3D ANIMATION
3D Dynamics (Practical)

Diploma in Multimedia and Animation (DMA)
3D Animation
Block – V: 3D Dynamics (Practical)

Odisha State Open University
3D Animation

This course has been developed with the support of the Commonwealth of Learning (COL). COL is an intergovernmental organisation created by Commonwealth Heads of Government to promote the development and sharing of open learning and distance education knowledge, resources and technologies.

Odisha State Open University, Sambalpur (OSOU) is the first Open and Distance learning institution in the State of Odisha, where students can pursue their studies through Open and Distance Learning (ODL) methodologies. Degrees, Diplomas, or Certificates awarded by OSOU are treated as equivalent to the degrees, diplomas, or certificates awarded by other national universities in India by the University Grants Commission.

© 2018 by the Commonwealth of Learning and Odisha State Open University. Except where otherwise noted, 3D Animation is made available under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) License: https://creativecommons.org/licenses/by-sa/4.0/legalcode

For the avoidance of doubt, by applying this license the Commonwealth of Learning does not waive any privileges or immunities from claims that it may be entitled to assert, nor does the Commonwealth of Learning submit itself to the jurisdiction, courts, legal processes or laws of any jurisdiction. The ideas and opinions expressed in this publication are those of the author/s; they are not necessarily those of Commonwealth of Learning and do not commit the organisation.
Acknowledgements

The Odisha State Open University and COL, Canada wishes to thank those Resource Persons below for their contribution to this DMA-04:

**Concept / Advisor**

Dr. Srikant Mohapatra  
Vice- Chancellor  
Odisha State Open University, Sambalpur

**Course Writer**

A.C.Balaji  
Head, Department of Animation  
Manipal University

**Course Editor**

S.Anuradha  
Bangalore based Freelancer.

**Video Production**

R. Mohana Sundaram  
Creative Director  
Jai Ram Institute of Visual Academy, Khurda, Odisha  
Guest Faculty, National Institute of Fashion Technology (NIFT), Bhubaneswar

**Published by :**

Dr. Jayanta Kar Sharma  
Registrar on behalf of Odisha State Open University, Sambalpur

**Contribution of following staff members of Odisha State Open University is acknowledged:**

- Sambit Mishra
- Debidatta Behera
- Prashansha Das
- Radhakanta Suna
- Abhinandan Tripathy

OSOU and COL acknowledge the support extended by Prof. Madhu Parhar, STRIDE, IGNOU, New Delhi in conducting several workshops in the process of preparation of course material for DMA.
# Contents

## Course Overview

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome to 3D Dynamics</td>
<td>3</td>
</tr>
<tr>
<td>Introduction to Dynamics</td>
<td>3</td>
</tr>
<tr>
<td>Rigid Body, Soft body and Constraints</td>
<td>4</td>
</tr>
<tr>
<td>Cloth Simulations</td>
<td>4</td>
</tr>
<tr>
<td>Fluid Simulation</td>
<td>5</td>
</tr>
<tr>
<td>Course outcomes</td>
<td>6</td>
</tr>
<tr>
<td>Timeframe</td>
<td>6</td>
</tr>
<tr>
<td>Study skills</td>
<td>6</td>
</tr>
<tr>
<td>Need help?</td>
<td>7</td>
</tr>
<tr>
<td>Assignments</td>
<td>7</td>
</tr>
<tr>
<td>Assessments</td>
<td>7</td>
</tr>
<tr>
<td>Video Resources</td>
<td>7</td>
</tr>
</tbody>
</table>

## Getting around this Course material

<table>
<thead>
<tr>
<th>Margin icons</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

## Unit-1

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Dynamics</td>
<td>9</td>
</tr>
<tr>
<td>Introduction</td>
<td>9</td>
</tr>
<tr>
<td>Outcomes</td>
<td>9</td>
</tr>
<tr>
<td>Terminology</td>
<td>10</td>
</tr>
<tr>
<td>Baking Physics Simulations</td>
<td>10</td>
</tr>
<tr>
<td>Compression</td>
<td>11</td>
</tr>
<tr>
<td>External</td>
<td>11</td>
</tr>
<tr>
<td>Multiple Caches</td>
<td>13</td>
</tr>
<tr>
<td>Common Field Settings</td>
<td>15</td>
</tr>
<tr>
<td>Unit summary</td>
<td>37</td>
</tr>
<tr>
<td>Assessment</td>
<td>38</td>
</tr>
<tr>
<td>Resources</td>
<td>39</td>
</tr>
</tbody>
</table>

## Unit 2

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to Rigid Body, Soft Body and Constraints</td>
<td>41</td>
</tr>
<tr>
<td>Introduction</td>
<td>41</td>
</tr>
<tr>
<td>Outcomes</td>
<td>41</td>
</tr>
<tr>
<td>Terminology</td>
<td>41</td>
</tr>
<tr>
<td>Terminology</td>
<td>41</td>
</tr>
<tr>
<td>Rigid Body</td>
<td>42</td>
</tr>
<tr>
<td>Soft Body</td>
<td>53</td>
</tr>
</tbody>
</table>
Rigid Body Constraint Types ................................................................. 60
Animation .............................................................................................. 67
Simulation Stability ............................................................................. 67
Combining Rigid Bodies with Other Simulations .................................. 68
Scaling Rigid Bodies ........................................................................... 68
Unit summary ....................................................................................... 69
Assignment .......................................................................................... 69
Assessment ........................................................................................... 69
Resources ............................................................................................. 70

Unit 3 ........................................................................................................ 71
Welcome to Cloth Simulations ............................................................... 71
Introduction ........................................................................................... 71
Outcomes ............................................................................................... 71
Cloth Simulations ................................................................................ 72
Collisions ............................................................................................... 78
Cloth Cache ........................................................................................... 83
Troubleshooting .................................................................................... 85
Using Simulation to Shape/Sculpt a Mesh ............................................. 86
Unit summary ........................................................................................ 89
Assignment ............................................................................................ 89
Assessment ............................................................................................ 89

Unit 4 ........................................................................................................ 91
Welcome to Fluid Simulation ................................................................. 91
Introduction ........................................................................................... 91
Outcomes ............................................................................................... 91
Terminology ........................................................................................... 91
How Materials Works ........................................................................... 92
Fluid World ............................................................................................ 98
Fluid Boundary ..................................................................................... 101
Fluid Particles ....................................................................................... 102
Fluid Object ........................................................................................... 104
Fluid Obstacle ....................................................................................... 105
Fluid Inflow / Outflow .......................................................................... 106
Fluid Particle ........................................................................................ 108
Fluid Control ........................................................................................ 109
Baking .................................................................................................... 112
Fluid Appendix ..................................................................................... 114
Unit summary ........................................................................................ 117
Assignment ............................................................................................ 117
Assessment ............................................................................................ 118
Resources ............................................................................................... 118
Course Overview

Welcome to 3D Dynamics

All real world physical phenomena are the branches of Physics. The software Blender offers a variety for different physics, for example you can use Blender to simulate to Smoke, Rain, Dust, Cloth, Water, Jelly Etc.,

Rigid Body is an idealization of a Solid Body in which deformation is neglected. In other words, the distance between any two given points of a Rigid Body remains constant in time regardless of external forces exerted on it.

In Cloth simulation you can also specify what Cloth is made of, and what a solid or collision object is. This is where you define Cloth and collision objects, assign properties, and execute the Simulation.

Fluid Simulation refers to Computer Graphics techniques for generating realistic animations of fluids such as liquid, water and smoke. You would come across Fluid Simulation that are typically focused on emulating the qualitative visual behavior of a fluid, with less emphasis placed on rigorously correct physical results, although they often still rely on approximate solutions that govern real fluid physics.

Introduction to Dynamics

All real world physical phenomena are the branches of Physics. The software Blender offers a variety for different physics, for example you can use Blender to simulate to Smoke, Rain, Dust, Cloth, Water, Jelly Etc., In this Unit, you will learn what is Physical Simulation to 3D objects and will practice real world physical simulation. You will also learn what gravity is and how to demonstrate the utility of Force Fields; Use Multiple field to create the required effects and Apply Gravity in 3D Simulation using Rigid Body and Soft Body.
Rigid Body, Soft body and Constraints

**Rigid Body** is an idealization of a **Solid Body** in which deformation is neglected. In other words, the distance between any two given points of a **Rigid Body** remains constant in time regardless of external forces exerted on it, and **Soft-body dynamics** that it is a field of computer graphics that focuses on visually realistic physical simulations of the motion and properties of deformable objects (or soft bodies). In this Unit, you will learn **what is Rigid Body and Soft Body** and how to Create Passive Body and Active Body; Demonstrate the utility of Rigid Body; Use Multiple constraints; Practice Soft Body animation; and to Use constraints to control dynamic animation of the Active and Passive bodies in real time.

Cloth Simulations

In this course, you will be introduced to **Cloth Simulations** and to describe which objects will be part of the Simulation and which are not. Once you have done this, you can define what the **objects are made of**. You can also specify what Cloth is made of, and what a solid or collision object is. This is where you define Cloth and collision objects, assign properties, and execute the Simulation. Other controls included is creating constraints, interactively dragging the cloth, and erasing parts of the Simulation. In this Unit, you will learn **what is Cloth Simulation** and how to Create different types of Cloth Simulation like cape, flag, curtain, banners and fabrics, its use with the help of 3D assets to effectively work on 3D scene; Make clothing to your 3D Assets; Use Cloth as extra effects in your 3D Scene; Use the Simulation to create effects to 3D Characters, 3D Assets and 3D Environment; **Bake and save Simulation** on moving objects using Cache and to Create Modifier Stacks and smooth Cloth Simulation.
Fluid Simulation

**Fluid Simulation** refers to **Computer Graphics** techniques for generating realistic animations of fluids such as liquid, water and smoke. You would come across Fluid Simulation that are typically focused on emulating the qualitative visual behavior of a fluid, with less emphasis placed on rigorously correct physical results, although they often still rely on approximate solutions that govern real fluid physics. Fluid Simulation can be performed with different levels of complexity, ranging from **time-consuming** and high-quality animations for films or visual effects, to **simple and fast animations** for real-time animations like computer games. In this Unit, you will learn what is Fluid Simulation and how to Define and use the Fluid Simulation in your 3D Scene, using the **global simulation parameter**; Create simulation for water, liquid, oil, flood, ocean etc.; Control Fluid Simulation using the physical properties of those elements; Interact with the collusion or obstacle objects; Describe to control the flow of the fluid; Demonstrate the use of physical properties of the fluid; Use **bake simulation** to save performance and to Reuse Existing simulation.

This video will provide a brief overview of this course.

<table>
<thead>
<tr>
<th>Topic</th>
<th>YouTube link</th>
<th>QR Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video 1 –Dynamics</td>
<td><a href="https://youtu.be/yQEgIWLqfYw">https://youtu.be/yQEgIWLqfYw</a></td>
<td></td>
</tr>
<tr>
<td>Video 2 –Cloth Simulation</td>
<td><a href="https://youtu.be/DCdMIYxdsg">https://youtu.be/DCdMIYxdsg</a></td>
<td></td>
</tr>
<tr>
<td>Video 3 –Particle Instance</td>
<td><a href="https://youtu.be/KOLSsvsvz0I">https://youtu.be/KOLSsvsvz0I</a></td>
<td></td>
</tr>
</tbody>
</table>
Course Overview

Course outcomes

Upon completion of 3D Dynamics you will be able to:

- Apply physical simulation to 3D Objects in your scene
- Demonstrate the utility of Force Fields
- Create Passive Body and Active Body
- Demonstrate the utility of Rigid Body
- Demonstrate the utility of Cloth Simulation
- Use Cloth Simulation to create flags and banners
- Define and Use Fluid Simulation in your 3D project for creating extra effects
- Describe to control the flow of the fluid

Timeframe

This course will be completed within “2” classes.

This course is of “1” credits.

2 Hour of study time is required for this unit.

Study skills

This is a totally practical oriented course.

Hence, you should have access to personal computer or personal laptop for better understanding of this unit.

Each and every options are explained step by step in the course material.

Apart from this course material, the learner has to adopt the tendency of learning from multiple sources i.e.,

- Internet tutorials
- Video tutorials on YouTube
- Collaboration with people working in the industry etc.

Only classroom study will not make you a professional. You have to be active to grab the opportunity of learning wherever you get a chance.
Need help?

In case of any help needed you can browse the internet sites like youtube.com for video tutorials about the subject.

Apart from that, you can contact the writer of this course material at balaji.ac@manipal.edu

Assignments

There will be some assignments at the end of each unit. These assignments are mostly practical based and should be submitted in CD or DVD. Theoretical assignments are to be submitted neatly written on A4 size sheet.

All assignments will be submitted to Regional centre of Odisha State Open University or as directed by Co-ordinator.

All assignment should be unit wise on separate CD/DVDs clearly mentioning course title and unit on Top. Theoretical Assignment will be neatly filed or spiral bind with cover clearly mentioning necessary information of course, student detain on top.

Assessments

There will be few assessment questions for each unit.

All practical assessment will be submitted to OSOU.

Assessment will take place once at the end of each unit.

Learner will be allowed to complete the assessment within stipulated time frame given by the university.

Video Resources

This study material comes with additional online resources in the form of videos. As videos puts in human element to e-learning at the same time demonstrating the concepts visually also improves the overall learning experience.

You can download any QR code reader from Google Play to view the videos embedded in the course or type the URL on a web browser.
Getting around this Course material

Margin icons

While working through this Course material you will notice the frequent use of margin icons. These icons serve to “signpost” a particular piece of text, a new task or change in activity; they have been included to help you to find your way around this Course material.

A complete icon set is shown below. We suggest that you familiarize yourself with the icons and their meaning before starting your study.

