Andreas Glomsrud

Deformable Body Based Salmon Model

For iterative development and testing of equipment within Isaac Sim

Master's thesis in Simulering og Visualisering
Supervisor: Adam Leon Kleppe
June 2023

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

Andreas Glomsrud

Deformable Body Based Salmon Model

For iterative development and testing of equipment within Isaac Sim

Master's thesis in Simulering og Visualisering
Supervisor: Adam Leon Kleppe
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of ICT and Natural Sciences

NTNU
Norwegian University of
Science and Technology

Kunnskap for en bedre verden

DEPARTMENT OF ICT AND NATURAL SCIENCES

MASTER THESIS IN SIMULATION AND VISUALIZATION

Deformable Body Based Salmon Model

For iterative development and testing of equipment within Isaac Sim

Author:
Andreas Glomsrud

Supervisor:
Adam Leon Kleppe

Date

16.06.2023

Abstract

English

In this thesis, the capabilities of Isaac Sim as a simulator of Soft Bodies have been utilized to create a robust and dynamic Fish Model. The model is based partially on previous knowledge obtained during work in the summer of 2022 in collaboration with MANULAB under the Fisk 4.0 Research Program. The model consists of a 3D-modelled Salmon, a rigid body structure, and a Finite Element Method mesh that together create a robust simulation model. The model can be utilized for the iterative testing of equipment and factory conveyor layouts.

Norsk

I denne oppgaven utnyttes Isaac Sim sin kapasitet for ”Soft Body” simulering for å lage en robust og dynamisk modell av Fisk. Modellen er delvis basert på tidligere kunnskap innhentet under arbeid utført i løpet av sommeren 2022 i samarbeid med MANULAB under Forskningsprogrammet Fisk 4.0. Modellen består av en 3D-modellert laks, en ”Rigid Body” modell og en Finite Element Method mesh som sammen skaper en robust simuleringsmodell. Modellen kan brukes til iterativ testing av utstyr og fabrikkoppsett for transportbånd.

Table of Contents

List of Figures

1 Introduction

There is currently a push within the Norwegian fishing industry to modernize methods and equipment to create more efficient and accurate processing streams. Fisk 4.0 is a research project spanning from 2022-2028 led by “Møreforskning” in collaboration with NTNU’s Department of Ocean Operations and Civil Engineering. The project is also partnered with many companies working within the ocean sector such as Optimar, Teknotherm, and GCE Blue Maritime. The focus of the project is to research the interactions between biological and technical expertise to improve production and processes within the fish industry through the implementation of automation and sensor technology. [The Research Council of Norway, 2023]

There is however one key problem hindering the creation of such software and hardware. There are no industry-standardized ways of iterating and testing systems that include interactions with non-rigid objects. Testing prototypes and theories for this kind of machinery is therefore both time-consuming and expensive. There are systems built for production lines that work with rigid materials such as steel and aluminum. These systems cannot be used for the purpose of simulating interactions between robots and fish due to the non-rigid nature of fish.

Finding a simulation method that can solve this problem will allow a single person to test their design within a fully digital environment. This iterative testing within a digital realistic representation will save time while reducing the cost of development. This reduction in time and resource spending will contribute to lower production costs and higher efficiency. There are programmes fit for this kind of iterative testing, one of the more ambitious projects in recent times is NVIDIA Omniverse. Specifically for this paper NVIDIA Isaac Sim. A simulation platform centered around realistic robotics simulation and AI development.

Figure 1: Robot systems developed and tested within Isaac Sim

Source: [NVIDIA, 2023a]

During the summer of 2022, a prototype simulator that would facilitate the interaction between a simulated fish and robotics was developed within the simulation tool Unity3D. The system was built to mimic the dynamics of a dead fish by utilizing multiple rigid body objects with joints with a visual mesh of a salmon. The digital fish model was separated into 5 parts according to this schematic.

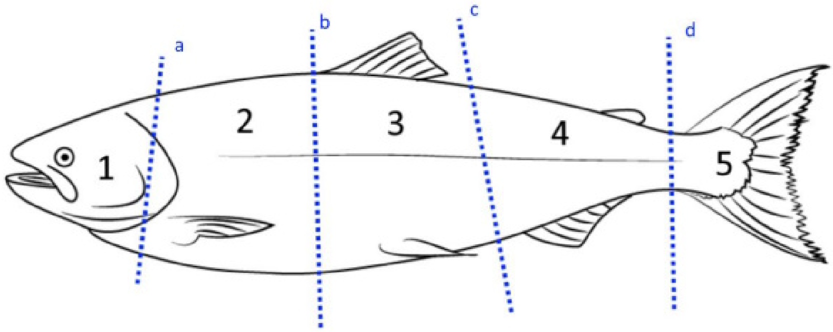

Figure 2: Schematic for division of fish

The joints were limited in movement in the x,y, and z axis by data measured from a simple bend test performed on selected fish. A dataset was collected by measurements done on fish from a fish farm. The fish had died of natural causes meaning that the data was a collection of differing sizes of fish from different stages of development. Digital fish were randomly generated with randomized assets with functions based on this dataset. Randomization was done on the length of the fish, weight, width, and height.

