computer graphics, it is often said that precomputing everything is simply not viable due to how many interactions can occur, but in a paper by Kim et al., they discuss precomputing secondary cloth effects. (D. Kim et al., 2013). In this case, secondary cloth effects refer to things such as folds

and wrinkles, these are expensive simulate in real-time. In this paper, they propose a method to use many thousand CPU-hours to calculate these advanced effects in advance, based on a character motion graph. They achieve this by representing a character with two motion graphs, the primary graph simply being the movement of the character, and the secondary being the cloth dynamics.

To evaluate their method, they acquired 12 unique motion clips for the primary graph, and used ARCSim cloth simulator for the cloth motion. They found that in their massive precomputation, they were able to generate good looking cloth animations for any random paths through the primary graph. Initially their secondary graph was just one to one mapping of character poses to cloth, but this proved insufficient for realistic motion, so the secondary graph was made much more complex. Overall, the inclusion of precomputing cloth motion, helped simulate incredibly detailed secondary features of cloth, such as wrinkles and folds. This research also links into game development in their discussion section, they propose that viewing a database of precomputed cloth motion could be a great benefit when creating games, as certain scenes could be viewed and edited, also bad looking cloth could be fixed by hand to help realism within the game.

In summary, it can be seen that offline simulations, compared to real-time, prioritise visual plausibility and realistic effects. These simulations can be achieved due to their non-real-time nature, being able to compute all necessary effects in advance, and being able to tweak things that don't quite match with others. Making offline simulations great for things such as: animation, fashion design and the film industry. This level of details contrasts with real-time simulations, where they must balance out visual quality with the demands of immediate computation.

2.5 Real Time Cloth Simulation

Real time cloth simulations are created by computing the cloth dynamics at runtime, allowing interaction with the user through input and changes from the environment. As the time between frames on real time applications is small, the dynamics need to be calculated and applied efficiently, usually on the GPU. The most common use case for real time cloth simulations are in video games, and as these games also have lots of other calculations going on in the background, means optimisation is even more important. This section will discuss the literature on methods to create these simulations and how they compare with each other.

2.5.1 Mass-Spring Systems

As mentioned in section 2.3, mass-spring systems are a popular approach to cloth simulations. They represent a cloth as a grid of particles that can exert forces on each other. The way they exert forces are through springs. The first example of this system, as noted in section 2.2, was introduced by Provot (Provot, 1995).

In his paper, he posited that the current state of cloth animation at the time had a lack of realism, especially in the realm of elasticity. His proposed mesh had three different types of springs to achieve different effects. Structural springs being the most basic, it connects adjacent particles in a grid formation. Shear springs connect particles diagonally, and bend springs connect particles by skipping a particle in between, e.g. connecting particle $[i,j]$ to particle $[i+2,j+2]$.

This research successfully modelled a more realistic cloth using mass-springs and inspired research into more efficient mass-spring systems for use in real time, such as the work by Kang and Cho (Kang & Cho, 2012). It is known that the mass-spring model suffers from instability, especially at larger time steps, Kang and Cho aim to resolve this instability by proposing a more stable and accurate integration model. They note that an implicit integration scheme would help stabilise the system but it cannot be easily parallelised, so an improved explicit method was chosen.

It is noted that Hooke's law is commonly used with mass-spring systems to calculate forces, but Kang and Cho decided to use a harmonic oscillation model, allowing for integration.

With their new integration method and parallelism implemented, they were able to simulate cloth containing 16,384 particles and 161,544 springs at over 68 frames per second. This shows that mass spring systems can be a good tool for simple simulations, although in their results they also show that once the springs get to a large size, in this case 1,200,000 springs, the frame rate drops to 15. It can be seen in current day simulations that the additional complexity from all the springs, stiffness tuning and overall accuracy lead to mass-spring systems not being the optimal method for real time cloth simulations.