<table>
<thead>
<tr>
<th>Activity</th>
<th>Assessment</th>
<th>Assignment</th>
<th>Case study</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discussion</td>
<td>Group activity</td>
<td>Help</td>
<td>Note it!</td>
</tr>
<tr>
<td>Outcomes</td>
<td>Reading</td>
<td>Reflection</td>
<td>Study skills</td>
</tr>
<tr>
<td>Summary</td>
<td>Terminology</td>
<td>Time</td>
<td>Tip</td>
</tr>
</tbody>
</table>
Unit-1

Introduction to Dynamics

Introduction

All real world physical phenomena are the branches of Physics. The software Blender offers a variety for different physics. For example you can use Blender to simulate to Smoke, Rain, Dust, Cloth, Water, Jelly etc.,

Particle Systems can be used to simulate many things: hair, grass, smoke, flocks. Hair is a subset of the particle system, and can be used for strand-like objects, such as hair, fur, grass, quills, etc.

Soft Bodies are useful for everything that tends to bend, deform, in reaction to forces like gravity or wind, or when colliding with other objects. It can be used for skin, rubber, and even clothes, even though there is separate Cloth Simulation specific for cloth-like objects. Rigid Bodies can simulate dynamic objects that are fairly rigid.

Fluids, which include liquids and gases, can be simulated, including Smoke. Force Fields can modify the behavior of simulations.

In this Unit, you will learn the physical simulation to 3D objects and practice real world physical simulation.

Outcomes

Upon completion of this unit you will be able to:

- Apply physical simulation to 3D Objects in your scene
- Demonstrate the utility of Force Fields
- Use Multiple field to create the required effects
- Practice real world physical simulation
Terminology

Baking: Baking, in general, is the act of pre-computing something, in order to speed up some other process later down the line.

Cache: A collection of items of the same type stored in a hidden or inaccessible place.

Gravity: Gravity is a global setting that is applied the same to all physics systems in a scene, which can be found in the scene tab. This value is generally fine left at its default value, at -9.810 in the Z-Axis, which is the force of gravity in the real world. Lowering this value would simulate a lower or higher force of gravity.

Tip

The gravity value per physics system can be scaled down in the Field Weights tab.

Baking Physics Simulations

Baking refers to the act of storing or caching the results of a calculation.

It is generally recommended to bake your physics simulations before rendering. Aside from no longer needing to go through the time-consuming process of simulating again, baking can help prevent potential glitches and ensure that the outcome of the simulation remains exactly the same every time.

Tip

Most physics simulators in Blender use a similar system, however, not all have exactly the same settings available. All the settings are covered here, however, individual physics types may not provide all these options.
Compression

Compression level for cache files. Some physics caches can be very large (such as smoke). Blender can compress these caches in order to save space.

- **None**
  Do not compress the cache.

- **Light**
  Compression optimizes speed of compressing/decompressing operations over file size.

- **Heavy**
  Compression will result in smaller cache file more than Light, however, requires more CPU time to compress/decompress.

External

Read and write the cache to disk using a user-specified file path.

- **Index Number**
  This number specifies which cache should be used when the specified cache directory contains multiple caches. 0 refers to the top-most cache, 1 to the second from the top, 2 to the third, and so on.

- **Use Lib Path**
  Share the disk cache when the physics object is linked into another blend-file.

  When this option is enabled, linked versions of the object will reference the same disk cache. When disabled, linked versions of the object will use independent caches.

- **Start**
  Frame on which to start the simulation.

- **End**
  Frame on which to stop the simulation.

- **Cache Step**
  Interval for storing simulation data.
Some physics systems (such as particles) allow for positions to be stored only on every nth frame, letting the positions for in-between frames be interpolated. Using a cache step greater than one will result in a smaller cache, however, the result may differ from the original simulation.

- **Bake**
  Start baking. Blender will become unresponsive during most baking operations. The cursor will display as a number representing the bakes’ progress.

- **Free Bake**
  Mark the baked cache as temporary. The data will still exist, however, will be removed with the next object modification and frame change. This button is only available when the physics system has been baked.

- **Calculate to Frame**
  Bake only up to the current frame. Limited by End frame set in the cache settings.

- **Current Cache to Bake**
  Store any temporarily cached simulation data as a bake. Note that playing the animation will try to simulate any visible physics simulations. Depending on the physics type, this data may be temporarily cached. Normally such temporary caches are cleared when an object or setting is modified, however, converting it to a bake will “save” it.

- **Bake All Dynamics**
  Bake all physics systems in the scene, even those of different types. This is useful for baking complex setups involving interactions between different physics types.

- **Free All Bakes**
  Free bakes of all physics systems in the scene, even those of different types.

- **Update All to Frame**
  Bake all physics systems in the scene to the current frame.
Multiple Caches

Blender allows for storing and managing multiple caches at once for the same physics object.

![Smoke Cache](image)

**Title**- Img. 1. Two different caches stored simultaneously.

**Attribution**-

Source-


Caches can be added and removed with the **Plus** and **Minus** buttons. Renaming a cache can be done by either **double clicking** or pressing **Ctrl-LMB** on the desired cache.

**Force Fields**

Force Fields offer a way to add extra movement to dynamic systems. **Particles, Soft Bodies, Rigid Bodies and Cloth objects** can all be affected by forces fields. Force Fields automatically affect everything. To remove a simulation or particle system from their influence, simply turn down the influence of that type of Force Field in its Field Weights panel.

- All types of objects and particles can generate fields, however, only curve object can bear **Curve Guides** fields.
- Force Fields can also be generated from particles.
- The objects need to share at least one common layer to have effect.

You may limit the effect on particles to a group of objects.
Creating a Force Field

Force field types

Title-Img. 1. 2Creating a Force Field.
Attribution-
Source-
Link-  

Reference
- **Mode**: Object Mode  
- **Panel**: Physics ▶ Fields

To create a single Force Field,

- Step 1: Select **Add ▶ Force Field** and  
- Step 2: Select the desired force field.

This method creates an Empty with the force field attached.

To create a field from an existing object,

- Select the object and change to the **Physics tab**.  
- Select the field type in the **Fields** menu.

The fields have many options in common; these common options are explained for the **Spherical field**.
After changing the Fields panel or deflection Collision panel settings, you must recalculate the particle, Soft Body or cloth system by Free Cache, this is not done automatically. You can clear the cache for all selected objects with Ctrl-B > Free cache selected.

Particles react to all kind of Force Fields, Soft Bodies only to Spherical, Wind, Vortex (they react on Harmonic fields However, not in a useful way).

Common Field Settings

Most Fields have the same settings, even though they act very differently. Settings unique to a field type are described below.

Curve Guide and Texture Fields have very different options.

- **Shape**
  The field is either a *Point*, with omni-directional influence, or a *Plane*, constant in the XY-plane, changes only in Z direction.

- **Strength**
  The strength of the field effect can be positive or negative to change the direction that the force operates in. A force field’s strength is scaled with the force object’s scale, allowing you to scale up and down scene, keeping the same effects.

- **Flow**
  Convert effector force into air flow velocity.

- **Noise**
  Adds noise to the strength of the force.

- **Seed**
  Changes the seed of the random noise.

- **Effect Point**
  You can toggle the field’s effect on particle *Location* and *Rotation*

- **Collision Absorption**
  Force gets absorbed by collision objects.
Falloff

Here you can specify the shape of the force field (if the Fall-off Power is greater than 0).

- **Sphere**
  Falloff is uniform in all directions, as in a sphere.

- **Tube**
  Fall off results in a tube-shaped force field. The Field’s Radial falloff can be adjusted, as well as the Minimum and Maximum distances of the field.

- **Cone**
  Fall off results in a cone shaped force field. Additional options are the same as those of Tube options.

- **Z Direction**
  Fall-off can be set to apply only in the direction of the positive Z Axis, negative Z Axis, or both.

- **Power (Power)**
  How the power of the force field changes with the distance from the force field. If \( r \) is the distance from the center of the object, the force changes with \( 1/r^{\text{power}} \). A Fall-off of 2 changes the force field with \( 1/r^2 \), which is the falloff of gravitational pull.

- **Max Distance**
  This makes the force field to take effect within a specified maximum radius (shown by an additional circle around the object).

- **Min Distance**
  The distance from the object center, up to where the force field is effective with full strength. If you have a Fall-off of 0 this parameter does nothing, because the field is effective with full strength up to Max Distance (or the infinity).
The **Force** field is the simplest of the fields. It gives a constant force towards (positive strength) or away from (negative strength) the object’s center. Newtonian particles are attracted to a field with negative strength, and are blown away from a field with positive strength.
For Boids Particles, a field with positive strength can be used as a **Goal**, a field with negative strength can be used as **Predator**. Whether Boids seek or fly goals/predators depends on the **Physics** settings of the Boids.

**Wind**

![Wind force field](https://docs.blender.org/manual/en/dev/physics/force_fields/types/wind.html)

The *Wind* force field gives a constant force in a single direction, along the force object’s local Z axis. The strength of the force is visualized by the spacing of the circles shown.

![UI for a Wind force field](https://docs.blender.org/manual/en/dev/physics/force_fields/types/wind.html)

Title-Img. 1. 5 Wind force field.
Attribution-
Source-
Vortex

The Vortex force field gives a spiraling force that twists the direction of points around the force object’s local Z axis. This can be useful for making a swirling sink, or tornado, or kinks in particle hair.

Title-Img. 1. 7Vortex force field.
Attribution-
Source-
Link-

Title-Img. 1. 8UI for a Vortex force field.
Attribution-
Source-
Link-

**Magnetic**

![Magnetic force field](https://docs.blender.org/manual/en/dev/physics/force_fields/types/magnetic.html)

This field depends on the speed of the particles. It simulates the force of magnetism on magnetized objects.

![UI for a Magnetic force field](https://docs.blender.org/manual/en/dev/physics/force_fields/types/magnetic.html)

**Title**-Img. 1. 9Magnetic force field.

**Attribution**-
Source-
Link-
Link-

Harmonic

![Harmonic force field](https://docs.blender.org/manual/en/dev/physics/force_fields/types/harmonic.html)

Title-Img. 1. 11Harmonic force field.
Attribution-
Source-
Link-

In a *Harmonic* force field, the source of the force field is **the zero point** of a harmonic oscillator (spring, pendulum). If you set the *Damping parameter to 1*, the movement is stopped in the moment the object is reached. This force field is really special if you assign it to particles.

![UI for a Harmonic force field](https://docs.blender.org/manual/en/dev/physics/force_fields/types/harmonic.html)

Title-Img. 1. 12UI for a Harmonic force field.
Attribution-
Introduction to Dynamics

Source-Link:

Options:
- **Rest Length**
  Controls the rest length of the harmonic force.

- **Multiple Springs**
  This causes every point to be affected by multiple springs. Normally every particle of the field system influences every particle of the target system. Not with *Harmonic*! Here every target particle is assigned to a field particle. So particles will move to the place of other particles, thus forming shapes.

Tutorial:
https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Particles_for_ming_Shapes

Charge

Title-Img. 1. 13Charge force field.

Attribution-
Source-Link:

A *Charge* force field is similar to *spherical field* except it changes behavior (attract/repulse) based on the effected particles charge.
field (negative/positive), like real particles with a charge. This means this field has only effect on particles that have also a **Charge field** (else, they have no “charge”, and hence are unaffected)!

Title - Img. 1. 14UI for a Charge force field.

Attribution - 
Source - 
Link -

---

**Lennard Jones**

Title - Img. 1. 15Lennard Jones force field.

Attribution - 
Source -
The *Lennard Jones* force field is a **very short range force** with a behavior determined by the sizes of the effector and effected particle. At a distance smaller than the combined sizes the field is very repulsive and after that distance it is attractive. It tries to keep the particles at an equilibrium distance from each other. Particles need to be at a close proximity to each other to be effected by this field at all.

Particles can have for example both a **charge and a Lennard-Jones potential**, which is probably something for the nuclear physicists amongst us.
You can use a **Texture force field** to create an **arbitrarily complicated** force field, which force in the three directions is color coded. **Red** is coding for the X-axis, **green** for the Y-axis and **blue** for the Z-axis (like the color of the coordinate axes in the 3D View). A value of **0.5** means no force, a value **larger than 0.5** acceleration in negative axis direction (like -Z), a value **smaller than 0.5** acceleration in positive axis direction (like +Z).

**Title**-Img. 1. 18UI for a Texture force field.

**Attribution**

Source-

Link-  

**Options**

**Texture mode**

This sets the way a force vector is derived from the texture.

- **RGB**

Uses the color components directly as the force vector components in the color encoded directions. You need an RGB texture for this, e.g. an image or a color band. So, a **Blend** texture without a color band would not suffice.

- **Gradient**
Calculates the force vector as the 3D-gradient of the intensity (grayscale) of the texture. The gradient vector always points to the direction of increasing brightness.

- **Curl**

Calculates the force vector from the curl of the 3D-RGB texture (rotation of RGB vectors). This also works only with a color texture. It can be used for example to create a nice-looking turbulence force with a color clouds texture with Perlin noise.

- **Nabla**

It is the offset used to calculate the partial derivatives needed for Gradient and Curl texture modes.

- **Use Object Coordinates**

Uses the emitter object coordinates (and rotation & scale) as the texture space the particles use. Allows for moving force fields that have their coordinates bound to the location coordinates of an object.

- **Root Texture Coordinates**

This is useful for hair as it uses the texture force calculated for the particle root position for all parts of the hair strand.

- **2D**

The 2D button disregards the particles z-coordinate and only uses particles x & y as the texture coordinates.

Remember that only procedural texture is truly 3D.