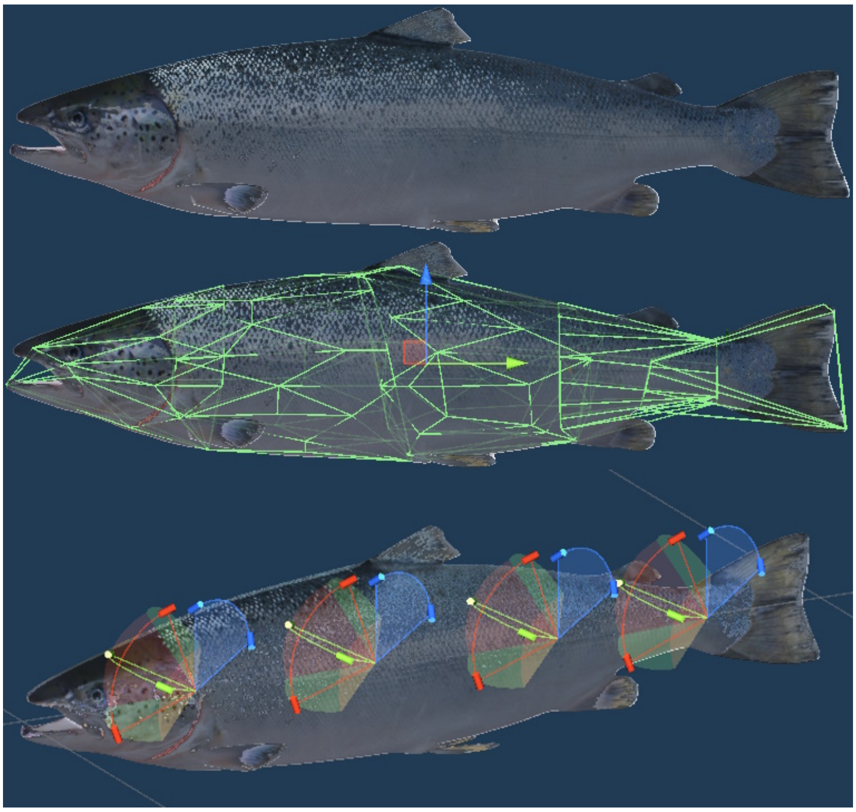

Figure 3: Unity Salmon model with colliders and joint limits

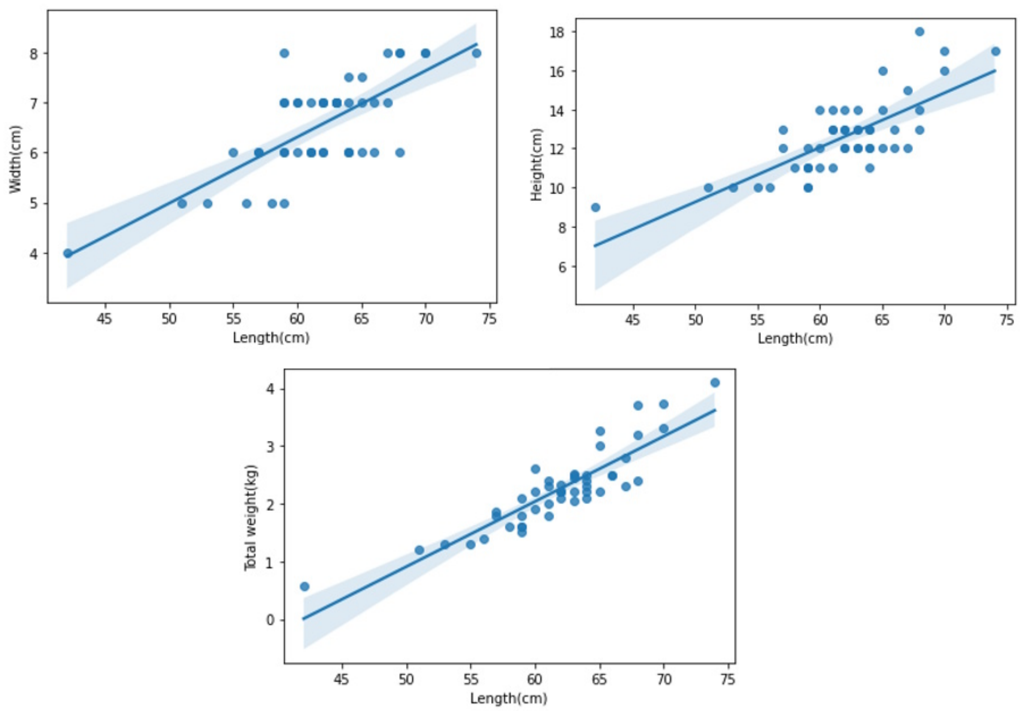

Figure 4: Weight, Width and Height of tested salmon distributed by their length

Due to either lack of experience or limitations of the engine, the resulting simulation was rather unstable. It is not certain what was the exact culprit of this instability but it was more than likely caused by the application of too many constraints upon the model. The instability could also have been a result of poor collider creation. Due to this instability, the platform was not suitable for anything but the most basic of analysis.

The goal of this project is to utilize the previously collected data along with previous knowledge gained from building the Unity simulator to create an improved system within the NVIDIA Omniverse Isaac Sim platform. The goal of the simulator is to create a model of a fish that reacts "naturally" to forces applied while working with built-in methods of interaction within the engine.

2 Theory

2.1 Finite Element Method

The finite element method is a generalized method for solving complex differential equations within the field of engineering. The method revolves around subdividing a complex system into simple finite pieces with individually known and well-defined rules of interaction. Originally used to solve stress analysis, since used in many other applications like flow, thermal and piezoelectric analysis. The method aims to mathematically determine the distribution of some variables in a system. Be it electrical charges in a material, pressure distribution in a pipe system, or the stress distribution of a support beam. The procedure of computational modeling using the FEM broadly consists of four steps:

- Modelling of the geometry
- Meshing (discretization)
- Specification of material properties
- Specification of boundary, initial and loading conditions

[Liu and Quek, 2013]

Modeling of geometry involves the creation of a digital representation of the system or object you wish to analyze. This is normally done by utilizing CAD (Computer aided design) software. While modeling for FEM analysis there is an active trade-off between model accuracy and solver speed. Especially when the object in question contains a lot of complex geometry. Complex shapes like curves are computationally expensive as they cannot be simulated utilizing mathematically simple curve equations.

The meshing of an object is performed to discretize the geometry to be able to analyze complex systems using relatively simple functions. Generally, the approach is to approximate the geometric shape of an object utilizing the simplest shape possible. For graphical representations, the simplest shape possible is a triangle. If modeling in 3 dimensions tetrahedrons are used instead. Re-meshing an object is a complex task but can be automatically generated by re-meshing algorithms.

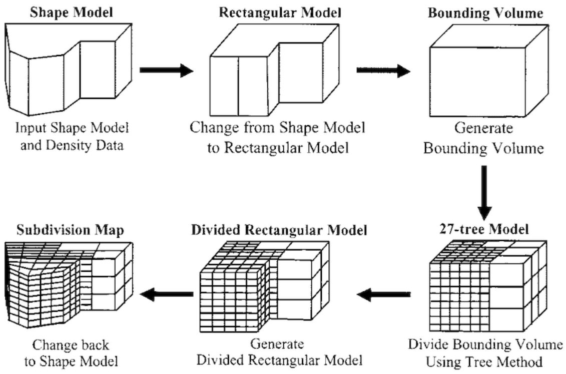

Figure 5: Example of 3d meshing algorithm

Source: [Nagakura et al., 2001]

To analyze different materials and interactions, material properties must be assigned to all parts of the meshed object. An object can be constructed using many different materials and for every new material in the wished analysis, the properties need to be assigned. For stress analysis, Young's modulus and other physical properties describing how it will react to outside factors will need to be assigned.

2.2 Rigid body Simulation

Within the world of simulation Rigid Body dynamics is one of the most utilized methods of simulation interactions between objects. Simulation of a rigid body generally requires 3 sets of variables to be known:

- Mass and center of the object
- Geometrical limits of the object
- World variables

With these values set you can simulate basically any rigid object and its interaction with the world. Rigid Body simulation is extremely efficient since it relies on relatively simple and well-known equations for acceleration, rotation, and positioning. Rigid Body simulation is built upon Rigid Body Dynamics, generally defined as the analysis of how rigid bodies interact and move given external forces. A rigid body is a physical object that does not deform given the application of outside forces. An example of this is, for example, a rock, a Bar of steel, a tree, etc... Most objects within the real world cannot be described as true mathematical "rigid bodies" because they usually have a point of failure where if enough force is applied, they either deform or break. By reducing the scope to be within these limits, however, we can mathematically calculate the movements and interactions of rigid objects by utilizing relatively simple mathematical rules.

Most rigid bodies within a simulated program contain two main components. A Rigid body module that handles the calculations of forces applied to the object and contains all the information about its mass and other contributing factors. And a collision component that encompasses the geometrical limits of the object and solves collisions between the object and the environment. These naturally work together to create a rigid body simulation of an object. A collider for any given object needs to be defined within the simulation to allow for a realistic simulation of the object. The collider and the visual mesh are not guaranteed to conform to each other without manual adjustment or by utilizing specialized algorithms.

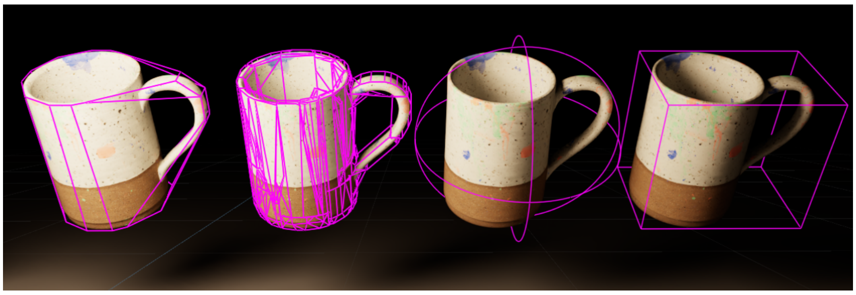

Figure 6: Multiple Collider approximations applied to a Cup

Source: [NVIDIA, 2023b]

2.3 PhysX Deformable Bodies

PhysX is an open-source physics engine developed and supported by NVIDIA; PhysX is specifically designed to work well with NVIDIA CUDA technology to accelerate graphics and physics processing.

PhysX has the capability to simulate non-rigid bodies by utilizing a combination of FEM and tetrahedral meshes. PhysX does include automatic systems for the generation of these tetrahedral meshes. The visual mesh of an object can be used as the basis for the other two required meshes, however they could also be manually designed.