2.5.2 The Continuum Model

While representing cloth with particles and springs can be straightforward and relatively pleasing to look at, there is a downside to this method, the real world representation of cloth. It is very hard to come up with a spring model that accurately represents various cloth materials such as nylon. This is where the continuum system for cloth fits in. The continuum method treats cloth as a continuous material made up of triangles rather than particles with space in-between, and instead of looking at spring forces, continuum looks at how each triangle is stretched and compressed.

As mentioned in section 2.2, Baraff and Witkin proposed one of the first continuum models in their paper (Baraff & Witkin, 1998). A few others attempted such models such as (Terzopoulos et al., 1987). Although in their implementation, the damping forces weren't explored in much depth. Another paper by (Carignan, Yang, Thalmann, & Thalmann, 1992) built upon the work by Terzopoulos et al. and improved the damping forces by adding a force that damps cloth stretch and shear. They used an explicit integration scheme so was unfortunately limited by small time steps. These works lead Baraff and Witkin to develop their model. Their goal was to come up with a method that allows for larger time steps to be taken in simulation, as most at the time were bottlenecked by small time steps to remain stable. In their method, they employ an implicit integration scheme which allowed for much better performance, when combined with the simple formulation of internal forces. With this continuum representation, they were able to successfully model the common forces like shearing, bending and stretching.

In their results, they tested multiple setups including clothing. They found that changing the bend stiffness factor, didn't affect in running times a great deal, which explicit simulators wouldn't be able to do. They were also able to estimate their performance in terms of number of particles, and found that it was slightly better than $O(n^{1.5})$ which was what they expected.

This paper laid the groundworks for many continuum implementations, one of which being a paper on data-driven elastic models for cloth simulation (H. Wang, O'Brien, & Ramamoorthi, 2011). Their goal was to more closely represent how different materials are simulated, ideally close to their real world counterpart. It does this by representing cloth with a piecewise linear elastic model. They explain that real world cloth items such as clothes often have anisotropic properties from the woven nature of cloth, and that this is often ignored by most cloth simulations for simplicity. A part of their re-

search was a way to measure real life samples of cloth, construct a database of parameters and use them directly in their cloth simulation which was a very interesting way to get realistic results.

Finite element analysis is a way to break down the cloth into discrete elements such as triangles, and the physical behaviour is calculated based on the material properties and forces acting on it. In 2020, Kim discussed how most cloth simulation are either made from finite element methods, position based dynamics or mass-spring systems, and that the Baraff-Witkin model doesn't really fit into these categories. In his paper he formulated a version of the Baraff-Witkin model but entirely in terms of finite elements (T. Kim, 2020).

2.5.3 Cloth Simulation with Camera Capture

As cloth simulation is such a wide field of research, there are many alternative methods and interesting papers to look at, one such idea is SimulCap by (Yu et al., 2019). SimulCap is a method of capturing human movement while also capturing dynamic details such as folds and wrinkles. It uses a RGBD (RGB depth) camera. Their problem with other recent implementations were the lack of realism, in part due to the human and cloth being considered as one piece of geometry, and also the results not being editable, which is a huge aspect of a lot of use cases, i.e. virtual dressing applications. So, for SimulCap, the authors decided to combine a layered representation of the human body, while also having a separate cloth simulation that matches the human movement. For their cloth simulation, they chose to use a mass-spring model for simplicity and also an explicit integration scheme with a small time step for stability. In their results, they compared their implementation with DoubleFusion, another previous system by (Yu et al., 2018), which is a state-of-the-art system for real time human capture. The authors found that SimulCap achieves a much more realistic cloth representation, mainly due to the separation of geometry as described earlier. The limitations of this system lie in the variation of clothing, e.g. thick jumpers remain challenging to simulate, also due to the simple nature of the cloth simulation, the realism could be improved upon by choosing a more advanced technique.

In a more recent paper by Wang et al., in 2023, they address the problem of extracting cloth movement combined with clothing motion, by proposing a novel method of capturing the human motion and clothing separately using double layer neural radiance fields (NeRF) (K. Wang, Zhang, Cong, & Yang,

2023). NeRF is a method of reconstructing a 3D scene using sparse 2D images. This makes it a perfect method for cloth simulation using camera capture. During the optimisation phase, they introduce a physics-aware cloth simulation network to properly handle the cloth dynamics to ensure realism.