**Examples**

A single colored texture (0.5, 0.0, 0.5) creates a force in the direction of the positive Y-axis, e.g. hair is orienated to the Y-axis.

A blend texture with color band can be used to create a force “plane”. E.g. on the left side (0.5, 0.5, 0.5), on the right side (1.0, 0.5, 0.5) you have a force plane perpendicular to XY (i.e. parallel to Z). If you use an object for the coordinates, you can use the object to push particles around.

An animated wood texture can be used to create a wave like motion.
The **Curve Guide** is used to force particles to follow a certain path defined by a Curve Object. A typical scenario would be to move a red blood cell inside a vein, or to animate the particle flow in a motor. You can use **Curve Guide**s also to shape certain hair strands.

**Tip**

You can also use the Particle Edit Mode to define a path.

Since you can animate curves as Soft Body or any other usual way, you may build very complex animations while keeping great control and keeping the simulation time to a minimum.

The option **Curve Follow** does not work for particles. Instead you must set **Angular Velocity** (Particle system tab) to **Spin** and leave the rotation constant (i.e. do not turn on **Dynamic**).

**Curve Guide** affects all particles on the same layer, independently from their distance to the curve. If you have several guides in a layer, their fields add up to each other (the way you may have
learned it in your physics course). However, you can limit their influence radius by changing their Minimum Distance.

Tip

The Curve Guide does not affect Soft Body.

Options

Title-Img. 1. 20UI for a Curve Guide force field.

Attribution-

Source-

Link- 


- Minimum Distance

It means the distance from the curve, up to where the force field is effective with full strength. If you have a Fall-off of 0 this parameter does nothing, because the field is effective with full strength up to Max Distance (or the infinity). Min Distance is shown with a circle at the endpoints of the curve in the 3D View.

- Free

Fraction of particle life time, which is not used for the curve.
Fall-off

This setting governs the strength of the guide between \textit{Min Distance} and \textit{Max Distance}. A Fall-off of 1 means a linear progression.

A particle follows a \textit{Curve Guide} during its lifetime, the velocity depends on its lifetime and the length of the path.

- \textbf{Additive}
  
  If you use \textit{Additive}, the speed of the particles is also evaluated depending on the Fall-off.

- \textbf{Weights}
  
  Use Curve weights to influence the particle influence along the curve.

- \textbf{Maximum Distance / Use Max}
  
  This maximum distance value influence radius which is shown by an additional circle around the curve object.

The other settings govern the form of the force field along the curve.

- \textbf{Clumping Amount}
  
  The particles come together at the end of the curve (1) or they drift apart (-1).

- \textbf{Shape}
  
  Defines the form in which the particles come together. +0.99: the particles meet at the end of the curve. 0: linear progression along the curve. -0.99: the particles meet at the beginning of the curve.

- \textbf{Kink}
  
  Changes the shape that the particles can take:

- \textbf{Curl}
  
  The radius of the influence depends on the distance of the curve to the emitter.

- \textbf{Radial}
  
  A three dimensional, standing wave.

- \textbf{Wave}
  
  A two dimensional, standing wave.
- **Braid**
  Pattern that follows along the curve.

- **Roll**
  A one dimensional, standing wave.

It is not so easy to describe the resulting shapes, so have a look at the example below.

**Title** - Img. 1. 21 Kink options of a curve guide. From left to right: Radial, Wave, Braid, and Roll.

**Attribution**
Source-
Link-

- **Frequency**
  The frequency of the offset.

- **Shape**
  Adjust the offset to the beginning/end.

- **Amplitude**
  The Amplitude of the offset.
Boid

Boid probably comes from theoretical works. Boids is an artificial life program, developed by Craig Reynolds (1986), which simulates the flocking behavior of birds. His paper on this topic was published in 1987 in the proceedings of the ACM SIGGRAPH conference. The name refers to a “bird-like object”, however, its pronunciation evokes that of “bird” in a stereotypical New York accent. As with most artificial life simulations, Boids is an example of emergent behavior; that is, the complexity of Boids arises from the interaction of individual agents (the Boids, in this case) adhering to a set of simple rules.
The rules applied in the simplest Boids world are as follows:

- **Separation**: steer to avoid crowding local flock mates
- **Alignment**: steer towards the average heading of local flock mates
- **Cohesion**: steer to move toward the average position (center of mass) of local flock mates

More complex rules can be added, such as obstacle avoidance and goal seeking.

**Turbulence**

A Turbulence force field creates a random & chaotic 3D noise effect, similar to jets of water or geysers under the ocean.
Title: Img. 1. 25UI for a Turbulence force field.
Attribution:
Source:

- **Size**
  Indicates the scale of the noise.

- **Global**
  Makes the size and strength of the noise relative to the world, instead of the object it is attached to.

Title: Img. 1. 26 Turbulence force field affecting a particle system.
Attribution:
Source:
Drag

A Drag force field resists particle motion by slowing it down.

Title-Img. 1. 27Drag force field.
Attribution-
Source-
Link-

Title-Img. 1. 28UI for a Drag force field.
Attribution-
Source-
3D Animation

Link-

- **Linear**
  
  Drag component proportional to velocity.

- **Quadratic**
  
  Drag component proportional to the square of the velocity.

**Smoke Flow**

![Image of Smoke Flow force field]

Title-Img. 1. Smoke Flow force field.

Attribution-

Source-

Link-

A *Smoke Flow* force field directs the smoke within a smoke simulation.
Title-Img. 1. 30UI for a Smoke Flow force field.
Attribution-
Source-
Link-
Unit summary

In this Unit, you have learnt what is Physical Simulation and Gravity and how to

- Use the real-world simulation effects in your 3D Projects like Dust, Smoke, Rain, Fire, Water etc.
- Use the force fields in your 3D simulation project.
- Apply Gravity in 3D Simulation using Rigid Body and Soft Body.

After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating 3D Interface.

Assignment

- Learners are expected to experiment the force fields options / setups and parameters available in the panel and to experiment the way the physical simulation works for different force fields from the links.
Assessment

- Describe Force Fields with examples.
- Define Magnetic Force Field.
- Explain Harmonic Force Field with examples.
- Define Charge Filed.
- Explain the Process of using curve as a guide in Force Field.
- Define Boid Field?
- List down the effects Turbulence filed can create.

Fill in the Blanks

1. Default value of Gravity in Blender is __________.
2. Baking refers to the act of __________ the results of a calculation.
3. Force Fields can also be generated from ______.
4. Field with positive strength can be used as a ________.
5. A field with negative strength can be used as ____________.
Resources

While studying this Unit, you can browse the following internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

- wiki.blender.org
- ia600207.us.archive.org
- archive.org
- www.blender.org
- docs.blender.org

**Study skills**

Links to download 3D Files for practice

2. [https://cloud.blender.org/training](https://cloud.blender.org/training)
Unit 2

Introduction to Rigid Body, Soft Body and Constraints

Introduction

In this Unit, you will be introduced to a Rigid Body which is an idealization of a Solid Body in which deformation is neglected. In other words, the distance between any two given points of a Rigid Body remains constant in time regardless of external forces exerted on it and Soft-body dynamics that it is a field of computer graphics that focus on visually realistic physical simulations of the motion and properties of deformable objects (or soft bodies).

You will also learn the different types of constraints available in Blender to control both Rigid and Soft Body.

Outcomes

Upon completion of this unit you will be able to:

- Create Passive Body and Active Body
- Demonstrate the utility of Rigid Body
- Use Multiple constraints
- Practice Soft Body animation
- Use constraints to control dynamic animation
- Reuse simulation as cache

Terminology

Active: Object is directly controlled by simulation results. One can select Active type with Add Active Button in the Physics tab of the Tool Shelf.
**Passive:** Object is directly controlled by animation system. Thus, this type is not available for Rigid Body Dynamics. The possibility to select this type also available with Add Passive Button in the Physics tab of the Tool Shelf.

**Dynamic:** Enables/disables Rigid Body simulation for object.

**Animated:** Allows the Rigid Body additionally to be controlled by the animation system.

**Mass:** Specifies how heavy the object is and “weights” irrespective of gravity. There is predefined mass preset available with the Calculate Mass Button in the Physics tab of the Tool Shelf.

---

**Rigid Body**

The Rigid Body simulation can be used to simulate the motion of solid objects. It affects the position and orientation of objects and does not deform them.

Unlike the other simulations in Blender, the Rigid Body simulation works closer with the animation system. This means that rigid bodies can be used like regular objects and be part of parent-child relationships, animation constraints and drivers.

---

Right now, only mesh objects can participate in the Rigid Body simulation.

To create Rigid Bodies,

- **Step 1:** Either click on Rigid Body Button in the Physics tab of the Properties editor
- **Step 2:** Or use Add Active/Add Passive buttons in the Physics tab of the Tool Shelf.

Title-Img. 2. 1Create Rigid Body.

Attribution-

Source-

There are two types of Rigid Body:

1. Active
2. Passive

*Active bodies* are dynamically simulated, *Passive bodies* remain static. Both types can be driven by the animation system when using the *Animated option*.

During the simulation, the Rigid Body system will *override* the position and orientation of dynamic Rigid Body objects. Note however, that the location and rotation of the objects is not changed, so the Rigid Body simulation acts similar to a *constraint*.

To apply the Rigid Body transformations, you can use the *Apply Transformation Button* in the *Physics* tab of the *Tool Shelf*.

The scale of the Rigid Body object also influences the simulation, however, is always controlled by the animation system.

Rigid Body physics on the object can be *removed* with the *Rigid Body Button* in the *Physics* tab or *Remove* Button in the *Physics* tab of the *Tool Shelf*.

**Properties of Rigid Body**
These are

- **Type**
  
  Role of the Rigid Body in the simulation. **Active objects** can be simulated **dynamically**, whereas **Passive object** remain **static**.

- **Active**

  Object is directly controlled by simulation results. The possibility to select this type also available with **Add Active Button** in the **Physics tab of the Tool Shelf**.

- **Passive**
Object is directly controlled by animation system. Thus, this type is not available for Rigid Body Dynamics. The possibility to select this type also available with **Add Passive Button** in the *Physics* tab of the *Tool Shelf*.

- **Dynamic**
  Enables/disables Rigid Body simulation for object.

- **Animated**
  Allows the Rigid Body additionally to be controlled by the animation system.

- **Mass**
  Specifies how heavy the object is and “weights” irrespective of gravity. There is predefined mass preset available with the **Calculate Mass Button** in the *Physics* tab of the *Tool Shelf*.

  - **Calculate Mass**
    Automatically calculate mass values for Rigid Body Objects based on volume. There are many useful presets available from the menu, patching real-world objects.

- **Note**
  Also, you can have *Custom* mass material type, which is achieved by setting a custom density value (kg/m$^3$).

**Rigid Body Collisions**

Rigid Body Collisions panel.

**Collision shapes**

The *Shape* option determines the collision shape of the object. The following Collision Shapes are available:

- **Primitive shapes**
  these are best in terms of memory/performance however, do not necessarily reflect the actual shape of the object. They are calculated based on the object’s bounding box. The center of gravity is always in the middle for now.
- **Box**
  Box-like shapes (i.e. cubes), including planes (i.e. ground planes). The size per axis is calculated from the bounding box.

- **Sphere**
  Sphere-like shapes. The radius is the largest axis of the bounding box.

- **Capsule**
  This points up the Z-Axis.

- **Cylinder**
  This points up the Z-Axis. The height is taken from the z-axis, while the radius is the larger of the x/y-axes.

- **Cone**
  This points up the Z-Axis. The height is taken from the z-axis, while the radius is the larger of the x/y-axes.

**Mesh based shapes**

These are calculated based on the geometry of the object so they are a better representation of the object. The center of gravity for these shapes is the object origin.

- **Convex Hull**
  A mesh-like surface encompassing (i.e. shrink-wrap over) all vertices (best results with fewer vertices). Convex approximation of the object, has good performance and stability.

- **Mesh**
  Mesh consisting of triangles only, allowing for more detailed interactions than convex hulls. Allows to simulate concave objects, however, is rather slow and unstable.

The changing collision shape is available also with **Change Shape Button** in the **Physics** tab of the **Tool Shelf**.

**Mesh source**
Users can now specify the mesh Source for Mesh bases collision shapes:

- **Base**
  The base mesh of the object.

- **Deform**
  Includes any deformations added to the mesh (shape keys, deform modifiers).

  **Deforming**

  Rigid Body deforms during simulation.

- **Final**
  Includes all modifiers.

**General settings**

- **Surface Response**
  **Friction**

  Resistance of object to movement. Specifies how much velocity is lost when objects collide with each other.

  **Bounciness**

  Tendency of object to bounce after colliding with another (0 to 1) (rigid to perfectly elastic). Specifies how much objects can bounce after collisions.

- **Collision Groups**

  Allows Rigid Body collisions allocate on different groups (maximum 20).

**Collision Margin**

**Margin**

Threshold of distance near surface where collisions are still considered (best results when non-zero).

The collision margin is used to improve performance and stability of rigid bodies. Depending on the shape, it behaves differently: some shapes embed it, while others have a visible gap around them.

The margin is **embedded** for these shapes:
• Sphere
• Box
• Capsule
• Cylinder
• Convex Hull: Only allows for uniform scale when embedded.

The margin is not embedded for these shapes:
• Cone
• Active Triangle Mesh
• Passive Triangle Mesh: Can be set to 0 most of the time.

Rigid Body Dynamics

Used to control the physics of the rigid body simulation. Rigid Body Dynamics panel.

This panel is available only for Active type of rigid bodies.

Deactivation

• Enable Deactivation
  Enable deactivation of resting rigid bodies. Allows object to be deactivated during the simulation (improves performance and stability, however, can cause glitches).

• Start Deactivated
  Starts objects deactivated. They are activated on collision with other objects.