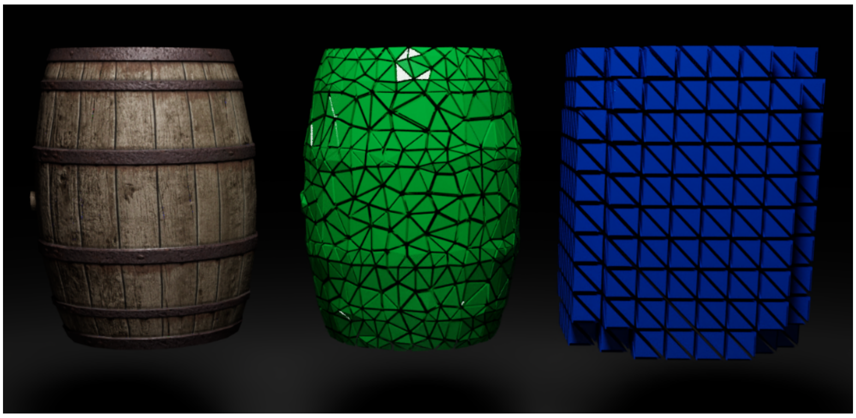

Figure 7: Visual, Collision and Deformable mesh of a barrel

Source: [NVIDIA, 2023c]

With the visual mesh as an input Tetrahedralization algorithms create the collision mesh required for the simulation of Deformable bodies. By utilizing these algorithms, the resulting collision mesh will accurately represent the input body and will therefore collide with the environment as expected. This collision mesh is then used to populate a Voxel representation of the input shape. This voxelized mesh will be used within the FEM simulation to determine the deformation of the object.

As described in FEM theory every FEM simulation requires some baseline values for the properties of the material you wish to simulate, These values can be assigned using a Physics material. The properties being assigned by this material are Density, Friction coefficient, Young's Modulus, Poisson's Ratio, and values controlling the dampening of the particles. These all contribute to the simulation and allow the user to customize the properties of the simulation to simulate different types of materials.

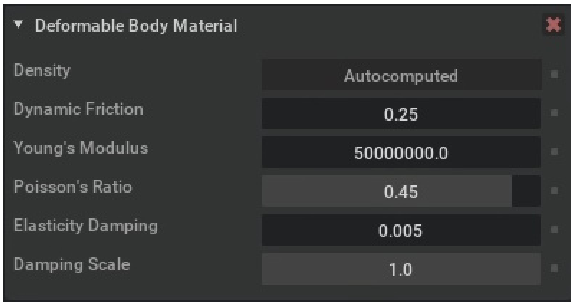

Figure 8: Deformable Body Material Properties

2.4 Isaac Sim

Isaac Sim is the robotics-focused simulation tool developed by NVIDIA. It utilizes the PhysX 5 open-source physics engine along with NVIDIA Raytracing technology to deliver realistic interactions and visuals of objects. Isaac sim specifically gives the user tools to train and test Robotics interactions utilizing standardized robotics communication tools like ROS. This makes Isaac Sim the optimal platform for the creation of digital twins of robots. Isaac sim also contains tools for the creation of synthetic data sets for training and testing of AI systems within a digital environment.

Isaac Sim utilizes the capabilities of the NVIDIA GPU to deliver its high-fidelity visual and physics simulation. This includes realistic real-time ray tracing for lighting. Utilizing the power of the GPU AI acceleration it is also possible to train AI systems with purely synthetic data to such a degree that it can be possible be deployed directly with no physical training data.

Isaac Sim is part of the Omniverse platform which contains tools for 3D model generation, simulation, and testing along with creative tools. The platform utilizes the USD (Universal Scene Description) format developed by Pixar. The standard is made to improve interoperability between different platforms to allow for full scene export and import. This is one of the reasons Omniverse and Isaac Sim is an attractive option as a platform for model verification. The USD format allows you to easily import fully developed CAD models and models from other platforms with little to no problems. [NVIDIA, 2023a]

2.5 Unity3D

Unity 3D is the world's most utilized 3D visualization and game creation tool. It has been in continuous development since 2007; when its first stable release was available. Unity3D relies upon the PhysX Physics engine for its physics simulation making for a robust platform for creation. The platform is built for accessibility and therefore may lack some more complex features that other engines have.

The overarching aim of Unity is to be an accessible platform for hobbyists and professionals alike. Unity allows for simple exporting to many platforms and allows anyone to create software that can be utilized on anything from mobile phones, consoles, phones, and computers all without changing your project workflow. This is due to Unity's exceptional visual editor and scripting language.

Due to the accessibility of Unity for people of every expertise level, the community of the Unity platform is a huge resource. Any problem you may encounter is probably already solved before and is available online through forums and videos. This openness and eagerness to share information allows for simple learning and along with the Unity Asset store you can create quite complex interactions with limited starting knowledge.

The Unity Asset Store is a built-in feature where people from any source can share and sell their products. These products can range from 3D models, sound packs, and control packages to even new additions to the physics engine. These assets can be sold or given away free of charge within the asset store. [Unity, 2023]

2.6 Young's Modulus

Young's modulus, also known as the modulus of elasticity, is the mechanical property that describes the stiffness or elasticity of any given material. Young's modulus is derived from Hooke's law which is described as "a description of the experimental observation that—when strains are small—the strain is very nearly proportional to the stress" [Ashby and Jones, 2019] Young's modulus (E) is expressed as:

$$E = \frac{\sigma}{\epsilon} \tag{1}$$

where:

E is Young's modulus,

σ is the applied stress,

ϵ is the resulting strain.

This is mainly true for isotropic materials and is most used when describing metals. Measurement of anisotropic materials can be done; however, you need to take into account the fact that the values will be different depending on the direction of applied forces. Young's Modulus can be measured in many ways but one of the most known methods is the Three Point Bend test.

This method entails placing a beam-shaped object of material over two equally spaced rigid spacers. Force is then applied to the middle of the object and strain can be measured with the equations found in 9. This method is preferable as it is gentle on the material meaning you can test both brittle and pliable materials, something that can be challenging using other methods such as stretching.

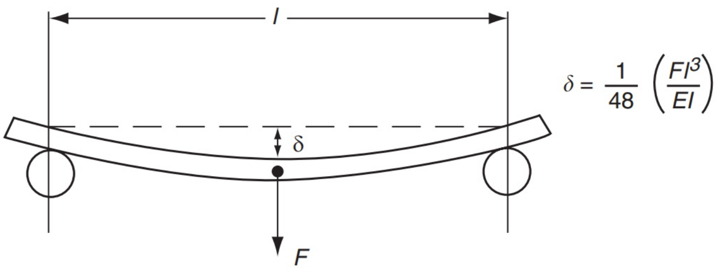

FIGURE 3.6
Three-point bend test.

Figure 9: Illustration of Three Point Bend Test

Source: [Ashby and Jones, 2019, Figure 3.6]

3 Method

3.1 Initial Testing

The goal of the project is to evaluate the efficacy of Isaac Sim as a platform for the training and testing of robotics and physical systems. Therefore development was focused on trying to find the best representation of a dead fish that would also interact well with the built-in systems within the Isaac Sim engine.

Initial testing began with exploring the capabilities of the Deformable Body module within the PhysX Engine. Due to previous work in the summer of 2022, the complexity of the system was taken into consideration. Finding a robust and simple solution that would work well as a general-purpose model for simulation was the most important goal.

A deformable body within the omniverse platform as described earlier is constructed of three parts, one visual, one for collision detection, and one for FEM-simulation of forces. These can be generated automatically by the Omniverse Platform by simply adding a Deformable body module to the visual mesh you wish to simulate. Doing this will automatically generate both a collision mesh and a voxelated Deformable mesh utilizing the “Tet Maker” within the PhysX engine.