They comment on the previous implementation, SimulCap, by saying it failed to track the motion of the clothing to achieve editable clothing on humans, which is a prerequisite to most VR/AR apps. Their solution, using NeRFs, model both the body and the clothing in template space, which can then be transformed into observation space using deformation fields. Then, using the two NeRFs from before, the rendered images can be synthesized.

This approach addresses the problems with previous camera capture implementations. Its physics network is taught from simulation data, to constrain the clothing and ensure physically plausible clothing deformations. In their comparison to state of the art methods, which include DeepCap and NerfCap, they found that their implementation vastly outperformed the other methods without a cloth template, and achieved a much higher accuracy during geometry tracking.

2.5.4 Cloth Simulation with Neural Networks

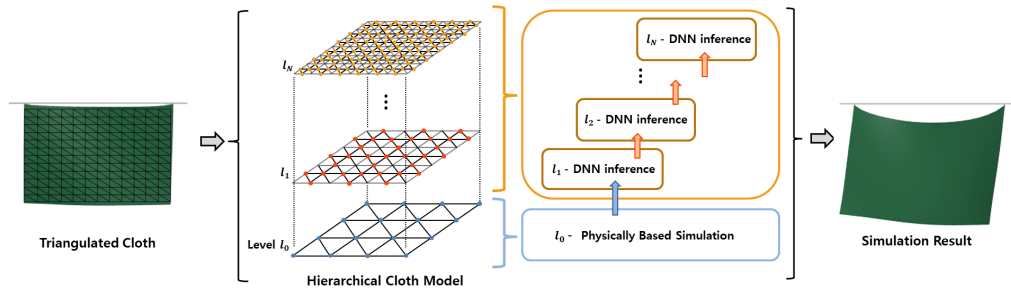
Another interesting alternative method to cloth can be found in the research of (Bertiche, Madadi, & Escalera, 2022), a cloth simulation framework that uses an unsupervised deep learning methodology. This has been attempted in the past but the other approaches do not handle cloth dynamics. The authors note that while supervised learning systems have been used before, it requires huge volumes of data, and it has to be repeated for every garment/body, so they chose an unsupervised approach. For their cloth representation, they chose to use a particle based system that also considers external forces like wind and collision, this cloth is then skinned to a 3D model. After training their model, they compared their results to other unsupervised approaches such as (Bertiche, Madadi, & Escalera, 2021). They evaluated their framework with multiple different body motions such as jumping and spinning. They found that their approach was the only one capable of achieving dynamic cloth effects. In terms of limitations, unsupervised learning compared to supervised takes much longer and is more computationally expensive, and neural cloth has yet to implement cloth self-collision which is a crucial component of most simulators.

Another example of cloth simulations using neural networks can be found in the works of Oh et al., where they propose a hierarchical cloth simulation method that combines traditional physical based methods with deep neural

networks (DNN) (Oh, Lee, & Lee, 2018). They note that proper physically based cloth animation is expensive, so being able to utilise DNNs to replace the complex computation with data inferred from the network will reduce the cost significantly.

Their proposed method would split the simulation into two parts. Firstly, a low resolution or coarse model of the cloth is calculated using a normal physically based method. Then, the DNN is used to calculate the finer levels of the cloth, reducing the computation cost. They note that due to only calculating the coarsest level of cloth using a physically-based method, the overall cloth simulation won't be as accurate as a fully physically-based model but it successfully prevents errors stemming from the DNN, and guarantees a reliable simulation. This can be seen in Figure 1.

Figure 1: An overview of how this implementation works



In their results, they constructed the simulation with three levels, one physically based used projective dynamics, and two DNN models. Projective dynamics will be covered in section 2.5.6. They compared time performance of their method against a normal implicit integration simulation, and against a projective dynamics implementation with an optimised algorithm applied to it called ADMM (alternating direction method of multipliers) (Narain, Overby, & Brown, 2016). They found that their hierarchical model that uses DNN improved performance, especially on configurations with many particles, e.g. flags and curtains. They do admit that the accuracy is not on the same level as completely physically based implementations but the results generated were of a high standard and were fast to generate. One final thing to note is that if the cloth is overstretched by external forces, the DNN can sometimes give unusual wrinkles near the bottom of the cloth. They aim to improve this in future work by updating the DNN model to also hold information such as velocity and momentum.