• Linear Velocity
  Specifies the linear deactivation velocity below which the Rigid Body is deactivated and simulation stops simulating object.

• Angular Velocity
  Specifies the angular deactivation velocity below which the Rigid Body is deactivated and simulation stops simulating object.

Damping

A reduction in the amplitude of an oscillation as a result of energy being drained from the system to overcome frictional or other resistive forces.
• **Translation**
  Amount of linear velocity that is lost over time.

• **Rotation**
  Amount of angular velocity that is lost over time.

**Rigid Body World**

Rigid Body objects and constraints are only taken into account by the simulation if they are in the groups specified in *Group* field of the *Rigid Body World* panel in the *Scene* tab.

• **Rigid Body World**
  Enable/disable evaluation of the Rigid Body simulation based on the Rigid Body objects participating in the specified group of Rigid Body World.

• **Remove Rigid Body World**
  Remove Rigid Body simulation from the current scene.

• **Group**
  Containing Rigid Body objects participating in this simulation.

• **Constraints**
  Containing Rigid Body object constraints participating in the simulation.

Simulation quality and timing settings:

• **Speed**
  Can be used to speed up/slow down the simulation.

• **Split Impulse**
  Enable/disable reducing extra velocity that can build up when objects collide (lowers simulation stability a little so use only when necessary). Limits the force with which objects are separated on collision, generally produces nicer results, however, makes the simulation less stable (especially when stacking many objects).

• **Steps Per Second**
Number of simulation steps made per second (higher values are more accurate however, slower). This only influences the accuracy and not the speed of the simulation.

- **Solver Iterations**
  Amount of constraint solver iterations made per simulation step (higher values are more accurate however, slower). Increasing this makes constraints and object stacking more stable.

**Rigid Body Cache**

The *Rigid Body Cache* panel specifies the frame range in which the simulation is active. Can be used to bake the simulation.

- **Start/End**
  First and last frame of the simulation.

- **Bake**
  Calculates the simulation and protects the cache. You need to be in *Object Mode* to bake.

- **Free Bake**
  Active after the baking of simulation. Clears the baked cache.

- **Calculate to Frame**
  Bake physics to current frame.

- **Current Cache to Bake**
  Bake from Cache.

- **Bake All Dynamics**
  Bake all physics.

- **Free All Bakes**
  Free all baked caches of all objects in the current scene.

- **Update All to Frame**
  Update cache to current frame.
If you haven’t saved the blend-file, the cache is created in memory, so save your file first or the cache may be lost.

**Rigid Body Field Weights**

As other physics dynamics systems, Rigid Body simulation are also influenced by external force effectors.

**Rigid Body Constraints**

Constraints (also known as Joints) for rigid bodies connect two Rigid Bodies.

The physics constraints available in the non-game modes are meant to be attached to an Empty object. The constraint then has fields which can be pointed at the two physics-enabled object which will be bound by the constraint. The Empty object provides a location and axis for the constraint distinct from the two constrained objects. The location of the entity hosting the physics constraint marks a location and set of axes on each of the two constrained objects. These two anchor points are calculated at the beginning of the animation and their position and orientation remain fixed in the local coordinate system of the object for the duration of the animation. The objects can move far from the constraint object; however, the constraint anchor moves with the object. If this feature seems limiting, consider using multiple objects with a non-physics Child-of constraint and animate the relative location of the child.

**Connect**

The quickest way to constrain two objects is to select both and click the **Connect Button** in the Physics tab of the Tool Shelf. This creates a new Empty object (named “Constraint”) with a physics constraint already attached and pointing at the two selected objects.

**Physics Tab**

Also, you can create Rigid Body Constraint on of the two constrained objects with **Rigid Body Constraint Button** of the Physics tab in the Properties editor. This constraint is dependent on the object location and rotation on which it was created. This way, there are no Empty object created for the constraint. The role of the Empty object is put on this object. The constrained object can be then set as Passive type for better driving the constrain.

Additional parameters appear in the Rigid Body Constraint panel of the Physics tab in the Properties editor for the selected
Empty object or the one of the two constrained objects with the created constraint.

**Common Options**

Rigid Body Constraint panel.

- **Enabled**
  Specifies whether the constraint is active during the simulation.

- **Disable Collisions**
  Allows constrained objects to pass through one another.

- **Object 1**
  First object to be constrained.

- **Object 2**
  Second object to be constrained.

- **Breakable**
  Allows constraint to break during simulation. Disabled for the *Motor* constraint.

- **Threshold**
  Impulse strength that needs to be reached before constraint breaks.

- **Override Iterations**
  Allows to make constraints stronger (more iterations) or weaker (less iterations) than specified in the Rigid Body world.

- **Iterations**
  Number of constraint solver iterations made per simulation step for this constraint.

- **Limits**
  By using limits, you can constrain objects even more by specifying a translation/rotation range on/around respectively axis (see below for each one individually). To lock one axis, set both limits to 0.
Soft Body

A Soft Body in general, is a simulation of a **soft or rigid deformable object**. In Blender, this system is best for simple **cloth objects and closed meshes**. There is dedicated Cloth Simulation physics that use a different solver, and is better for cloth.

This simulation is done by applying forces to the vertices or control points of the object. There are exterior forces like gravity or force fields and interior forces that hold the vertices together. This way you can simulate the shapes that an object would take on in reality if it had volume, was filled with something, and was acted on by real forces.

Soft Bodies can interact with other objects through **Collision**. They can interact with themselves through **Self Collision**.

The result of the Soft Body simulation can be converted to a **static object**. You can also **bake edit** the simulation, i.e. edit intermediate results and run the simulation from there.

**Typical scenarios for using Soft Bodies**

- Elastic objects with or without collision.
- Flags, fabric reacting to forces.
- Certain modeling tasks, like a cushion or a table cloth over an object.
• Blender has another simulation system for clothing. However, you can sometimes use Soft Bodies for certain parts of clothing, like wide sleeves.
• Hair (as long as you minimize collision).
• Animation of swinging ropes, chains and the like.

Creating Soft Body

Soft Body simulation works for all objects that have vertices or control points:
• Meshes.
• Curves.
• Surfaces.
• Lattices.

To activate the Soft Body simulation for an object:
• **Step 1:** In the Properties editor, go to the *Physics* tab (it is all the way on the right, and looks like a bouncing ball).
• **Step 2:** Activate the *Soft Body* button.
• **Step 3:** You start a Soft Body simulation with Alt-A.
• **Step 4:** You pause the simulation with Spacebar, continue with Alt-A.
• **Step 5:** You stop the simulation with Esc.

Simulation Quality

The settings in the *Soft Body Solver* panel determine the accuracy of the simulation.

• **Min Step**
Minimum simulation steps per frame. Increase this value, if the Soft Body misses fast moving collision objects.

• **Max Step**
Maximum simulation steps per frame. Normally the number of simulation steps is set dynamically (with the *Error Limit*) however, you have probably a good reason to change it.

• **Auto-Step**
Use Velocities for automatic step sizes.
• **Error Limit**
Rules the overall quality of the solution delivered, default value is 0.1. The most critical setting that says how precise the solver should check for collisions. Start with a value that is 1/2 the average edge length. If there are visible errors, jitter, or over-exaggerated responses, decrease the value. The solver keeps track of how “bad” it is doing and the ErrorLimit causes the solver to do some “adaptive step sizing”.

• **Fuzzy**
Simulation is faster, however, less accurate.

• **Choke**
Calms down (reduces the exit velocity of) a vertex or edge once it penetrates a collision mesh.

**Diagnostics**

• **Print Performance to Console**
Prints on the console how the solver is doing.

• **Estimate Matrix**
Estimate matrix. Split to COM, ROT, SCALE

**Cache and Bake**
Soft Bodies and other physic simulations use a unified system for caching and baking.

The results of the simulation are automatically cached to disk when the animation is played, so that the next time it runs, it can play again quickly by reading in the results from the disk. If you *Bake the simulation*, the cache is protected and you will be asked when you are trying to change a setting that will make a recalculating necessary.

**Tip**

*Beware of the Start and End settings*

The simulation is only calculated for the frames in-between the Start and End frames (Bake panel), even if you do not actually bake the simulation! So if you want a simulation longer than the default setting of 250 frames you have the change the End frame.
Caching

- As animation is played, each physics system writes each frame to disk, between the simulation start and end frames. These files are stored in folders with prefix blend cache, next to the blend-file.
- The cache is cleared automatically on changes. However, not on all changes, so it may be necessary to free it manually, e.g. if you change a force field. Note that for the cache to fill up, one must start playback before or on the frame that the simulation starts.
- If you are not allowed to write to the required sub-directory caching will not take place.
- The cache can be freed per physics system with a Button in the panels, or with the Ctrl-B shortcut key to free it for all selected objects.
- You may run into trouble if your blend-file path is very long and your operating system has a limit on the path length that is supported.

Baking

- The system is protected against changes after baking.
- The Bake result is cleared also with Ctrl-B for all selected objects or click on Free Bake for the current Soft Body system.
- If the mesh changes the simulation is not calculated anew.
- For render farms, it is best to bake all the physics systems, and then copy the blend cache to the render farm as well.

Interaction in real time

To work with a Soft Body simulation, you will find it handy to use the Timeline editor. You can change between frames and the simulation will always be shown in the actual state. The option Continue Physics in the Playback menu of the Timeline editor lets you interact in real time with the simulation, e.g. by moving collision objects or shake a Soft Body object.
If the objects have an even vertex distribution then the Soft Bodies work especially well. You need enough vertices for good collisions. You change the deformation (the stiffness) if you add more vertices in a certain region.

The calculation of collisions may take a long time. If something is not visible, why calculate it?

To speed up the collision calculation it is often useful to collide with an additional, simpler, invisible, somewhat larger object.

Use Soft Bodies only where it makes sense. If you try to cover a body mesh with a tight piece of cloth and animate solely with Soft Body, you will have no success. Self-collision of Soft Body hair may be activated, however, that is a path that you must wander alone. We will deal with Collisions in detail later.

Try and use a **Lattice** or a **Curve Guide Soft Body** instead of the object itself. This may be magnitudes faster.

**Simple Example**

Here are some simple examples showing the power of Soft Body physics.

**Bouncing Cube**

The Process

![Timeline](https://docs.blender.org/manual/en/dev/physics/soft_body/examples.html?highlight=bouncing%20cube)
Step 1: First, change your start and end frames to 1 and 150.
  - **Step 2:** Then, add a plane, and scale it five times.
  
  - **Step 3:** Next, go to the physics tab, and add a collision. The default settings are fine for this example.
  
  - **Step 4:** Now add a cube, or use the default cube.
  
  - **Step 5:** Tab into edit mode and subdivide it three times.
  
  - **Step 6:** Add a bevel modifier to it to smoothen the edges
  
  - **Step 7:** Add a little more, press r twice, and move your cursor a bit.

When finished, your scene should look like this:

![Image 2.5](https://docs.blender.org/manual/en/dev/physics/soft_body/examples.html?highlight=bouncing%20cube)

**Title**- Img. 2.5 The scene, ready for Soft Body physics.

**Attribution**-

**Source**-


Everything is ready to add the Soft Body physics.

  - **Step 1:** Go to Properties > Physics and choose **Soft Body**.
• **Step 2:** Uncheck the Soft Body goal, and check Soft Body self-collision.

• **Step 3:** Also, under Soft Body edges, increase the bending to 10.

• **Step 4:** Playing the animation with Alt-A will now give a slow animation of a bouncing cube.

• **Step 5:** To speed things up, we need to bake the Soft Body physics.

• **Step 6:** Under *Soft Body Cache*, change the start and end values to your start and end frames. In this case 1 and 150.

• **Step 7:** Now, to test if everything is working, you can take a cache step of 5 or 10, however, for the final animation it is better to reduce it to 1, to cache everything.

When finished, your physics panel should look like this:

---

**Title** - Img. 2. 6 The physics settings.

**Attribution**

**Source**

**Link**

You can now bake the simulation, give the cube materials and textures and render the animation.

The Result
The rendered bouncing cube:

Rigid Body Constraint Types

Fixed Constraint

Title-Img. 2. 7 Options available to a Fixed constraint.
Attribution-
Source-
Link-

This constraint causes the two objects to move as one. Since the physics system does have a tiny bit of slop in it, the objects do not move as rigidly as they would if they were part of the same mesh.
Point Constraint

The objects are linked by a point bearing allowing any kind of rotation around the location of the constraint object, however, no relative translation is permitted. The physics engine will do its best to make sure that the two points designated by the constraint object on the two constrained objects are coincident.

Hinge

Title- Img. 2. 9 Options available to a Hinge constraint.
Attribution-
The Hinge permits **1 degree of freedom between two objects**. Translation is completely constrained. Rotation is permitted about the **Z axis** of the object hosting the Physics constraint (usually an Empty, distinct from the two objects that are being linked). Adjusting the position and rotation of the object hosting the constraint allows you to control the anchor and axis of the hinge.

The Hinge is the only **1-axis** rotational constraint that uses the **Z axis** instead of the **X axis**. If something is wrong with your hinge, check your other constraints to see if this might be the problem.

**Options - Limits**

- **Z Angle**
  Enables/disables limit rotation around Z axis.

- **Lower**
  Lower limit of Z axis rotation.

- **Upper**
  Upper limit of Z axis rotation.

**Slider Constraint**

The Slider constraint allows relative translation along the **X axis** of the constraint object, however, permits no relative rotation, or relative translation along other axes.

**Options - Limits**

- **X Axis**
  Enables/disables limit translation around X axis.

- **Lower**
  Lower limit of X axis translation.