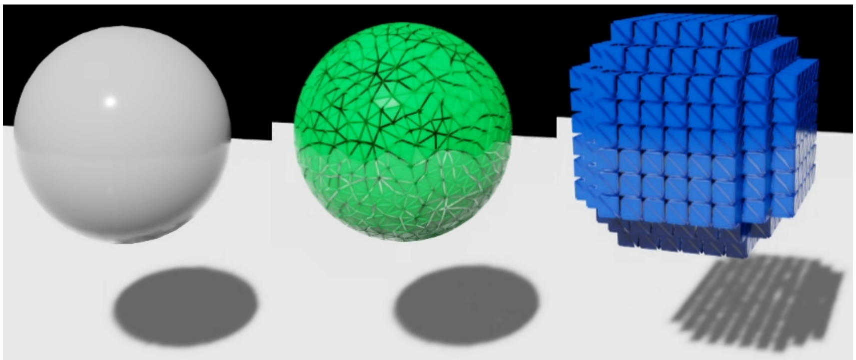

Figure 10: Deformable Body with default settings on a sphere

The deformable body package within Isaac Sim only has the capability of simulating isometric materials, which Fish is not. Fish contains meat, bones, and other internal organs which all behave differently to outside forces. However, for the goal of creating an analog within the digital space that could represent the general behavior of a dead fish, it is acceptable to simplify. For the purpose of simpliciation, we assume the fish is behaving as an isometric material and will therefore adjust values according to this assumption. This will mean that values set within this paper for values of stiffness or other physical properties may not match measurements done on separated pieces of the fish, say for example a fish fillet.

Real time simulation of deformable meshes utilizing FEM is very computationally expensive. Given any force applied must be distributed over X nodes within the simulation mesh within the network. This is why iterative solvers are utilized within the Isaac Sim Package. The computational time and resource utilization of any given simulated mesh within the Isaac Sim program relies mostly upon the iterations of the solver and the complexity of the mesh. As demonstrated in 10, the default setting for the resolution of the mesh is often not sufficient to represent the mesh properly. As you may see, the FEM mesh is only a rough estimation of a sphere at standard settings.

Finding the point of convergence between the desired accuracy of the simulation and simulation time is important. Isaac Sim makes this simple and allows you to re-mesh the object at higher or lower resolutions automatically. A higher resolution will however require a higher solver iteration count, resulting in higher computational expense. Finding the convergence of these values is important to create a simulation that acts realistically while still staying within the bounds of the physical limits of your hardware.

Taking the Sphere in 10 as an example. The FEM representation is crude at the standard resolution of 10. Increasing this resolution to 20 and 30 gives us two meshes that come closer to a true representation of the full curvature of the sphere. Simulating these however with no changes to the solver iterations the mesh seems to become softer for each resolutional increase. By taking the same delta of x + x*y where y is in this case 0,1 and 2 we get solver iteration values that result in closely equal behavior. There is However a performance penalty to this increasing resolution as it causes a decreasing FPS count. Performance was measured after a simple drop before the object can enter "sleep" where it no longer is actively calculating.

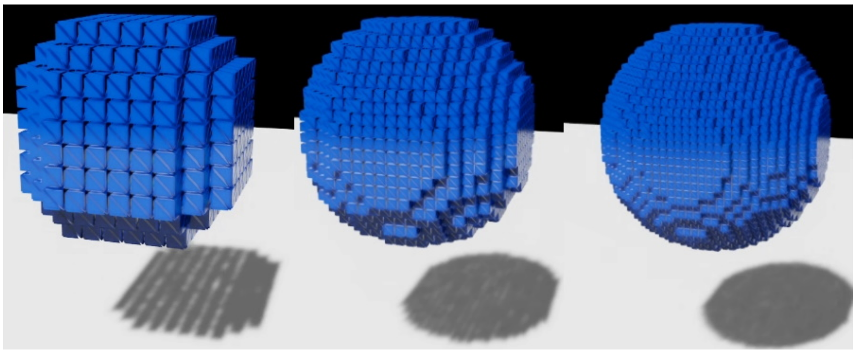

Figure 11: Deformable body of sphere with increasing resolution [10,20,30]

Resolution	Iteration Count	FPS
10	16	42
20	32	28
30	48	16

3.2 FEM material properties

As described before, a FEM simulation cannot be completed without accompanying values describing the material properties. This is addressed by adding a deformable body material to the simulation. This physics material contains the necessary values to represent any type of isotropic material within digital space. The values described are the following: Density, Dynamic friction, Young's Modulus, Poisson's Ratio, Elastic Dampening, and Dampening scale.

Isaac Sim has some standardized values preset for the deformable body material that generally result in quite a good simulation. To my knowledge, there is no analysis done on certain materials that are used as a basis for these numbers so I cannot verify the accuracy of them within use as a basis for the simulation of Fish.

In the material description of the deformable body, the most important metrics for the simulation of a fish are Young's modulus and Poisson's Ratio. As they determine the deformation of the simulated mesh given applied force. Most methods of measuring these values rely upon the material being isotropic. A fish is not an isotropic material, and therefore the act of measuring the elasticity (Young's modulus and Poisson's ratio) is a nontrivial task. Experiments have been done to measure Young's modulus of pieces that make up a fish, including fish meat and bones.

These papers describe fish of differing species from salmon, however, their measured values prove the point that it is as good as impossible to utilize them as a basis for FEM simulation of a whole fish. In Isaac sim, the fish is modeled as an isometric material and the range of measured values from bone to flesh varies so greatly.

Material	Average Modulus (GPa)
Teleost Fish Bone (b)	6.48
Tuna Meat (a)	0.0426

[a] [Ogawa and Hagura, 2011].
[b] [Horton and Summers, 2009].

A possible way of measuring these values for the purpose of simulation within Isaac sim is a three-point bend test. In a three-point bending test, a piece of material is placed over two equally spaced points and allowed to come to a rest. Then a controlled force is applied to the middle of the object and the deformation of the object is measured. Young's modulus can then be calculated by examining the deformation of the material compared to the force applied. This could create values that would be applicable to the simulation of fish within Isaac Sim due to it being measured as an isometric material.

3.3 Model verification

The process of verifying the realism of a digital model is important to ensure that the model is representative. It is however also one of the most complex tasks within the space of simulation. The act of creating a realistic simulation highly depends on the scope of the simulator and what its intended purpose is.

If the intended purpose of a simulation is to accurately predict some chosen variables within a system, the model must be highly controlled and the simulation scope narrow. a general-purpose simulation of a car for example cannot accurately predict every minute detail of the interactions between the car and the environment it is placed within. The complexity becomes too high to simulate within any realistic time frame.

Therefore, when evaluating and developing the fish model described in this paper the project goal must always be kept in mind. Verification of the model has been done in an iterative fashion by asking the simple question "Does this look like a fish". Due to the complexity of the dynamics of a fish and the challenges that arise from representing a complex system with an isometric representation, the best verification tool available is simply general knowledge and your eyes.

Humans are in general great at determining if something looks unnatural, and by utilizing this human judgment verification can be done quite quickly and efficiently. By modifying only one value at a time and simulating iteratively, values for simulation can be homed into a place of relative realism. The values on their own cannot be verified to be accurate but the overall behavior of the system will be verified to the extent that it behaves realistically.

When developing simulations such as this one, exact results are never the goal, the goal of a general-purpose simulation is to simulate the general behavior of a system within a set scope. This means that even if you can replicate every minute detail of the starting conditions exactly within the real and virtual space. The result might still not be exactly equal.

3.4 Conveyor interactions

Conveyors and transport systems are widely used within the fish processing industry, and it is therefore vital that the model presented can interact with these systems naturally. Isaac Sim comes packaged with built-in methods for creating conveyor belts. These built-in systems rely upon a standardized building method using animations as a basis for the visual aspect and relatively simple collision geometry for the conveyor function. This built-in conveyor builder picks from pre-generated assets and positions them according to your instructions allowing you to simply and quickly build realistic-looking and acting conveyor systems

Figure 12: Sample of pre-generated conveyor assets available within Isaac Sim

Source: [NVIDIA, 2023d]

Generally, when you want to move objects within the virtual space you rely upon rigid body kinematics. And for a conveyor, the approach is to give a set speed to any object in contact with the conveyor. This generates a realistic-looking interaction between the conveyor and the object where it is moving at a set speed along the direction speed is applied. However, within the PhysX engine, soft-bodied colliders are not allowed to interact with rigid bodies configured in this manner. You cannot induce a force directly upon a soft body in this manner.

Due to this, some relatively rudimentary tests of fully soft body conveyors were tested. This was done by modeling a conveyor consisting of two rollers, a middle solid piece, and a band that would constitute the deformable body of this conveyor system. It was meant to mimic how a conveyor system is built with rollers and belts. However, the system was extremely computationally expensive and resulted in unnatural movement of the conveyor. I would say this is because the simulation of soft bodies within Isaac Sim is generally not meant for simulating something so delicate as the interactions found in a conveyor belt.