2.5.5 Shape Matching

Shape matching is a technique that allows an object to deform but tries to maintain the form of the original shape. In 2005, Müller et al., utilised shape matching to come up with a new method to simulate deformable objects (Müller, Heidelberger, Teschner, & Gross, 2005). It does this by replacing energies by geometric constraints, and forces by the distance of current positions to goal positions. These goal positions are determined by shape matching, given a rest state. They mention that because the points are always drawn to goal positions, the overshooting issue with explicit methods is completely removed.

In their algorithm, their idea is given a set of particles with an initial configuration, every time step the particles are moved towards their goal position, and these goal positions are determined by the original configuration of the shape. In terms of performance, it depends on the number of points used, and they note it is hard to compare it to another model such as mass-spring systems, as the technique depends on the number of clusters chosen, which cannot be reflected in the other models.

Overall, this shape matching approach to simulation provides an unconditionally stable simulation that is inexpensive to compute, although the model is not physically accurate.

This research inspired a paper on cloth simulation using shape matching in 2013 (Bender, Weber, & Diziol, 2013). As mentioned before, using shape matching for simulations proved unconditionally stable, so a model specifically designed for cloth simulation was created. They note that normal shape matching approaches that use regions, cannot differentiate between shearing, bending and stretching, which are crucial to represent cloth accurately. To combat that, they introduce a novel approach to use a multi-resolution model instead of region based. This multi-resolution model will allow for the simulation of fabrics that have different stretching/shearing stiffness. It is also noted that this method is fast due to it scaling linearly with the size of the cloth.

A shape matching approach that used regions was attempted before (Stumpp, Spillmann, Becker, & Teschner, 2008). But in this model, it introduces forces that influence the bending stiffness and depend on the size of the model, and this is undesirable in cloth simulation. Another problem is that due to small regions within a large model, high stretching and shearing stiffness cannot be obtained. This lead to the development of the multi-

resolution approach.

In terms of performance, they simulated meshes between 10000 and 100000 triangles to test their scalability, and confirmed their linear relationship between computation time and number of triangles. They also implemented a GPU version of the method which resulted in a much faster compute time. They found in similar setups, the multi-resolution method was around 9 times faster than a standard mass-spring system, and around 15 times faster than a corotational finite element method, a method created by Eitzmuss et al. (Eitzmuß, Keckeisen, & Straßer, 2003). Overall, the multi-resolution approach provided a very noticeable speed-up compared to other approaches, at the cost of physical accuracy.

Can write about this too <https://dl.acm.org/doi/10.1145/3134302.3134332>

2.5.6 Position Based Dynamics

A great number of techniques discussed above have been force based, and these forces are used to calculate accelerations. Finally these accelerations are integrated to find the velocities and positions of particles. The Position based dynamics (PBD) technique takes a unique approach to physically based animations by omitting the velocity layer and directly working on the positions of particles. (Müller et al., 2007) The main advantage is the controllability, the integration instability discussed in section 2.5.1 is completely avoided due to the lack of explicit integration. As this is a real time technique, it is greatly suited to games development, as it is very helpful to have direct control over positions of objects within the game.

It was noted that even though PBD was the first fully position based approach, there had been other attempts to utilize positions in this way, such as (Jakobsen, 2001). In his approach, Jakobsen manipulated the positions directly using a Verlet integrator. The authors mention that Jakobsen didn't go into depth in terms of constraints, which is crucial in cloth simulation. PBD aims to improve upon it by giving a general approach to constraints.