- **Upper**
  Upper limit of X axis translation

**Piston Constraint**
A piston permits translation along the X axis of the constraint object. It also allows rotation around the X axis of the constraint object. It is like a combination of the freedoms of a slider with the freedoms of a hinge (neither of which is very free alone).

**Options - Limits**

- **X Axis**
  Enables/disables limit translation around X axis.

- **Lower**
  Lower limit of X axis translation.

- **Upper**
  Upper limit of X axis translation.

- **X Angle**
  Enables/disables limit rotation around X axis.

- **Lower**
  Lower limit of X axis rotation.

- **Upper**
  Upper limit of X axis rotation

**Generic Constraint**

The generic constraint has a lot of available parameters.

The **X, Y, and Z axis** constraints can be used to limit the amount of translation between the objects. Clamping the min/max to zero has the same effect as the Point constraint.

Clamping the relative rotation to **zero** keeps the objects in alignment. Combining an absolute rotation and translation clamp would behave much like the **Fixed constraint**.

Using a **non-zero spread** on any parameter allows it to rattle around in that range throughout the course of the simulation.

**Options - Limits**

- **X Axis/Y Axis/Z axis**
  Enables/disables limit translation on X, Y or Z axis respectively.

  - **Lower**
Lower limit of translation for X, Y or Z axis respectively.

- **Upper**

Upper limit of translation for X, Y or Z axis respectively.

- **X Angle/Y Angle/Z Angle**

Enables/disables limit rotation around X, Y or Z axis respectively.

- **Lower**

Lower limit of rotation for X, Y or Z axis respectively.

- **Upper**

Upper limit of rotation for X, Y or Z axis respectively.

**Generic Spring Constraint**

The generic spring constraint adds some spring parameters for the X/Y/Z axes to all the options available on the Generic constraint. Using the spring alone allows the objects to bounce around as if attached with a spring anchored at the constraint object. This is usually a little too much freedom, so most applications will benefit from enabling translation or rotation constraints.
If the damping on the springs is **set to 1**, then the spring forces are prevented from realigning the anchor points, leading to strange behavior. If your springs are acting weird, check the damping.

**Options - Limits**

- **X Axis/Y Axis/Z axis**
  Enables/disables limit translation on X, Y or Z axis respectively.
  
  - **Lower**
    Lower limit of translation for X, Y or Z axis respectively.
  
  - **Upper**
    Upper limit of translation for X, Y or Z axis respectively.

- **X Angle/Y Angle/Z Angle**
  Enables/disables limit rotation around X, Y or Z axis respectively.
  
  - **Lower**
    Lower limit of rotation for X, Y or Z axis respectively.
  
  - **Upper**
    Upper limit of rotation for X, Y or Z axis respectively.

**Options - Springs**

- **X/Y/Z**
  Enables/disables springs on X, Y or Z axis respectively.
  
  - **Stiffness**
    Spring Stiffness on X, Y or Z axis respectively. Specifies how “bendy” the spring is.
  
  - **Damping**
    Spring Damping on X, Y or Z axis respectively. Amount of damping the spring has.

**Motor Constraint**
The motor constraint causes translation and/or rotation between two entities. It can drive two objects apart or together. It can drive simple rotation, or rotation and translation (although it will not be constrained like a screw since the translation can be blocked by other physics without preventing rotation).

The rotation axis is the X axis of the object hosting the constraint. This is in contrast with the Hinge which uses the Z axis. Since the Motor is vulnerable to confusing perturbations without a matching Hinge constraint, special care must be taken to align the axes. Without proper alignment, the motor will appear to have no effect (because the Hinge is preventing the motion of the motor).

**Options - Linear motor/Angular motor**

- **Enable**
  
  Enable linear or angular motor respectively.

- **Target Velocity**
  
  Target linear or angular motor velocity respectively.

- **Max Impulse**
  
  Maximum linear or angular motor impulse respectively.
Animation

The most common trick is to **key frame animate** the location or rotation of an *Active* physics object as well as the *animated* checkbox. When the curve on the *animated* property switches to disabled, the physics engine takes over using the object’s last known location, rotation and velocities.

Animating the strengths of various other parameters (a Motor’s Target Velocity, a Hinge’s limits, etc.) can be used to accomplish a wide variety of interesting results.

Enabling a constraint during the **physics simulation** often has dramatic results as the physics engine tries to bring into alignment two objects which are often dramatically out of alignment. It is very common for the affected objects to build up enough **kinetic energy** to bounce themselves out of camera (and into orbit, although the physics engine is not yet capable of simulating a planet’s gravity well, so scratch that).

Rigid Body dynamics can be **baking to normal keyframes** with *Bake to Keyframes Button* in the *Physics* tab of the *Tool Shelf*.

Simulation Stability

The simplest way of improving simulation stability is to **increase the steps per second**. However, care must be taken since making too many steps can cause problems and make the simulation even less stable (if you need **more than 1000 steps**, you should look at other ways to improve stability).

Increasing the number of solver iterations helps making constraints stronger and also improves object stacking stability.

It is best to **avoid small objects**, as they are currently unstable. Ideally, objects should be at **least 20 cm in diameter**. If it is still necessary, setting the **collision margin to 0**, while generally not recommended, can help making small object behave more naturally.
When objects are small and/or move very fast, they can pass through each other. Besides what is mentioned above it’s also good to avoid using mesh shapes in this case. Mesh shapes consist of individual triangles and therefore do not really have any thickness, so objects can pass through more easily. You can give them some thickness by increasing the collision margin.

**Combining Rigid Bodies with Other Simulations**

Since the Rigid Body simulation is part of the animation system, it can influence other simulations just like the animation system can.

In order for this to work, the Rigid Body object needs to have a Collision Modifier. Simply click on Collision in the Physics tab.

**Scaling Rigid Bodies**

Rigid Body objects can be scaled, also during the simulation. This work well in most cases, however, can sometimes cause problems.

If dynamic scaling is not needed, Rigid Body objects should have the scale applied by using the Apply Scale Command Ctrl-A.
Unit summary

In this Unit, you have learnt what is Rigid Body and Soft Body and how to

- Use the Constraints to control the dynamic motion of the Active and Passive bodies in real time.
- Create different type of dynamic animation using the Active, Passive bodies to match the real-world physics.

After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating 3D Interface.

Assignment

- Create a **Simple bouncing ball** using one – Soft Body and one – Rigid Body.
- Use the key words “**bouncing ball**” on [www.google.com](http://www.google.com) to collect the reference video to build your scene.

Assessment

1. Describe Active and Passive Rigid Bodies
2. List the Parameters of Rigid Body Collision
3. Explain the Primitive Shapes with examples
4. Explain Hinge Constraint
5. Explain Soft Body.
6. Define Motor Constraint
Fill in the Blanks

1. There are two types of Rigid Body: ________ and _________
2. The ______ option determines the collision shape of the object
3. The quickest way to constrain two objects is to select both and click the ________ Button
4. Constraint is dependent on the object ___________
5. ________ allows to make constraints stronger.

Resources

While studying this Unit, you can browse the internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

- wiki.blender.org
- ia600207.us.archive.org
- archive.org
- www.blender.org
- docs.blender.org

Study Skills
Unit 3

Welcome to Cloth Simulations

Introduction

In this Unit, you will be introduced with Cloth Simulations and to describe which objects will be part of the Simulation and which are not. Once you have done this, you can define what the objects are made of. You can also specify what Cloth is made of, and what a solid or collision object is. This is where you define Cloth and collision objects, assign properties, and execute the Simulation. Other controls included is creating constraints, interactively dragging the cloth, and erasing parts of the Simulation.

Outcomes

Upon completion of this unit you will be able to:

- Demonstrate the utility of Cloth Simulation
- Use Cloth Simulation to create flags and banners
- Practice Cloth Simulation to make clothing to your 3D Assets
- Use Cloth as extra effects in your 3D Scene

Terminology

Mass: The mass of the Cloth material.

Bending: It is the Wrinkle coefficient, higher the value creates more large folds on the cloth.

Damping: Damping of Cloth velocity. Higher values give a smoother result (less jiggling).

Velocity: Damps the velocity to help the Cloth reach the final resting position faster.
Cloth Simulations

Cloth Simulation is one of the hardest aspects of Computer Graphics (CG), because it is a deceptively simple real-world item that is taken for granted, yet actually has very complex internal and environmental interactions. After years of development, Blender has a very robust Cloth simulator that is used to make clothing, flags, banners, and so on. Cloth interacts with and is affected by other moving objects, the wind and other forces, as well as a general aerodynamic model, all of which is under your control.

A piece of Cloth is any Mesh, open or enclosed, that has been designated as cloth. The Cloth panels are located in the Physics tab and consist of three panels of options. Cloth is either an open or closed Mesh and is mass-less, in that all Cloth is assumed to have the same density, or mass per square unit.

Cloth is commonly modeled as a Mesh grid primitive, or a cube, but can also be, for example, a teddy bear. However, Blender’s Soft body system provides better Simulation of closed meshes; Cloth is a specialized Simulation of fabrics.

Once the object is designated as Cloth, a Cloth Modifier will be added to the object’s Modifier Stack automatically. As a Modifier, then, it can interact with other Modifiers, such as Armature and Smooth. In these cases, the ultimate shape of the Mesh is computed in accordance with the order of the Modifier stack. For
example, you should smooth the Cloth after the Modifier computes the shape of the cloth.

So, you edit the Cloth settings in two places:

1. Use the Physics buttons to edit the properties of the Cloth

2. Use the Modifier Stack to edit the Modifier properties related to display and interaction with other Modifiers.

You can apply the Cloth Modifier to freeze, or lock in, the shape of the Mesh at that frame, which removes the Modifier. For example, you can drape a flat Cloth over a table, let the Simulation run, and then apply the Modifier. In this sense, you are using the simulator to save yourself a lot of modeling time.

Results of the Simulation are saved in a cache, so that the shape of the Mesh, once calculated for a frame in an animation, does not must be recomputed again. If changes to the Simulation are made, you have full control over clearing the cache and re-running the Simulation. Running the Simulation for the first time is fully automatic and no baking or separate step interrupts the workflow.

Computation of the shape of the Cloth at every frame is automatic and done in the background; thus, you can continue working while the Simulation is computed. However, it is CPU-intensive and depending on the power of your PC and the complexity of the Simulation, the amount of CPU needed to compute the Mesh varies, as does the lag you might notice.

If you set up a Cloth Simulation but Blender has not computed the shapes for the duration of the Simulation, and if you jump ahead a lot of frames forward in your animation, the Cloth simulator may not be able to compute or show you an accurate Mesh shape for that frame, if it has not previously computed the shape for the previous frame(s).

**Cloth Workflow**

A general process for working with Cloth is to:

- Model the Cloth object as a general starting shape.

- Designate the object as a “cloth” in the Physics tab of the Properties editor.
• Model other deflection objects that will interact with the cloth. Ensure the Deflection Modifier is last on the Modifier stack, after any other Mesh Deforming Modifiers.

• Light the Cloth and assign materials and textures, UV-unwrapping if desired.

• If desired, give the object particles, such as steam coming off the surface.

• Run the Simulation and adjust Options to obtain satisfactory results. The Timeline editors’ VCR controls are great for this step.

• Optionally age the Mesh to some point in the Simulation to obtain a new default starting shape.

• Make minor edits to the Mesh on a frame-by-frame basis to correct minor tears.

Tip
To avoid unstable Simulation, ensure that the Cloth object does not penetrate any of the deflection objects or an unstable Simulation will result.

Cloth Settings

Cloth

• Presets
Contains a number of preset Cloth examples.

• Quality
Set the number of Simulation steps per frame. Higher values result in better quality, but is slower.

• Speed
Adjust how fast time flows in the Cloth Simulation.

Material

• Mass
The mass of the Cloth material.
- **Structural**
  Overall stiffness of the cloth.

- **Bending**
  Wrinkle coefficient. Higher creates more large folds.

**Damping**

- **Spring**
  Damping of Cloth velocity. Higher values give a smoother result (less jiggling).

- **Air**
  Air normally has some thickness which slows falling things down.

- **Velocity**
  Damps the velocity to help the Cloth reach the final resting position faster.

**Pinning**

The first thing you need when pinning Cloth is a Vertex Group. There are several ways of doing this including using the Weight Paint tool to paint the areas you want to pin. The weight of each vertex in the group controls how strongly it is pinned.
Once you have a vertex group set, things are pretty straightforward; all you must do is press the **Pinning of Cloth** button in the **Cloth** panel and select which vertex group you want to use, and the stiffness you want it at.

- **Stiffness**
  Target position stiffness. You can leave the stiffness as it is; the default value of 1 is fine.

**Cloth Pinning to an Armature**

Clothing can be **simulated and pinned** to an armature. For example, a character could have a baggy tunic pinned to the character’s waist with a belt.

- The typical workflow for pinning:
  - Set the armature to its bind pose.
  - Model clothing that encloses but does not penetrate the character’s Mesh.
  - Parent the clothing objects to the armature. The armature will now have several children meshes bound to it.
  - Create a new vertex group on each Cloth object for its pinned vertices
  - Add vertexes to be pinned to this vertex group and give these vertices non-zero weights (you probably want weight = 1). For example, the belt area of the tunic would be in the vertex group and have weight one.
  - Designate the clothing objects as “cloth” in the Physics tab of the Properties editor. Make sure the Cloth Modifier is below the Armature Modifier in the Modifier stack.
  - Press the **Pinning of Cloth** button in the **Cloth** panel and select the vertex group.
  - Designate the character’s Mesh as “collision” object in the Physics tab of the Properties editor.
• The clothing is now ready. Non-pinned vertices will be under control of the Cloth Modifier. Pinned vertices will be under control of the Armature Modifier.