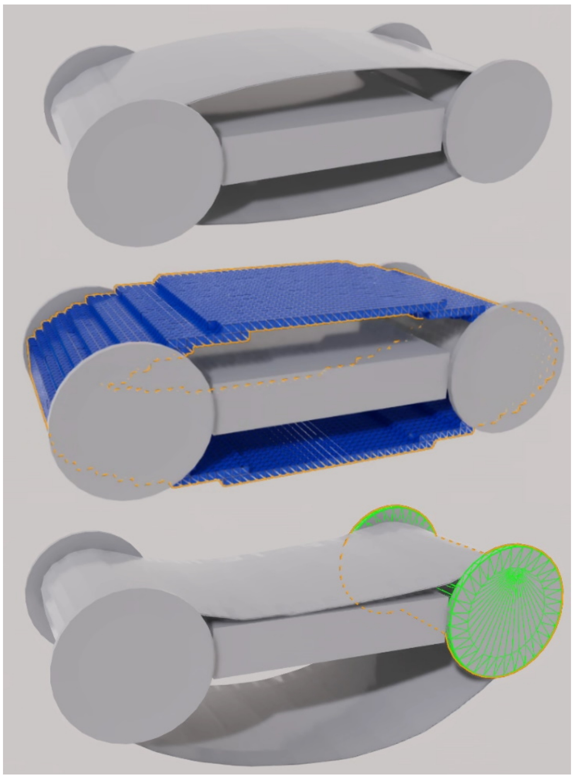

Figure 13: Deformable body conveyor test

These tests were not very thorough, but they showed that the work of chasing this kind of deformable body-based conveyor belt system would require a lot of work to make it functional. The resulting simple simulation ran at a very slow 20 Frames Per Second and would often "bug out" and create unrealistic behavior. This is also working against the goal of making a model that would work well with standardized and simple solutions within the Isaac Sim Platform.

Deformable bodies do however interact well with other rigid bodies that do not enforce this kind of acceleration upon other colliding objects. Within the world of conveyor belts, there are designs that include cleats to transfer objects that may slip up and down slopes. This kind of cleat could be created such that it would act as the main driving force for pushing the deformable body along a chosen axis. This was already utilized within the Unity3D simulator where cleated conveyor belts were used to transport models upwards where they would normally slip down due to low friction.

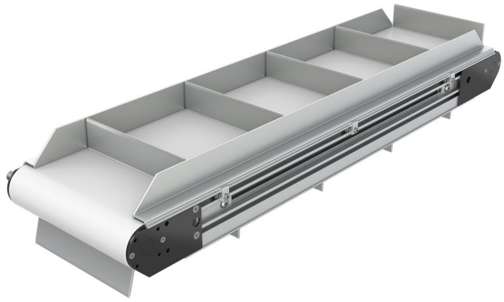

Figure 14: Cleated conveyor belt by ”Dorner Conveyors”

Source: [Dorner Conveyors, 2023]

A possible solution for solving this is to create a custom-generated conveyor system that relies solely on the movement of rigid bodies. An example of this kind of system has been shown by Software Engineer ”Ben Notier” in his blog post about the development of a AI Bin sorting system developed within Isaac Sim. The system relies upon spawning rigid bodies that move along an axis at a set speed and continuously spawning new ones to fill the void when they move out of range. This kind of conveyor system would work well as a system that interacts with a purely deformable body model of a salmon. However, it would require that for every wished simulation of a conveyor belt, a customized approximation of the conveyor belt had to be created. And it would not make it possible to do simple iterative tests for the development of conveyor belt shapes and designs.

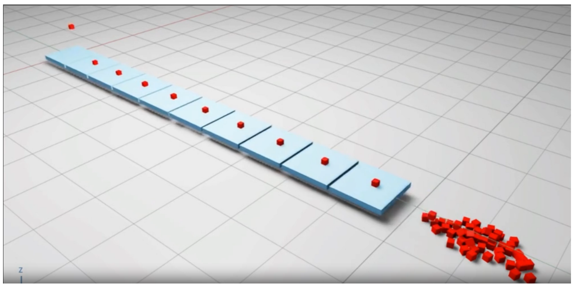

Figure 15: Fully Rigid body conveyor system

Source: [Bjorn Nortier, 2022]

The goal of this work is to evaluate Isaac Sim as a platform for the simulation of fish dynamics. A specialized system like the one described above is something that should be avoided if possible. If this platform is to be utilized as a general simulation platform a better and simpler solution to this problem must be formulated.

3.5 Deformable Body Attachments

Deformable bodies within Isaac Sim have the capability of being attached to both other deformable bodies and other rigid bodies with the attachment system. This system is demonstrated by NVIDIA Researchers to be usable as a means of controlling a deformable body. By for example allowing a deformable body model of a person to be manipulated by internal rigid body "bones". This method of including Rigid body Structures within the deformable body is something that allows for the creation of a generalized fish model that works well with the proposed general solution for conveyor systems. This solution also retains the positives of utilizing a Deformable Body instead of a purely rigid body structure as in Unity3D.

To make a system that combines both a rigid body representation and a deformable body representation of a fish an approximation of the fish utilizing rigid bodies must be created. During the work done for the creation of the Unity3D simulator such a representation has been previously developed. The rigid body representation is made up of 5 separate rigid bodies, each representing different pieces of the body. Separated according to the schematic used when measuring values during the summer project of 2022. 2

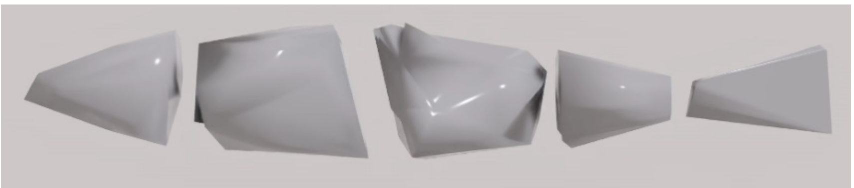

Figure 16: Simplified Separated Rigidbody Approximation of Salmon

This rigid body representation can be inserted within the deformable body and attached using Deformable Body Attachments. By attaching the rigid body to the deformable body utilizing attachments the rigid body approximation will deform with the deformable body. However, when in contact with a conveyor system, the deformable body will not interact with the conveyor as stated earlier, but the rigid body system will. Therefore, the whole model will be able to interact with these conveyor systems.

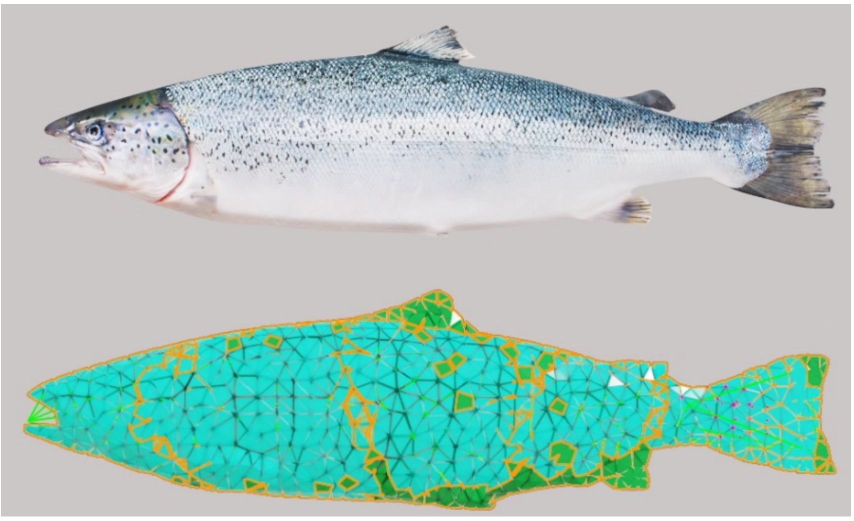

Figure 17: Deformable mesh with internal Rigid Body. Blue shows where attachments are formed

3.6 Inner Rigid body Size Selection

When including Rigid Bodies within the deformable body mesh there are some things to keep in consideration. A rigid body mesh within the simulation will have its own weight and will be simulated in tandem with the deformable mesh. To keep the interactions of the fish as close to a pure deformable body mesh as possible, the limits of these rigid bodies must be selected with care. There are two main variables of the rigid body that determine its interactions, mainly its mass and its size.