To fully discuss the technique, the algorithm must be analysed. In the algorithm, an object is represented by N vertices and M constraints. The vertices are the particles which have mass m , position x and velocity v . The constraints contain a few more properties: cardinality, a function, set of indices, stiffness and a type of either equality or inequality. It is noted that the equality property is satisfied if the function of that constraint is equal to 0, and likewise if it has type inequality, it is satisfied if the function is more

than equal to zero. The stiffness affects the strength of the constraints and it ranges from 0 to 1.

Algorithm 1 Position Based Dynamics Algorithm

```

1: for all vertices  $i$  do
2:   initialize  $x_i = x_i^0$ ,  $v_i = v_i^0$ ,  $w_i = 1/m_i$ 
3: end for
4: loop
5:   for all vertices  $i$  do
6:      $v_i \leftarrow v_i + \Delta t w_i f_{ext}(x_i)$ 
7:   end for
8:   dampVelocities( $v_1, \dots, v_N$ )
9:   for all vertices  $i$  do
10:     $p_i \leftarrow x_i + \Delta t v_i$ 
11:   end for
12:   for all vertices  $i$  do
13:    generateCollisionConstraints( $x_i \rightarrow p_i$ )
14:   end for
15:   loop solverIterations times
16:    projectConstraints( $C_1, \dots, C_{M+M_{coll}}, p_1, \dots, p_N$ )
17:   end loop
18:   for all vertices  $i$  do
19:     $v_i \leftarrow (p_i - x_i)/\Delta t$ 
20:     $x_i \leftarrow p_i$ 
21:   end for
22:   velocityUpdate( $v_1, \dots, v_N$ )
23: end loop

```

Now it can be broken down to see what each part does. Lines 1 to 3 is just initialising each particle with initial position, velocity and inverse mass. Line 6 is where external forces get added to the system, Müller et al. notes that if gravity is the only external force then the line becomes $v_i \leftarrow v_i + \Delta t g$ where g is the gravitational acceleration. Next, on line 10, the new positions are estimated with a explicit Euler integration step, they are then adjusted in the line 15 loop as many times as needed until all constraints are satisfied. Once completed, the line 18 loop then moves the positions to the calculated position and updates the velocity.

This algorithm can be classed as unconditionally stable as the integration method in lines 19 and 20 do not use future positions/velocities, but instead calculate physically possible configurations and move the vertices to them.

In the paper, the authors also discuss their implementation of a cloth simulator for use in games. Their simulator accepts a triangle mesh and creates two constraints, stretching and bending constraints. Much like the springs in mass-spring systems, these constraints force the positions of the particles to satisfy them in order to continue the simulation, this relates to the loop in line 15 of the algorithm.

To test their cloth simulation, they integrated the method into a game environment and conducted various experiments. They note that a positive to this method is that their bending and stretching terms are independent which can lead to more results when testing values.

In one of their tests they used one way coupling to link cloth to a stationary object, and two way coupled the bottom of the cloth to a rigid body, it resulted in realistic twisting and swaying, but more importantly ran at more than 380fps which showcases the efficiency of this method.

They were also able to test tearing of the cloth. Their tearing process is as follows: when an edge stretches past a threshold value then it is split, with the triangles above being assigned to the original vertex and the lower triangles assigned to the duplicate. They claim this is also extremely stable.

Overall, this paper was a very interesting approach to cloth simulation at the time, and influenced a great deal of work into real time cloth, eventually being used in Nvidia's physX software which in turn is used very frequently in game development.

Extended Position Based Dynamics In 2016, Müller collaborated with Miles Macklin and Nuttapong Chentanez to extend position based dynamics and improve upon the original method (Macklin et al., 2016). Their prob-

lems with the original stemmed with iteration count and time step affecting the stiffness of the simulated object, specifically: constraints become stiff as iteration count increases/time step decreases. This effect is discussed in the 2014 survey on position-based methods (Bender, Müller, Otaduy, Teschner, & Macklin, 2014).

Their solution to this problem involved creating a new constraint formulation that corresponds to elastic potential energy. It also introduces a total Lagrange multiplier, which is a technique used to find the local maxima and minima of a function that is subject to constraints, e.g. in cloth this could be a distance constraint. This Lagrange multiplier allows the constraints to be solved irrespective of time step.