Tip

When animating or posing the character, you must begin from the bind pose. Move the character to its initial pose over several frames so the physics engine can simulate the clothing moving. Very fast movements and teleport jumps can break the physics Simulation.

Dynamic Mesh

Normally Cloth uses the state of the object in the first frame to compute the natural rest shape of the cloth, and keeps that constant throughout the Simulation. This is reasonable for fully realistic scenes, but does not quite work for clothing on **cartoon style characters** that use a lot of squash and stretch.

When **Dynamic Mesh** is enabled, the rest shape is recalculated every frame, allowing unpinned Cloth to squash and stretch following the character with the help of an Armature Modifier, but otherwise move freely under control of the physics Simulation.

**Dynamic Mesh** is incompatible with using a shape key to specify the rest shape.

Cloth Sewing Springs

Another method of restraining Cloth similar to pinning is **Sewing Springs**. Sewing Springs are **virtual springs** that pull vertices in one part of a Cloth Mesh toward vertices in another part of the Cloth Mesh. This is different from pinning which binds vertices of the Cloth Mesh in place or to another object. A clasp on a cloak could be created with a Sewing Spring. The spring could pull two corners of a cloak about a character’s neck. This could result in a more realistic Simulation than pinning the cloak to the character’s neck since the cloak would be free to slide about the character’s neck and shoulders.

Sewing Springs are created by **adding extra edges to a Cloth Mesh** that are not included in any faces. They should connect vertices in the Mesh that should be pulled together. For example, the corners of a cloak.

To activate the springs, enable the **Cloth Sewing Springs** panel.

Options
Unit 3  Welcome to Cloth Simulations

- **Sewing Force**
  Maximum force that can be applied by Sewing Springs. Zero means unbounded, but it is not recommended to leave the field at zero in most cases, as it can cause instability due to extreme forces in the initial frames where the ends of the Sewing Springs are far apart.

  The *Cloth Sewing Springs* panel also contains **controls for shrinking** the actual Cloth faces.

  **Shrinking Group**
  Vertex group that is used to vary the intensity of the shrinking effect over the cloth.
  - **Min**
    Fraction of the size to shrink the Cloth by around vertices with weight 0 (or those not in vertex group.) The value 0.01 means shrink by 1% etc.
  - **Max**
    Fraction of the size to shrink the Cloth by around vertices with weight 1.

    Like unbounded sewing forces, immediately applying a large amount of shrink can cause instability, so it is advisable to Keyframe these fields and ease in from 0 during draping.

  **Collisions**
  In most cases, a piece of Cloth **does not just hang** there in 3D space, it **collides with other** objects in the environment. To ensure proper Simulation, there are several items that must be set up and working together:
  - The *Cloth* object must be told to participate in collisions.
  - Optionally (but recommended) tell the Cloth to collide with itself.
  - Other objects must be visible to the *Cloth* object *via* shared layers.
  - The other objects must be Mesh objects.
- The other objects may move or be themselves deformed by other objects (like an armature or shape key).

- The other Mesh objects must be told to deflect the Cloth object.

- The blend-file must be saved in a directory so that Simulation results can be saved.

- You then *Bake* the Simulation. The simulator computes the shape of the Cloth for a frame range.

- You can then edit the Simulation results, or make adjustments to the Cloth Mesh, at specific frames.

You can make adjustments to the environment or deforming objects, and then re-run the Cloth Simulation from the current frame forward.

**Collision Settings**

![Cloth Collision settings panel]

Now you must tell the *Cloth* object that you want it to participate in collisions. For the Cloth object, locate the *Cloth Collision* panel, shown to the right:

- **Quality**
  
  A general setting for how fine and good a Simulation you wish. Higher numbers take more time but ensure less tears and penetrations through the cloth.

- **Distance**

**Title**-Img. 3. 2Cloth Collisions panel.

**Attribution**-

**Source**-

**Link**-

As another object gets this close to it (in Blender Units), the Simulation will start to push the Cloth out of the way.

- **Repel**
  Repulsion force to apply when Cloth is close to colliding.

- **Repel Distance**
  Maximum distance to apply repulsion force. Must be greater than minimum distance.

- **Friction**
  A coefficient for how slippery the Cloth is when it collides with the Mesh object. For example, silk has a lower coefficient of friction than cotton.

**Self-Collisions**

Real Cloth cannot permeate itself, so you normally want the Cloth to self-collide.

- **Enable Self Collisions**
  Click this to tell the Cloth object that it should not penetrate itself. This adds to Simulation compute time, but provides more realistic results. A flag, viewed from a distance does not need this enabled, but a close-up of a cape or blouse on a character should have this enabled.

- **Quality**
  For higher self-collision quality just increase the *Quality* and more self-collision layers can be solved. Just keep in mind that you need to have at least the same *Collision Quality* value as the *Quality* value.

- **Distance**
  If you encounter problems, you could also change the *Min Distance* value for the self-collisions. The best value is 0.75; for fast things, you better take 1.0. The value 0.5 is quite risky (most likely many penetrations) but also gives some speedup.

- Regression blend-file:
Shared Layers

Suppose you have two objects: a pair of Pants on layers 2 and 3, and your Character Mesh on layers 1 and 2. You have enabled the Pants as Cloth as described above. You must now make the Character “visible” to the Cloth object, so that as your character bends its leg, it will push the cloth. This principle is the same for all Simulations; Simulations only interact with objects on a shared layer. In this example, both objects share layer 2.

- To view/change an object’s layers, RMB click to select the object in Object Mode in the 3D View.
- To bring up the “Move Layers” pop-up, which shows you all the layers that the object is on.
- To put the object on a single layer, LMB click the layer button.
- To put the object on multiple layers, Shift-LMB the layer buttons.
- To remove an object from a selected layer, simply Shift-LMB the layer button again to toggle it.

Mesh Objects Collide

If your colliding object is not a Mesh object, such as a NURBS surface, or text object, you must convert it to a Mesh object. To do so, select the object in object mode, and in the 3D View header, select Object > Convert Object Type Alt-C, and select Mesh from the pop-up menu.

Cloth - Object collisions
The Cloth object needs to be deflected by some other object.

- To deflect a cloth, the object must be enabled as an object that collides with the Cloth object.
- To enable Cloth - Object collisions, you must enable deflections on the collision object (not on the Cloth object).

In the Properties editor, Object tab and Physics tab, locate the Collision panel shown to the right. It is also important to note that this collision panel is used to tell all Simulations that this object is to participate in colliding/deflecting other objects on a shared layer (particles, soft bodies, and cloth).

There are three different Collision panels, all found in the Physics tab. The first (by default), a tab beside the Fields panel, is the one needed here. The second panel, a tab in the Soft Body group, concern soft bodies (and so has nothing to do with cloth). And we have already seen the last one, by default a tab beside the Cloth panel.

Mesh Object Modifier Stack
The object’s shape deforms the cloth, so the Cloth Simulation must know the “true” shape of that Mesh object at that frame. This true shape is the basis shape as modified by shape keys or armatures. Therefore, the Collision Modifier must be after any of those. The image to the right shows the Modifiers panel for the Character Mesh object (not the Cloth object).

**Cloth Cache**

Cache settings for Cloth are the same as with other dynamic systems.

**Bake Collision**
After you have set up the deflection Mesh for the frame range you intend to run the Simulation (including animating that Mesh via armatures), you can now tell the Cloth Simulation to compute (and avoid) collisions.

- Select the Cloth object
- In the Object tab, Physics tab, set the Start and End settings for the Simulation frames you wish to compute,
- Click the Bake button.

You cannot change Start or End without clearing the Bake Simulation. When the Simulation has finished, you will notice you have the option to free the bake, edit the bake and re-bake:

> There are a few things you will probably notice right away. First, it will bake significantly slower than before, and it will probably clip through the box pretty badly as in the picture on the right.

**Editing the Cached Simulation**

The cache contains the shape of the Mesh at each frame. You can edit the Cached Simulation, after you have baked the Simulation.
and pressed the **Bake Editing** button. Just go to the frame you want to fix and Tab into **Edit Mode**. There you can move your vertices using all of Blender’s Mesh shaping tools. When you exit, the shape of the Mesh will be recorded for that frame of the animation. If you want Blender to resume the Simulation using the new shape going forward, LMB click **Rebake from next Frame** and play the animation. Blender will then pick up with that shape and resume the Simulation.

Edit the Mesh to correct minor tears and places where the colliding object has punctured the cloth.

If you add, delete, extrude, or remove vertices in the Mesh, Blender will take the new Mesh as the starting shape of the Mesh back to the **first frame** of the animation, replacing the original shape you started with, up to the frame you were on when you edited the Mesh. Therefore, if you change the content of a Mesh, when you Tab out of **Edit Mode**, you should unprotect and clear the cache so that Blender will make a consistent Simulation.

**Troubleshooting**

If you encounter some problems with collision detection, there are two ways to fix them:

- The fastest solution is to increase the **Min Distance** setting under the **Cloth Collision** panel. This will be the fastest way to fix the clipping; however, it will be less accurate and will not look as good. Using this method tends to make it look like the Cloth is resting on air, and gives it a very rounded look.

- A second method is to increase the **Quality** (in the first **Cloth** panel). This results in smaller steps for the simulator and therefore to a higher probability that fast-moving collisions get caught. You can also increase the **Collision Quality** to perform more iterations to get collisions solved.

- If none of the methods help, you can easily edit the cached/baked result in **Edit Mode** afterwards.

- The Cloth is torn by the deforming Mesh – he “Hulks Out”: Increase its structural stiffness (**Structure Stiffness** setting, **Cloth** panel), very high, like 1000.
Subdivision Surface Modifier

A bake/cache is done for every subdivision level so please use the equal subdivision level for render and preview.

Examples

To start with cloth, the first thing you need, of course, is some fabric. So,

- Let us delete the default cube and add a plane.
- In order to get some good floppy and flexible fabric, you will need to subdivide it several times; about eight is a good number.
- So, Tab into Edit Mode, and press W - Subdivide multi, and set it to 8.
- Now, we will make this Cloth by going to the Physics tab.
- Scroll down until you see the Cloth panel, and press the Cloth button.
- Now, a lot of settings will appear, most of which we will ignore for now.
- That is all you need to do to set your Cloth up for animating, but if you press Alt-A, your lovely fabric will just drop very un-spectacularly.

That is what we will cover in the next two sections about Pinning and Colliding.

Using Simulation to Shape/Sculpt a Mesh

You can apply the Cloth Modifier at any point to freeze the Mesh in position at that frame. You can then re-enable the cloth, setting the start and end frames from which to run the Simulation forward.

Another example of Aging is a Flag.

- Define the flag as a simple grid shape and
- Pin the edge against the flagpole.
- Simulate for 50 frames or so, and the flag will drop to its “rest” position.
- Apply the Cloth Modifier.
If you want the flag to flap or otherwise move in the scene, re-enable it for the frame range when it is in camera view.

**Smoothing of Cloth**

Now, if you followed this from the previous section, your Cloth is probably looking a little blocky. In order to make it look nice and smooth like the picture you need to apply a *Smooth* and/or *Subdivision Surface* Modifier in the *Modifiers* tab. Then, in the same editor, find the *Links and Materials* panel (the same one you used for vertex groups) and press *Set Smooth*.

Now, if you **press Alt-A**, things are starting to look pretty nice, do not you think?

**Cloth on armature**

- Cloth deformed by armature and also, respecting an additional collision object: https://wiki.blender.org/index.php/Media:Cloth-regression-armature.blend

**Cloth with animated vertex groups**

- Cloth with animated pinned vertices: https://wiki.blender.org/index.php/Media:Cloth_animated_vertex.blend

**UNSUPPORTED**: Starting with a goal of 0 and increasing it, but still having the vertex not pinned will not work (e.g. from goal = 0 to goal = 0.5).

**Cloth with Dynamic Paint**

- Cloth with Dynamic Paint using animated vertex groups: https://wiki.blender.org/index.php/Media:Cloth_dynamic_paint.blend

**UNSUPPORTED**: Starting with a goal of 0 and increasing it, but still having the vertex not pinned will not work (e.g. from goal = 0 to goal = 0.5) because the necessary “goal springs” cannot be generated on the fly.

**Using Cloth for Soft bodies**
Title-Img. 3. 6Using Cloth for soft bodies.
Attribution-
Source-

Cloth can also be used to simulate soft bodies. It is for sure not its main purpose but it works nonetheless. The example image uses standard Rubber material, no fancy settings, just Alt-A.

- Blend file for the example image: https://wiki.blender.org/index.php/Media:Cloth-sb1.blend

Cloth with Wind

Title-Img. 3. 7Flag with wind applied.
Attribution-
Source-

- Regression blend-file for Cloth with wind and self-collisions (also the blend for the image above): https://wiki.blender.org/index.php/Media:Cloth-flag2.blend
Unit summary

In this Unit, you have learnt what is Cloth Simulation and how to

- Use the Simulation to create effects to 3D Characters, 3D Assets and 3D Environment.

- Bake and save Simulation on moving objects using Cache.

- Create Modifier stacks and smooth Cloth Simulation.

- Create different types of Cloth Simulation like cape, flag, curtain, banners and fabrics, its use with the help of 3D assets to effectively work on 3D scene.

After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating 3D Interface.

Assignment

- Create a Pole with a Flag and apply external forces like wind to simulate flag fluttering

- Use the key words “flag fluttering” on www.google.com to collect the reference video to build your scene.

Assessment

1. Describe Cloth Simulation
2. Explain Dynamic Mesh
3. Define Sewing Force
4. Define Self-Collision
5. Explain Cloth Pinning with the steps and illustration
6. Describe Smoothing of Cloth
7. List out the Properties of Clothing in Simulation.
Fill in the Blanks

1. The first thing you need when pinning Cloth is a __________

2. When animating or posing the character, you must begin from the _______ pose

3. Sewing Springs are ___________ that pull vertices in one part of a Cloth Mesh toward vertices in another part of the Cloth Mesh.