The size of the rigid body directly corresponds with the mass of the object. Optimizing the size of the rigid body is important. The internal rigid body needs to be of sufficient size such that it covers the collisions necessary with the chosen conveyor belt meshes while still being as small as possible to limit the disturbance of the deformable body.

An experiment was conducted where analysis of different sizes of internal rigid bodies was tested. Both on a standardized cube and on the fish model itself. The size of the internal rigid body ranged from 0 to 90% with an iteration of 10%. The goal of the test was to find the percentage of the internal rigid body compared to the overall size that would allow for good interaction with a simulated conveyor belt while still behaving as close as possible to a model with no internal rigid body.

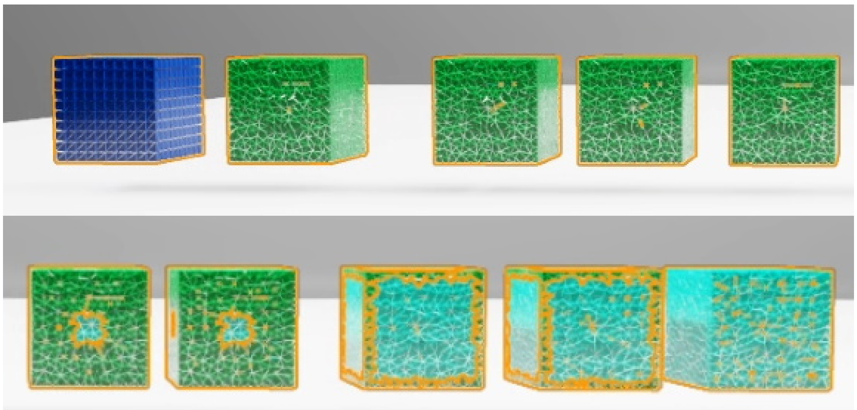

Figure 18: Cubes with increasing inner rigid-body size, increasing from top left ranging from 0.90% inner size

As you may see from 18, for a cube any rigid body inner size lower than 50% has no real attachment points to the deformable body. When exceeding 70% inner size the attachment points starts to cover most of the object. Fish with an internal size below 70% was therefore excluded from testing

A test was conducted on the salmon by a drop test upon a tilted simulated conveyor. The conveyor was not given a lot of thickness so that you could visually inspect the clipping of the deformable body through the object. The salmon were evaluated based on how they move on the conveyor along with how they react to outside forces when dropped off the conveyor. The salmon had an inner rigid body size spanning from 75% to 95% with an interval of 5% due to testing on cubes finding that anything below 70% inner size not creating enough attachment points.

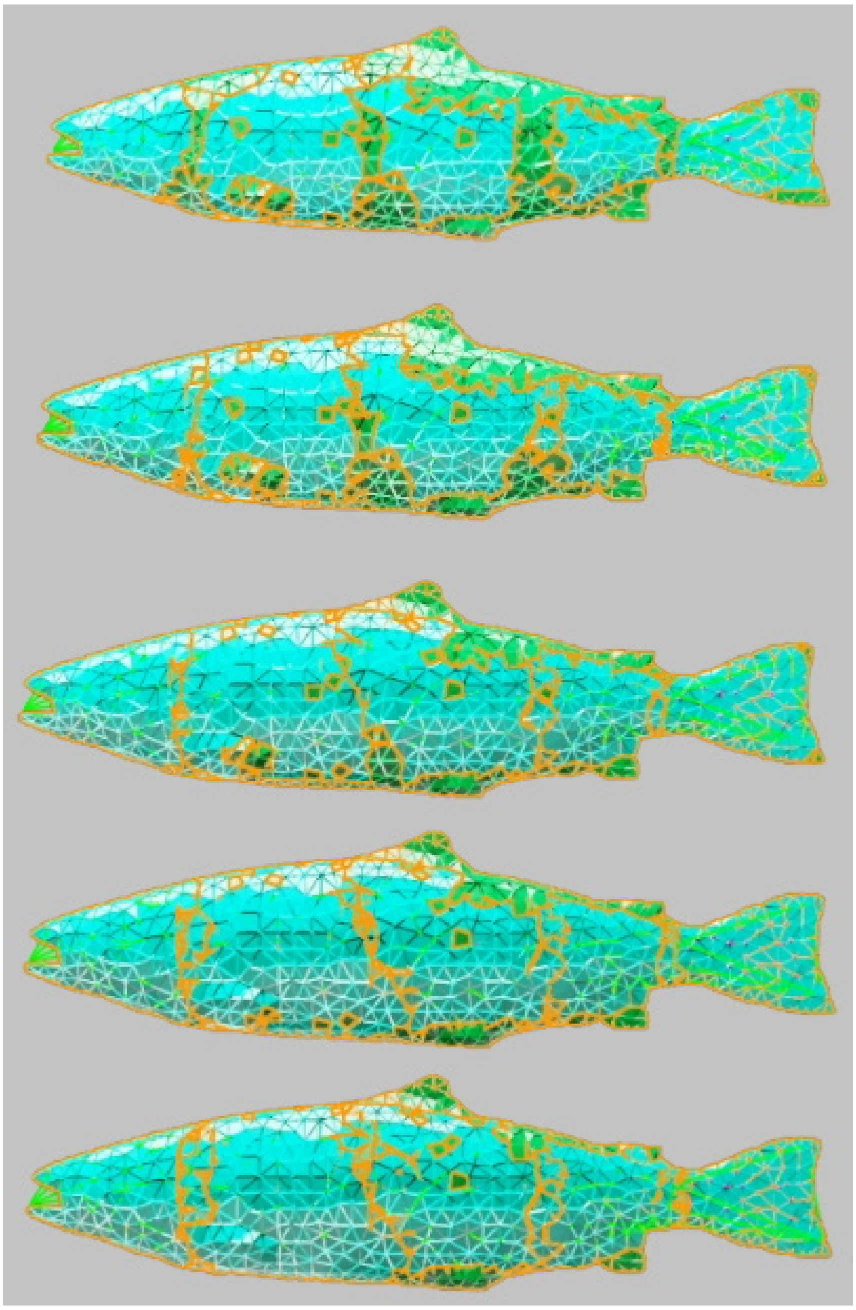

Figure 19: Salmon model with inner rigid body ranging from 70-95%

4 Result and Discussion

4.1 Model

After research and iterative testing, the following model has been deemed to fulfill the criteria given for compatibility and realism within the simulation. This model is not verified with realistic accuracy. It does however show a possible solution of how you would represent a salmon within Isaac Sim and explains the values necessary for realistic simulation.