To test this on cloth, they set up the same cloth configuration for PBD and XPBD and compared them. They found that PBD became progressively stiffer as the iteration count increased but XPBD remained consistent. They also found a small downside of XPBD in terms of performance per-iteration, but this was less than 2% of total simulation time.

In the limitations, the authors note that the method is only an approximation of an implicit Euler integrator, but this still creates very realistic real time results. Even so, if the application requires great accuracy, it would be advisable to choose a more traditional simulation method.

Other PBD methods todo: can talk about other implementations that utilize PBD, such as projective dynamics by Bouaziz et al.

bridging finite element methods and PBD <https://dl.acm.org/doi/10.1145/2601097.2601116>

2017 survey on PBD, lots more info <http://mmacklin.com/2017-EG-CourseNotes.pdf>
maybe some future trends in here.

2.6 Future Trends

todo: this section will discuss any future trends that cloth simulation could have

2.7 Conclusion

todo: conclusion of lit review

3 Rest of Diss

3.0.1 Overview Of Project Content and Milestones

This is a sub sub section with a list of bullet points.

- A working X, that will be used for this investigation.
- Investigation of current tools and their potential use during an investigation of X .
- Programming of X with related frameworks Y and Z.
- That is all.

4 Chapter 2

The following bibliographic information on Writing a literature review is contained in the *bibliography.bib* file. The template automatically starts new chapters on a new page. The associated guidelines tell you what the available styles do and also how to structure a report. There is a section break on this page that you should be careful NOT to delete otherwise the references and appendices will be numbered continuously with the rest of the document.

5 Additional Information / Knowledge Required

Experience with Linux and managing Virtual machines, networking. So on and so forth...

References

- Baraff, D., & Witkin, A. (1998). Large steps in cloth simulation. In *Proceedings of the 25th annual conference on computer graphics and interactive techniques* (p. 43–54). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/280814.280821> doi: 10.1145/280814.280821
- Bender, J., Müller, M., Otaduy, M. A., Teschner, M., & Macklin, M. (2014). A survey on position-based simulation methods in computer graphics. In *Computer graphics forum* (Vol. 33, pp. 228–251).
- Bender, J., Weber, D., & Diziol, R. (2013). Fast and stable cloth simulation based on multi-resolution shape matching. *Computers and Graphics*, 37(8), 945–954. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0097849313001283> doi: <https://doi.org/10.1016/j.cag.2013.08.003>
- Bertiche, H., Madadi, M., & Escalera, S. (2021). *Pbns: Physically based neural simulator for unsupervised garment pose space deformation*.
- Bertiche, H., Madadi, M., & Escalera, S. (2022, nov). Neural cloth simulation. *ACM Trans. Graph.*, 41(6). Retrieved from <https://doi.org/10.1145/3550454.3555491> doi: 10.1145/3550454.3555491
- Breen, D. E., House, D. H., & Getto, P. H. (1992). A physically-based particle model of woven cloth. *The Visual Computer*, 8, 264–277.
- Carignan, M., Yang, Y., Thalmann, N. M., & Thalmann, D. (1992, jul). Dressing animated synthetic actors with complex deformable clothes. *SIGGRAPH Comput. Graph.*, 26(2), 99–104. Retrieved from <https://doi.org/10.1145/142920.134017> doi: 10.1145/142920.134017
- Cutler, L. D., Gershbein, R., Wang, X. C., Curtis, C., Maigret, E., Prasso, L., & Farson, P. (2005). An art-directed wrinkle system for cg character clothing. In *Proceedings of the 2005 acm siggraph/eurographics symposium on computer animation* (p. 117–125). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/1073368.1073384> doi: 10.1145/1073368.1073384
- Etzmuß, O., Keckeisen, M., & Straßer, W. (2003). A fast finite element solution for cloth modelling. In *11th pacific conference on computer graphics and applications, 2003. proceedings.* (pp. 244–251).
- Jakobsen, T. (2001, 01). Advanced character physics. In *Game Developers*

Conference Proceedings.