4. If your colliding object is not a Mesh object, you must convert it to a__________.

5. The cache contains the shape of the Mesh at __________

Resources

While studying this Unit, you can browse the following internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

Links to download 3D Files for practice


2. https://cloud.blender.org/training

3. wiki.blender.org

4. ia600207.us.archive.org

5. archive.org

6. www.blender.org

7. docs.blender.org
Unit 4

Welcome to Fluid Simulation

Introduction

Fluid Simulation refers to Computer Graphics techniques for generating realistic animations of fluids such as liquid, water and smoke. You would come across Fluid Simulation that are typically focused on emulating the qualitative visual behavior of a fluid, with less emphasis placed on rigorously correct physical results, although they often still rely on approximate solutions that govern real fluid physics. You will also learn that Fluid Simulation can be performed with different levels of complexity, ranging from time-consuming, and high-quality animations for films or visual effects, to simple and fast animations for real-time animations like computer games.

Outcomes

Upon completion of this unit you will be able to:

- Define and Use Fluid Simulation in your 3D project for creating extra effects
- Describe to control the flow of the fluid
- Demonstrate the use of physical properties of the fluid
- Use bake simulation to save performance
- Reuse Existing simulation

Terminology

Volume: The inside of the object is initialized as fluid all. This works only if the closed mesh.

Shell: It is initialized as a thin fluid layer of the surface of the mesh. This can also be used in the mesh open.

Viscosity: The “thickness” of the fluid and actually the force needed to move an object of a certain
surface area through it at a certain speed

**Inflow Velocity:** Speed of the fluid that is created inside of the object.

**How Materials Works**

Fluid physics are used to simulate physical properties of liquids especially water. While creating a scene in Blender, certain objects can be marked to participate in the **Fluid Simulation**. These can include but not limited to, being a fluid or as an obstacle. For a **Fluid Simulation**, you must have a domain to define the space that the simulation takes up. In the domain settings, you will be able to define the global simulation parameters (such as viscosity and gravity).

![Example of Fluid Simulation](https://docs.blender.org/manual/en/dev/physics/fluid/introduction.html)

**Title**-Img. 4. Example of Fluid Simulation.

**Attribution**-

**Source**-


**Workflow**

In general, you follow these steps:

1. First you want to set the simulation domain,

2. Next set the fluid source(s), and specify the physical properties.

3. In some cases, you may want to set other objects to Control the Flow of the fluid.

4. You can also depend on your scene add other objects related to the fluid, like: Obstacles, Particles floating on the fluid.

5. And lastly you must Bake the Simulation.
Baking is done on the Domain object!
When you calculate the Fluid Simulation, you bake the simulation on the domain object.

**For this reason:**
All the baking options are visible only when selecting the Domain Object.
Baking options are explained in the baking section of the Domain manual page.

**Common Options**

**Animated Mesh/Export**

Click this button if the network is animated (Example: Deformed by an armature, shape keys, or lattice). It can become very slow and is not necessary if the network’s position and rotation are animated. (i.e. only object transformations).

**Volume Initialization Type**

A common option among the different fluid types is **Volume Initialization**.

- **Volume**
  The inside of the object is initialized as fluid all. This works only if the closed mesh.

- **Shell**
  It is initialized as a thin fluid layer of the surface of the mesh. This can also be used in the mesh open.

- **Both**
  It is a state, such as the sum of the Volume and Shell. This also must be a closed mesh.
Fluid Domain

The Domain Object

The bounding box of the object serves as the boundary of the simulation. All fluid objects **must be in the domain**. Fluid objects outside the domain will not bake. No tiny droplets can move outside this domain; it’s as if the fluid is contained within the 3D space by invisible force fields. There can be only a single Fluid Simulation domain object in the scene.

The shape of the object does **not** matter because it will **always** be treated like a **box** (The lengths of the bounding box sides can be different). So, usually there will not be any reason to use another shape than a box. If you need obstacles or other boundaries than a box to interfere with the fluid flow, you need to insert additional obstacle objects **inside** the domain boundary.

This object will be **replaced by the fluid** during the simulation.

---

**Tip**

**Baking is done on the Domain object**

When you calculate the Fluid Simulation, you bake the simulation on the domain object. For this reason, all the baking options are visible only when selecting the Domain Object.
Options

Title-Img. 4. 3Fluid Domain Settings.
Attribution-
Source-
Link-

Resolution

- **Render resolution**

  The granularity at which the actual Fluid Simulation is performed. This is probably the most important setting for the simulation as it determines the amount of details in the fluid, the memory and disk usage as well as computational time.
The amount of required memory quickly increases: a resolution of \textbf{32 requires ca. 4MB, 64 requires ca. 30MB}, while \textbf{128 already needs more than 230MB}. Make sure to set the resolution low enough, depending on how much memory you have, to prevent Blender from crashing or freezing. Remember also that many operating systems limit the amount of memory that can be allocated by a single \textit{process}, such as Blender, even if the \textit{machine} contains much more than this. Find out what limitations apply to your machine.

\textbf{Resolution and Real-size of the Domain}

Be sure to set the resolution appropriate to the real-world size of the domain. If the domain is not cubic, the resolution will be taken for the longest side. The resolutions along the other sides will be reduced according to their lengths (therefore, a non-cubic domain will need less memory than a cubic one, resolutions being the same).

\begin{itemize}
  \item \textbf{Preview resolution}

    This is the resolution at which the preview surface meshes will be generated. So, it does not influence the actual simulation. Even if “there is nothing to see” in the preview, there might be a thin fluid surface that cannot be resolved in the preview.

  \item \textbf{Display quality}

    How to display a baked simulation in the \textbf{3D View} (menu \textit{Viewport Display}) and for \textbf{rendering} (menu \textit{Render Display}):
- **Geometry**
  Use the original geometry (before simulation).

- **Preview**
  Use the preview mesh.

- **Final**
  Use the final high definition mesh.

When no baked data is found, the original mesh will be displayed by default.

After you have baked a domain, it is displayed (usually) in the Blender window as the preview mesh. To see the size and scope of the original domain box, select *Geometry* in the left selector.

**Time**

- **Start**
  It is the simulation start time (in seconds).

  This option makes the simulation computation in Blender start later in the simulation. The domain deformations and fluid flow prior to the start time are not saved.

  For example, if you wanted the fluid to appear to already have been flowing for 4 seconds before the actual first frame of data, you would enter 4.0 here.

- **End**
  It is the simulation ending time (in seconds).

> **Tip**

Start and end times have nothing to do with how many frames are baked

If you set **Start time** to 3.0, and **End time** to 4.0, you will simulate 1 second of fluid motion. That one second of fluid motion will be spread across however-many frames are set in **Render ▶ Dimensions**.

This means, for example, that if you have Blender set to make 250 frames at 25 fps, the fluid will look like it had already been flowing for 3 seconds at the start of the simulation, but will play in slow motion.
(one-tenth normal speed), since the 1 second fluid simulation plays out over the course of 10 video seconds. To correct this, change the end time to 13.0 (3.0 + 10.0) to match the 250 frames at 25 fps. Now, the simulation will be real-time, since you set 10 seconds of fluid motion to simulate over 10 seconds of animation. Having these controls in effect gives you a “speed control” over the simulation.

Generate Speed Vector

If this button is clicked, no speed vectors will be exported. So, by default, speed vectors are generated and stored on disk. They can be used to compute image based motion blur with the compositing nodes.

Reverse fluid frames

The simulation is calculated backward

Fluid World

Viscosity Presets

The “thickness” of the fluid and actually the force needed to move an object of a certain surface area through it at a certain speed.

For manual entry, please note that the normal real-world viscosity (the so-called Dynamic Viscosity) is measured in Pascal-seconds
(Pa.s), or in Poise units (P, equal to 0.1 Pa.s, pronounced pwaz, from the Frenchman Jean-Louis Poiseuille, who discovered the laws on “the laminar flow of viscous fluids”), and commonly centiPoise units (cP, equal to 0.001 Pa.s, sentipwaz). Blender, on the other hand, uses the kinematic viscosity (which is dynamic viscosity in Pa.s, divided by the density in kg.m⁻³, unit m².s⁻¹). The table below gives some examples of fluids together with their dynamic and kinematic viscosities.

### Blender Viscosity Unit Conversion

<table>
<thead>
<tr>
<th>Fluid</th>
<th>Dynamic Viscosity (in cP)</th>
<th>Kinematic Viscosity (Blender, in m².s⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water (20°C)</td>
<td>1.002×10⁰ (1.002)</td>
<td>1.002×10⁻⁶ (0.000001002)</td>
</tr>
<tr>
<td>Oil SAE 50</td>
<td>5.0×10² (500)</td>
<td>5.0×10⁻⁵ (0.00005)</td>
</tr>
<tr>
<td>Honey (20°C)</td>
<td>1.0×10⁴ (10,000)</td>
<td>2.0×10⁻³ (0.002)</td>
</tr>
<tr>
<td>Chocolate Syrup</td>
<td>3.0×10⁴ (30,000)</td>
<td>3.0×10⁻³ (0.003)</td>
</tr>
<tr>
<td>Ketchup</td>
<td>1.0×10⁵ (100,000)</td>
<td>1.0×10⁻¹ (0.1)</td>
</tr>
<tr>
<td>Melting Glass</td>
<td>1.0×10¹⁵</td>
<td>1.0×10⁰ (1.0)</td>
</tr>
</tbody>
</table>

Manual entries are specified by a floating-point number and an exponent. These floating point and exponent entry fields (scientific notation) simplify entering very small or large numbers. The viscosity of water at room temperature is 1.002 cP, ou 0.001002 Pa.s; the density of water is about 1000 kg.m⁻³, which gives a kinematic viscosity of 0.000001002 m².s⁻¹ — so the entry would be 1.002 times 10 to the minus six (1.002×10⁻⁶ in scientific notation). Hot Glass and melting iron is a fluid, but very thick; you
should enter something like $1.0 \times 10^0$ ($= 1.0$) as its kinematic viscosity (indicating a value of $1.0 \times 10^6$ cP).

Note that the simulator is not suitable for non-fluids, such as materials that do not “flow”. Simply setting the viscosity to very large values will not result in rigid body behavior, but might cause instabilities.

**Tip**

Viscosity varies

The default values in Blender are considered typical for those types of fluids and “look right” when animated. However, actual viscosity of some fluids, especially sugar-laden fluids like chocolate syrup and honey, depend highly on temperature and concentration. Oil viscosity varies by SAE rating. Glass at room temperature is basically a solid, but glass at 1500 degrees Celsius flows (nearly) like water.

**Real World Size**

Size of the domain object in the real world in meters. If you want to create a mug of coffee, this might be 10 cm (0.1 meters), while a swimming pool might be 10m. The size set here is for the longest side of the domain bounding box.

**Optimization**

- **Grid level**
  How many adaptive grid levels to be used during simulation. Setting this to -1 will perform automatic selection.

- **Compressibility**
  If you have problems with large standing fluid regions at high resolution, it might help to reduce this number (note that this will increase computation times).
Fluid Boundary

Title- Img. 4. Fluid Boundary panel.
Attribution-
Source-

This box has all the slip and surface options.

Boundary type

The stickiness of the surface of the obstacle, to determine the “tacky surface (Surface Adhesion).” In the real world, and the tackiness and fluid, the granularity of the object surface, tack, determined by the elasticity.

- **No Slip**
  Fluid will stick to snugly (speed 0).

- **Free Slip**
  Fluid will move on the object (0 normal direction of speed).

- **Part Slip**
  It is a two intermediate. It is almost No slip, 1 in the Free exactly the same in 0.

Surface

- **Surface Smoothing**
  Amount of smoothing to be applied to the fluid surface. 1.0 is standard, 0 is off, while larger values increase the amount of smoothing.

- **Subdivisions**
Allows the creation of high-res surface meshes directly during the simulation (as opposed to doing it afterwards like a subdivision modifier). A value of 1 means no subdivision, and each increase results in one further subdivision of each fluid voxel. The resulting meshes thus quickly become large, and can require large amounts of disk space. Be careful in combination with large smoothing values – this can lead to long computation times due to the surface mesh generation.

Fluid Particles

Here you can add particles to the fluid simulated, to enhance the visual effect.

- **Tracer Particles**
  Number of tracer particles to be put into the fluid at the beginning of the simulation. To display them create another object with the Particle fluid type, explained below, that uses the same bake directory as the domain.

- **Generate Particles**
  Controls the amount of fluid particles to create (0=off, 1=normal, >1=more). To use it, you have to have a surface subdivision value of at least 2.
Title: Img. 4. 7 An example of Particles' effects.
Attribution: 
Source: 
Link: 

Left: Simulated without; Right: With particles and subdivision enabled
Fluid Object

All regions of this object that are inside the domain bounding box will be used as actual fluid in the simulation. If you place more than one fluid object inside the domain, they should currently not intersect. Also make sure the surface normals are pointing outwards. In contrast to domain objects, the actual mesh geometry is used for fluid objects.

- **Volume Initialization Type**
  Refer [Volume Initialization Type](https://docs.blender.org/manual/en/dev/physics/fluid/types/fluid_object.html)

- **Animated Mesh/Export**

- **Initial velocity**
  Speed of the fluid at the beginning of the simulation, in meters per second.

The direction of Surface Normals makes a big difference!

Blender uses the orientation of the Surface Normals to determine what is “inside of” the Fluid object and what is “outside”. You want all of the normals to face **outside** (in *Edit Mode*, use Ctrl-Nor press Spacebar and choose *Edit?? Normals?? Calculate Outside*). If the normals face the wrong way, you will be rewarded with a “gigantic flood of water”
because Blender will think that the volume of the object is outside of its mesh! This applies regardless of the *Volume init* type setting.