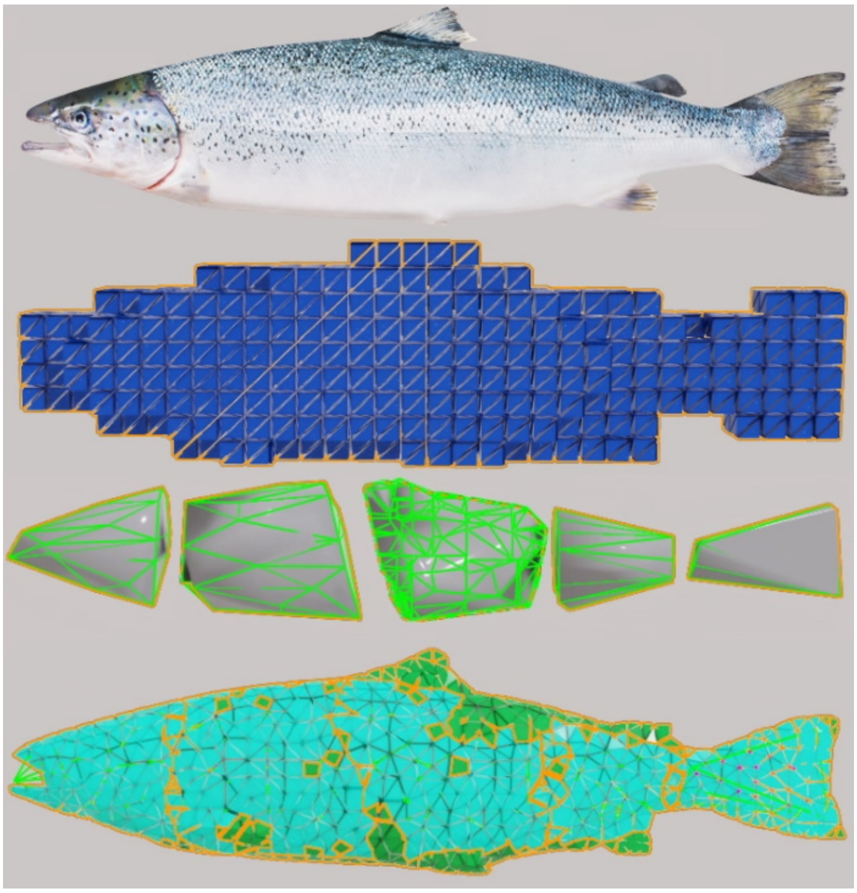

Figure 20: Finalized Model with appropriate values assigned

The model is comprised of a 3D model of a salmon with an accompanying deformable body and collision mesh. The collision mesh and Deformable mesh is auto-generated utilizing built-in tools within Isaac Sim. Values for the model is not verified with any real measurements, it is assigned according to the intuition of what values "look right". The model also contains an inner rigid-body representation with two-way coupling to the deformable body. This is to ensure compatibility and allow for simulation with simulated conveyor belt systems within Isaac Sim.

	Utilized Value	Recommended Range
Deformable Mesh Resolution	32	30-48
Internal Rigid body Size	80%	70-85%
Solver Iterations	32	28-48
Friction Coefficient	0.25	0.05-0.25
Young's Modulus	50MPa	15-50MPa
Poisson's Ratio	0.45	0.35-0.45
Dampening	0.005	0.005

4.2 Deformable Body

The 3d model utilized is gathered from the online model sharing site "Turbosquid.com". The model represents an Atlantic salmon, a type of salmon that is commonly found in Norwegian waters. This model was utilized due to it being one of the most farmed types of fish within Norway and therefore being the type of fish with the most ease of access for measurement of data.

The deformable body was auto-generated utilizing the built-in tools within Isaac Sim at a "Simulation Mesh Resolution" of 32. 16 is standard in the range from 0-64. 32 was selected due to it being one of the lowest possible values that resulted in a simulation mesh that resembled the fish geometry. The trade-off between simulation speed and accuracy was taken into account where the goal was to have a simulation mesh detailed enough to represent the fish geometry while optimizing for simulation speed.

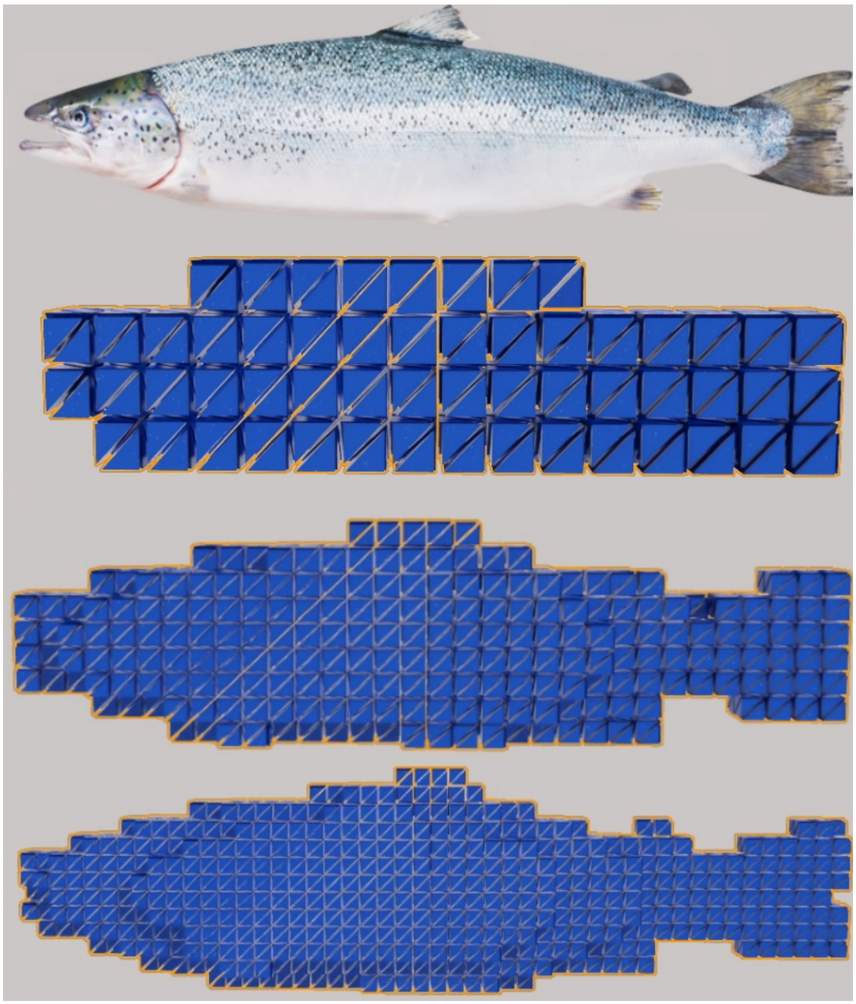

Figure 21: Deformable body resolution of 16,32 and 48

As mentioned explicitly in the documentation for PhysX Soft Bodies, the mesh resolution is directly tied to the iterations of the solver. A higher resolution requires a higher solver positional accuracy, therefore a higher computational load. The "Solver Position Iteration" count ranges from 0-255 and determines how many iterations of calculation done per time-step. The standard value is 16, and due to the effective doubling of the "Simulation Mesh Resolution", the solver iterations is also doubled. Higher values were tested and were deemed to not have a big impact on the simulation, as they were relatively indistinguishable from the 32-iteration simulation. 32 might not be optimal but as 16 was too soft it is suggested to select a value in between 16 and 32 with a mesh resolution of 32.

To accompany the simulation mesh, a physics material is assigned. the material has the following properties:

- Density
- Dynamic Friction
- Young's Modulus
- Poisson's Ratio
- Elasticity Dampening
- Dampening Scale

Density can be excluded as the density of the material is automatically calculated, it can be set manually but it is not something that has been explored. The impact of the other values and suggestions for values will be described here.

During testing of the "Elasticity Dampening Coefficient" it was found that changing this value from the standard of 0.005 only made the fish move less realistically compared to what was expected. The higher the value was moved the more the fish acted as if it was made of cloth. Therefore the value of 0.005 was kept as the working value as changing it in any meaningful way only diminished the realism of the simulation.

Dynamic friction will determine how slow or fast the object will come to a stop when put in motion. The value ranged from 0-inf and is not really based on a standardized measurement method. Within the physics engine, PhysX 0 represents the lack of friction and 1 generally means full friction. Meaning the object stops moving basically immediately as applied forces are removed. Fish are generally quite slippery when handled, Especially in the conditions this model is expected to represent as the fish has been relatively recently removed from the water. Therefore it would be natural to set this lower than the standard value of 0.25. Due to interactions with the necessary internal Rigid body System, it was kept at the standard value to improve stability.

The Poisson's Ratio was also kept at a standard value due to it not having a big impact on the simulation during testing. The standard value is set at 0.45. It is also expected that the value would be quite high given experience handling fish. As this value describes the volume preservation during stress and organic materials as meats generally keep their volume quite well. A measured value might be lower than this but it was not deemed necessary to research given its relatively low impact on the simulation.