- Kang, Y.-M., & Cho, C.-S. (2012). Photorealistic cloth in real-time applications. *Computer Animation and Virtual Worlds*, 23(3-4), 253–265.
- Kim, D., Koh, W., Narain, R., Fatahalian, K., Treuille, A., & O'Brien, J. F. (2013). Near-exhaustive precomputation of secondary cloth effects. *ACM Transactions on Graphics (TOG)*, 32(4), 1–8.
- Kim, T. (2020). A finite element formulation of baraff-witkin cloth. *Computer Graphics Forum*, 39(8), 171–179. Retrieved from <https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14111> doi: <https://doi.org/10.1111/cgf.14111>
- Macklin, M., Müller, M., & Chentanez, N. (2016). Xpbd: position-based simulation of compliant constrained dynamics. In *Proceedings of the 9th international conference on motion in games* (pp. 49–54).
- Müller, M., Heidelberger, B., Hennix, M., & Ratcliff, J. (2007). Position based dynamics. *Journal of Visual Communication and Image Representation*, 18(2), 109–118.
- Müller, M., Heidelberger, B., Teschner, M., & Gross, M. (2005, jul). Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3), 471–478. Retrieved from <https://doi.org/10.1145/1073204.1073216> doi: 10.1145/1073204.1073216
- Narain, R., Overby, M., & Brown, G. E. (2016). Admm projective dynamics: fast simulation of general constitutive models. In *Proceedings of the acm siggraph/eurographics symposium on computer animation* (p. 21–28). Goslar, DEU: Eurographics Association.
- Oh, Y. J., Lee, T. M., & Lee, I.-K. (2018). Hierarchical cloth simulation using deep neural networks. In *Proceedings of computer graphics international 2018* (p. 139–146). New York, NY, USA: Association for Computing Machinery. Retrieved from <https://doi.org/10.1145/3208159.3208162> doi: 10.1145/3208159.3208162
- Provot, X. (1995). Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface* (pp. 147–147).
- Selle, A., Su, J., Irving, G., & Fedkiw, R. (2009). Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction. *IEEE Transactions on Visualization and Computer Graphics*, 15(2), 339–350. doi: 10.1109/TVCG.2008.79
- Stumpp, T., Spillmann, J., Becker, M., & Teschner, M. (2008, 01). A geometric deformation model for stable cloth simulation. In (p. 39–46). doi: 10.2312/PE/vriphys/vriphys08/039-046
- Terzopoulos, D., Platt, J., Barr, A., & Fleischer, K. (1987). Elastically

- deformable models. In *Proceedings of the 14th annual conference on computer graphics and interactive techniques* (pp. 205–214).
- Wang, H., O’Brien, J. F., & Ramamoorthi, R. (2011, jul). Data-driven elastic models for cloth: Modeling and measurement. *ACM Trans. Graph.*, 30(4). Retrieved from <https://doi.org/10.1145/2010324.1964966>
doi: 10.1145/2010324.1964966
- Wang, K., Zhang, G., Cong, S., & Yang, J. (2023, June). Clothed human performance capture with a double-layer neural radiance fields. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)* (p. 21098-21107).
- Weil, J. (1986). The synthesis of cloth objects. , 20(4).
- Yu, T., Zheng, Z., Guo, K., Zhao, J., Dai, Q., Li, H., . . . Liu, Y. (2018). *Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor*.
- Yu, T., Zheng, Z., Zhong, Y., Zhao, J., Dai, Q., Pons-Moll, G., & Liu, Y. (2019, June). Simulcap : Single-view human performance capture with cloth simulation. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition (cvpr)*.

Appendices

A Project Overview

A.A Example sub appendices

...

B Second Formal Review Output

Insert a copy of the project review form you were given at the end of the review by the second marker

C Diary Sheets (or other project management evidence)

Insert diary sheets here together with any project management plan you have

D Appendix 4 and following

insert content here and for each of the other appendices, the title may be just on a page by itself, the pages of the appendices are not numbered, unless an included document such as a user manual or design document is itself pager numbered.