**Fluid Obstacle**

This object will be used as an obstacle in the simulation. As with a fluid object, obstacle objects currently should not intersect. As for fluid objects, the actual mesh geometry is used for obstacles. For objects with a volume, make sure that the normals of the obstacle are calculated correctly, and radiating properly (use the *Flip Normal* button, in *Edit Mode*, *Mesh Tools* panel, in the Tool shelf), particularly when using a spinned container. Applying a Subdivision Surface Modifier before baking the simulation could also be a good idea if the mesh is not animated.

- **Volume Initialization Type**
  
  Refer [Volume Initialization Type](#)

- **Boundary type**

  Determines the stickiness of the obstacle surface, called “Surface Adhesion”. Surface Adhesion depends in real-world on the fluid and the graininess or friction/adhesion/absorption qualities of the surface.

  - **No Slip**
    
    Causes the fluid to stick to the obstacle (zero velocity).

  - **Free Slip**

    Allows movement along the obstacle (only zero normal velocity).

  - **Part Slip**

    Mixes both types, with 0 being mostly no slip, and 1 being identical to free slip.

Note that if the mesh is moving, it will be treated as no slip automatically.
Title-Img. 4. 9 Example of the different boundary types for a drop falling onto the slanted wall. From left to right: no-slip, part-slip 0.3, part-slip 0.7 and free-slip.

Attribution-
Source-

- Animated Mesh/Export
  Refer Animated Mesh/Export

- Part Slip Amount
  Amount of mixing between no- and free-slip, described above.

- Impact Factor
  Amount of fluid volume correction for gain/loss from impacting with moving objects. If this object is not moving, this setting has no effect. However, it if is and the fluid collides with it, a negative value takes volume away from the Domain, and a positive number adds to it. Ranges from -2.0 to 10.0.

Fluid Inflow / Outflow

To control the volume of the Fluid Simulation, you can set objects in the scene to add or absorb fluid within the Fluid Domain.

Inflow
Title - Img. 4. 10 Fluid Inflow Settings.

Attribution -

Source -


- **Volume Initialization Type**
  Refer [Volume Initialization Type](https://docs.blender.org/manual/en/dev/physics/fluid/types/flow.html)

This object will put fluid into the simulation, like a **water tap**.

- **Inflow Velocity**
  Speed of the fluid that is created inside of the object.

- **Local Coordinates/Enable**
  Use local coordinates for the inflow. This is useful if the inflow object is moving or rotating, as the inflow stream will follow/copy that motion. If disabled, the inflow location and direction do not change.

- **Animated Mesh/Export**

**Outflow**

Title - Img. 4. 11 Fluid Outflow Settings.

Attribution -
Any fluid that enters the region of this object will be deleted (think of a drain or a black hole). This can be useful in combination with an inflow to prevent the whole domain from filling up. When enabled, this is like a **tornado** (waterspout) or “**wet vac**” vacuum cleaner, and the part where the fluid disappears will follow the object as it moves around.

- **Volume Initialization Type**
  Refer [Volume Initialization Type](https://docs.blender.org/manual/en/dev/physics/fluid/types/flow.html)

- **Animated Mesh/Export**

**Fluid Particle**

This type can be used to display particles created during the simulation. For now, only tracers swimming along with the fluid are supported.

Note that the object can have any shape, position or type. Once the particle button is pressed, a particle system with the Fluid Simulation particles will be created for it at the correct position. When moving the original object, it might be necessary to delete the particle system, disable the fluidism particles, and enable...
them again. The fluidism particles are currently also unaffected by any other particle forces or settings.

- **Influence**
  - **Size Influence**
  
  The particles can have different sizes, if this value is 0 all are forced to be the same size.

  - **Alpha Influence**

  If this value is >0, the alpha values of the particles are changed according to their size.

- **Particle type**
- **Drops**

  Surface splashes of the fluid result in droplets being strewn about, like fresh water, with low Surface Tension.

  - **Floats**

  The surface tension of the fluid is higher and the fluid heavier, like cold seawater and soup. Breakaways are clumpier and fall back to the surface faster than *Drops*, as with high Surface Tension.

  - **Tracer**

  Droplets follow the surface of the water where it existed, like a fog suspended above previous fluid levels. Use this to see where the fluid level has been.

- **Path (bake directory)**

  The simulation runs to load the particles. This should usually have the same value as the fluid domain object (e.g. copy by Ctrl-C, Ctrl-V).

### Fluid Control

Using the **Lattice-boltzman method**, the fluid is controlled using particles which define local force fields and are generated automatically from either a physical simulation or a sequence of target shapes. At the same time, as much as possible of the natural fluid motion is preserved.
### Options

<table>
<thead>
<tr>
<th>Fluid Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enabled</td>
<td>Controls whether the control object contributes to the fluid system. This is useful when animating the fluid control object.</td>
</tr>
<tr>
<td>Quality</td>
<td>Higher quality results in more control particles for the fluid control object.</td>
</tr>
<tr>
<td>Reverse Frames</td>
<td>The control particle movement gets reversed.</td>
</tr>
<tr>
<td>Attraction Force</td>
<td>The attraction force specifies the force which gets emitted by the fluid control object. Positive force results in attraction of the fluid, negative force in avoidance.</td>
</tr>
<tr>
<td>Velocity Force</td>
<td></td>
</tr>
</tbody>
</table>

**Enabled**

Controls weather the control object contributes to the fluid system. This is useful when animating the fluid control object.

**Quality**

Higher quality results in more control particles for the fluid control object.

**Reverse Frames**

The control particle movement gets reversed.

**Time**

You specify the start and end time during which time the fluid control object is active.

**Attraction force**

The attraction force specifies the force which gets emitted by the fluid control object. Positive force results in attraction of the fluid, negative force in avoidance.

**Velocity force**
If the fluid control object moves, the resulting velocity can also introduce a force to the fluid.

Examples
In this example, we use the Fluid Control option to control part of the fluid so that it has a certain shape (the sphere drop or the teapot drop) before it falls in the rest of the fluid:

Title-Img. 4. 14Falling drop.
Attribution-
Source-
Link-

Title-Img. 4. 15“Magic Fluid Control.”
Attribution-
Source-
Link-
Baking

Title-Img. 4. 16 The Fluid Simulation options with Domain selected.
Attribution-
Source-

Bake Button

- Perform the actual Fluid Simulation.

Blender will continue to work normally, except there will be a progress bar in the header of the Info Editor, next to the render pulldown. Pressing Esc or the “x” next to the status bar will abort the simulation. Afterwards two .bobj.gz (one for the Final quality, one for the Preview quality), plus one .bvel.gz (for the Final quality) will be in the selected output directory for each frame.

Bake directory

- Directory and file prefix to store baked surface meshes.

This is similar to the animation output settings, only selecting a file is a bit special: when you select any of the previously generated surface meshes (e.g. test1_fluidsurface_final_0132.bobj.gz), the prefix will be automatically set (test1_ in this example). This way the simulation can be done several times with different settings, and allows quick changes between the different sets of surface data.

Notes
• **Unique domain**

Because of the possibility of spanning and linking between scenes, there can only be one domain in an entire blend-file.

• **Selecting a Baked Domain**

After a domain, has been baked, it changes to the fluid mesh. To re-select the domain so that you can bake it again after you have made changes, go to any frame and select RMB the fluid mesh. Then you can click the *bake* button again to recomputed the fluid flows inside that domain.

• **Baking always starts at Frame #1**

The fluid simulator disregards the Start setting in the Animation panel, it will always bake from frame 1. If you wish the simulation to start later than frame 1, you must key the fluid objects in your domain to be inactive until the frame you desire to start the simulation.

• **Baking always ends at the End Frame set in the Animation panel**

If your frame-rate is 25 frames per second, and ending time is 4.0 seconds, then you should (if your start time is 0) set your animation to end at frame $4.0 \times 25 = 100$

• **Freeing the previous baked solutions**

Deleting the content of the “Bake” directory is a destructive way to achieve this. Be careful if more than one simulation uses the same bake directory (be sure they use different filenames, or they will overwrite one another)!

• **Reusing Bakes**

Manually entering (or searching for) a previously saved (baked) computational directory and filename mask will switch the fluid flow and mesh deformation to use that which existed during the old bake. Thus, you can re-use baked flows by simply pointing to them in this field.

• **Baking processing time**
Baking takes a lot of compute power (hence time). Depending on the scene, it might be preferable to bake overnight.

If the mesh has modifiers, the rendering settings are used for exporting the mesh to the fluid solver. Depending on the setting, calculation times and memory use might exponentially increase. For example, when using a moving mesh with Subdivision Surface as an obstacle, it might help to decrease simulation time by switching it off, or to a low subdivision level. When the setup/rig is correct, you can always increase settings to yield a more realistic result.

**Fluid Appendix**

**Hints**

Some useful hints about Fluid Simulation in Blender:

- Do not be surprised, but you will get whole bunch of mesh (.bobj.gz) files after a simulation. One set for preview, and another for final. Each set has a .gz file for each frame of the animation. Each file contains the simulation result – so you will need them.
- Currently these files will not be automatically deleted, so it is a good idea to e.g. create a dedicated directory to keep simulation results. Doing a Fluid Simulation is similar to clicking the animation button. Currently you must take care of organizing the fluid surface meshes in some directory by yourself. If you want to stop using the Fluid Simulation, you can simply delete all the *fluid*.bobj.gz files.
- Before running a high-resolution simulation that might take hours, check the overall timing first by doing lower resolution runs.
- Fluid objects must be completely inside the bounding box of the domain object. If not, baking may not work correctly or at all. Fluid and obstacle objects can be meshed with complex geometries. Very thin objects might not appear in the simulation, if the chosen resolution is too coarse to resolve them (increasing it might solve this problem).
- Do not try to do a complicated scene all at once. Blender has a powerful compositor that you can use to combine multiple animations.
• For example, to produce an animation showing two separate fluid flows while keeping your domain small, render one .avi using the one flow. Then move the domain and render another .avi with the other flow using an alpha channel (in a separate B&W .avi?). Then, composite both .avi’s using the compositor’s add function. A third .avi is usually the smoke and mist and it is laid on top of everything as well. Add a rain sheet on top of the mist and spray and you will have quite a storm brewing! And then lightning flashes, trash blowing around, all as separate animations, compositing the total for a truly spectacular result.

Limitations & Workarounds

• If the setup seems to go wrong, make sure all the normals are correct (hence, enter Edit Mode, select all, and recalculate normals once in a while).

• Currently there is a problem with zero gravity simulation. It could be avoided by simply selecting a very small gravity until this is fixed.

• If an object is initialized as Volume, it must be closed and have an inner side (a plane will not work). To use planes, switch to Shell, or extrude the plane.

• Blender freezes after clicking bake. Pressing Esc makes it work again after a while – this can happen if the resolution is too high and memory is swapped to hard disk, making everything horribly slow. Reducing the resolution should help in this case.

• Blender crashes after clicking bake – this can happen if the resolution is really high and more than 2GB are allocated, causing Blender to crash. Reduce the resolution. Many operating systems limit the total amount of memory that can be allocated by a process, such as Blender, even if the machine has more memory installed.

• The meshes should be closed, so if some parts of e.g. a fluid object are not initialized as fluid in the simulation, check that all parts of connected vertices are closed.
meshes. Unfortunately, the **Suzanne (monkey)** mesh in Blender is **not a closed mesh** (the eyes are separate).

- If the Fluid Simulation exits with an error message (stating e.g. that the “init has failed”), make sure you have valid settings for the domain object, e.g. by resetting them to the defaults.

- Note that first frame may well take only a few hundred MBs of RAM memory, but latter ones go **over one GB**, which may be why your bake fails after a while. If so, try to bake one frame at the middle or end at full res so you will see if it works.

- Memory used doubles when you set surface subdivision from **1 to 2**.

- Using “generate particles” will also add memory requirements, as they increase surface area and complexity. Ordinary fluid-sim generated particles probably eat less memory
In this Unit, you have learnt about Fluid Simulation and how to

- Include the Fluid Simulation in your 3D Scene, using the global simulation parameter.
- Create simulation for water, liquid, oil, flood, ocean etc.
- Control Fluid Simulation using the physical properties of those elements
- Interact with the collusion or obstacle objects.

After learning this Unit, you can download the Open Source Software available on the internet for free of cost to practice the possibilities of creating 3D Interface.

Assignment

- Create a Simple Fluid Simulation of pouring liquid from a tea pot to a cup.
- Use the key words “Teapot Pouring Tea” on www.google.com to collect the reference video to build your scene.
Assessment

1. Describe Domain Object
2. Define Speed Vector
3. Explain Viscosity
4. Describe Fluid Object
5. Explain Fluid Particles
6. Define Baking

State the limitations of Fluid Simulation.

Fill in the Blanks

1. _______ are used to simulate physical properties of liquids especially water.

2. Speed Vectors can be used to compute image based _______ with the compositing nodes.

3. The thickness of the fluid is called ____________

4. _______ Controls the movement of the liquid.

5. _______ is a thin fluid layer of the surface of the mesh.

Resources

While studying this Unit, you can browse the following internet links for online video tutorials and several books and training DVDs available in the Blender Store and on the Blender Cloud.

- wiki.blender.org
- ia600207.us.archive.org
- archive.org
- www.blender.org
links to download 3d files for practice

1. [https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Particles_forming_Shares](https://en.wikibooks.org/wiki/Blender_3D:_Noob_to_Pro/Particles_forming_Shares)

2. [https://cloud.blender.org/training](https://cloud.blender.org/training)