Young's Modulus is the most important value for controlling the dynamics of the model as it directly controls the stiffness of the simulated mesh. The modulus is measured in Pascals and the normal range of values from materials is usually between 1000GPa at the high end for diamonds and 0.001GPa (1MPa) for soft rubbers and other soft polymers.) [Ashby and Jones, 2019, Chapter 3.6]

Isaac Sim has a standard value of 50MPa, a value that is too high for a fully deformable body simulation of a fish. When testing only the deformable body by itself a value between 5-15MPa is recommended. When introducing the inner rigid body structure, a higher value is necessary due to added weight and two-way coupling with the internal rigid body. Simulation of a combined system with a too-low value for Young's Modulus created some unintended results. Therefore when simulating a combined system values above 30MPa are recommended and for simplicity it was therefore kept at a standard value of 50MPa.

4.3 Rigid-body structure

The internal rigid bodies were created directly by slicing the model into 5 pieces and completing their shell within Blender. The model was sliced according to the measurement sheet utilized during the summer of 2022 to match the method of data collection utilized within that project. The 3d models were utilized within the Unity build as the only means of collision detection. The internal rigid bodies are therefore a close match geometrically to the accompanying visual mesh. The bodies are evenly positioned along the fish with space in between.

The bodies are attached to the deformable body by utilizing Isaac Sim Deformable Body Attachments. These attachments create a two-way coupling between the vertices and faces of the internal rigid body and the outside Deformable Body Mesh. The positioning of attachments is shown in 20, attachment points colored blue.

The size of the internal rigid body system was considered carefully. A test was done on both cubes and fish Deformable models with differing sizes of internal rigid bodies. This was done to try to find the optimal size of the internal rigid body where the model would interact well with the conveyor belt system, along with being as close as possible in dynamics to the model with no internal rigid body structure. The internal rigid body will always have an impact upon the model but attempting to minimize it is the goal.

The tests performed showed that any internal structure of size lower than 65% created unrealistic interactions with the conveyor belt systems as the internal structure was too small and allowed for excessive clipping. It was also noted that any value above 85% or so created problems with the dynamics as the weight of the deformable body was too high causing unrealistic behavior of the deformable body. Values between 70-85% are recommended as they closely match the outside deformable mesh while not causing any unwanted interactions or "bugs".

4.4 Stability Considerations

One of the main considerations taken throughout the entire workflow of this project was maintaining stability. Experience gained from the previous Unity3D project advises simplifying the model as much as possible. One of the big problems that was faced when trying to utilize the simulator was the stability of the model. The source of this instability was never properly researched, but it is expected to be caused due to excessive restraining of the model.

Within the Unity3D project, the exact degrees of freedom of the joint between each rigid body were possible to assign. These values were therefore measured and applied to the model where applicable. Weight was also assigned to the model to each individual piece of the rigid body internal module. This created instability in the simulation where collision detection or forces caused joints and colliders to move outside the allowed regions. The physics engine naturally tries to compensate for this compensation often caused unnatural acceleration and movement.

With the usage of an internal rigid body system within the Deformable Body model used within this paper, it was also possible to assign the same limitations of movement with joints in between the rigid bodies. When this was attempted the same kinds of errors occurred where unwanted and unrealistic movement caused the models to shoot off and out of the scene. This behavior can also happen without the joints but was hugely exacerbated with joints implemented.

The consideration of stability also applies to the internal rigid body structure. Including this rigid body structure does improve the compatibility of the model immensely, allowing it to work with simple conveyor belt models. This allows the model to be utilized to analyze different layouts and conveyor belt models. This structure does add some instability and weight that would optimally not be necessary. It is recommended to evaluate well before utilizing this model within a test. If a test is possible to be done utilizing a pure deformable body model this is highly recommended.

5 Conclusion

Utilizing the many powerful tools within Isaac Sim for the creation of realistic Deformable Body Based models, a robust model has been created. The model is stable and is able to simulate and test conveyor systems and layouts of differing geometry. The model is simulated utilizing the Finite Element Method (FEM) along with an internal rigid-body approximation for interaction with systems that do not interact with the FEM mesh. The values for the FEM simulation have not been verified with any physical tests, but lays the foundation for generalized simulation that "looks like a fish". The stability of this system compared to previous Unity3D-based simulation methods is vastly improved. Isaac sim as a platform is more powerful than Unity3D and allows for a wider array of tests to be performed with less specialized tools or training.

6 Future Work

6.1 Measurement of values

During development measuring values from a physical test was a goal. Due to the complex nature of the system, a proposed method was not found until relatively late in the research process. Due to this, the test was never done. The basis for how this test would be done has been put forward in this paper and the act of performing this test would be a good addition to verify the realism of the model. It is however not extremely important as the end goal of this model is to be realistic enough that you could use it in a macro sense and test possible layouts of processing systems. If this model is to be used for example to train AI systems, the model needs to be verified to a different standard.

6.2 Review of Internal Rigid Body

To allow for the simulation of differing geometry of conveyor systems within Isaac Sim it was necessary to include a rigid body-based approximation within the model. This inclusion does however create unwanted behavior since the rigid bodies also are simulated along with the Finite Element Mesh. This means that they increase the inertia of the system and allow for possible buggy behavior. Researching the possibilities of Isaac sim to perhaps simulate conveyor systems in a different way to allow for a pure Deformable Body based model would be beneficial

6.3 Model Performance Optimization

During the Development of this model Optimization of performance was not the most important consideration. The goal was to create a model that would be stable and interact well with the internal systems of Isaac Sim. Therefore going through every value that impacts performance and optimizing them further will allow for bigger-scaled tests to be performed. This optimization will require a model of a physical system along with mirroring Physical tests. This would be necessary to ensure that the model remains accurate to Physical tests.

References

[Ashby and Jones, 2019] Ashby, M. F. and Jones, D. R. (2019). *Materials: Engineering, Science, Processing and Design*. Butterworth-Heinemann. Accessed on 15.06.2023.

[Bjorn Nortier, 2022] Bjorn Nortier (2022). AI Picker Pt1 - A novel industrial robotic application. Accessed on 15th June 2023.

[Dorner Conveyors, 2023] Dorner Conveyors (2023). Cleated Belt Conveyors - Dorner Conveyors. Accessed on 15th June 2023.

[Horton and Summers, 2009] Horton, J. M. and Summers, A. P. (2009). The material properties of acellular bone in a teleost fish. *Journal of Experimental Biology*, 212(9):1413–1420.

[Liu and Quek, 2013] Liu, G.-R. and Quek, S. S. (2013). *The finite element method: a practical course*. Butterworth-Heinemann.

[Nagakura et al., 2001] Nagakura, S., Noguchi, S., Kaneda, K., Yamashita, H., and Cingoski, V. (2001). Automatic quadrilateral mesh generation for fem using dynamic bubble system. *IEEE transactions on magnetics*, 37(5):3522–3525.

[NVIDIA, 2023a] NVIDIA (2023a). Isaac Sim - NVIDIA Developer. https://developer.nvidia.com/isaac-sim. Accessed on 15.06.2023.

[NVIDIA, 2023b] NVIDIA (2023b). NVIDIA Omniverse Documentation. https://docs.omniverse.nvidia.com/app_create/prod_extensions/ext_physics/rigid-bodies.html. Accessed on 15.06.2023.

[NVIDIA, 2023c] NVIDIA (2023c). NVIDIA Omniverse Documentation. https://docs.omniverse.nvidia.com/prod_extensions/prod_extensions/ext_physics/deformable-bodies.html. Accessed on 15.06.2023.

[NVIDIA, 2023d] NVIDIA (2023d). Omni.ISAAC Conveyor - NVIDIA Omniverse Documentation. Accessed on 15th June 2023.

[Ogawa and Hagura, 2011] Ogawa, Y. and Hagura, Y. (2011). Measurement of young's modulus and poisson's ratio of tuna fish. *Transactions of the Japan Society of Refrigerating and Air Conditioning Engineers*, 9(3):283–290.

[The Research Council of Norway, 2023] The Research Council of Norway (2023). FISH 4.0- Industrialization of the marine value chain. https://prosjektbanken.forskningsradet.no/en/project/FORISS/331829. Accessed on 15.06.2023.

[Unity, 2023] Unity (2023). Unity. https://unity.com/. Accessed on 15.06.2023.

NTNU

Norwegian University of
Science and Technology