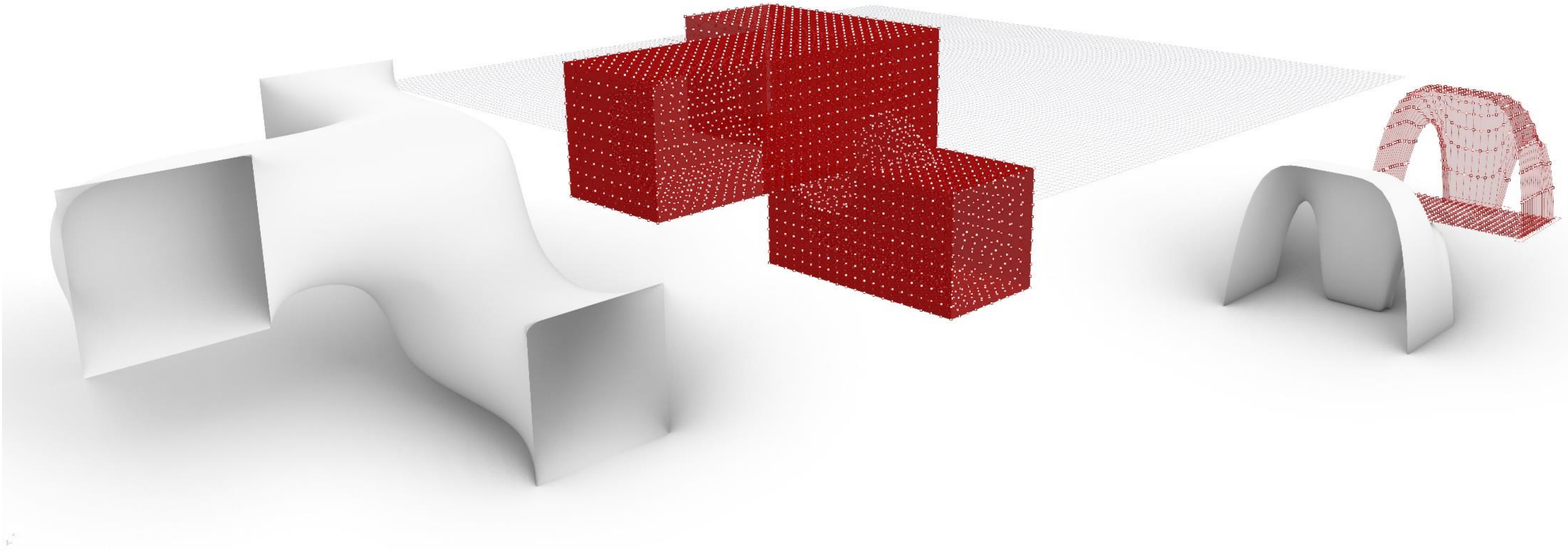
Dynamic Relaxation & Force-Directed Graph Drawing

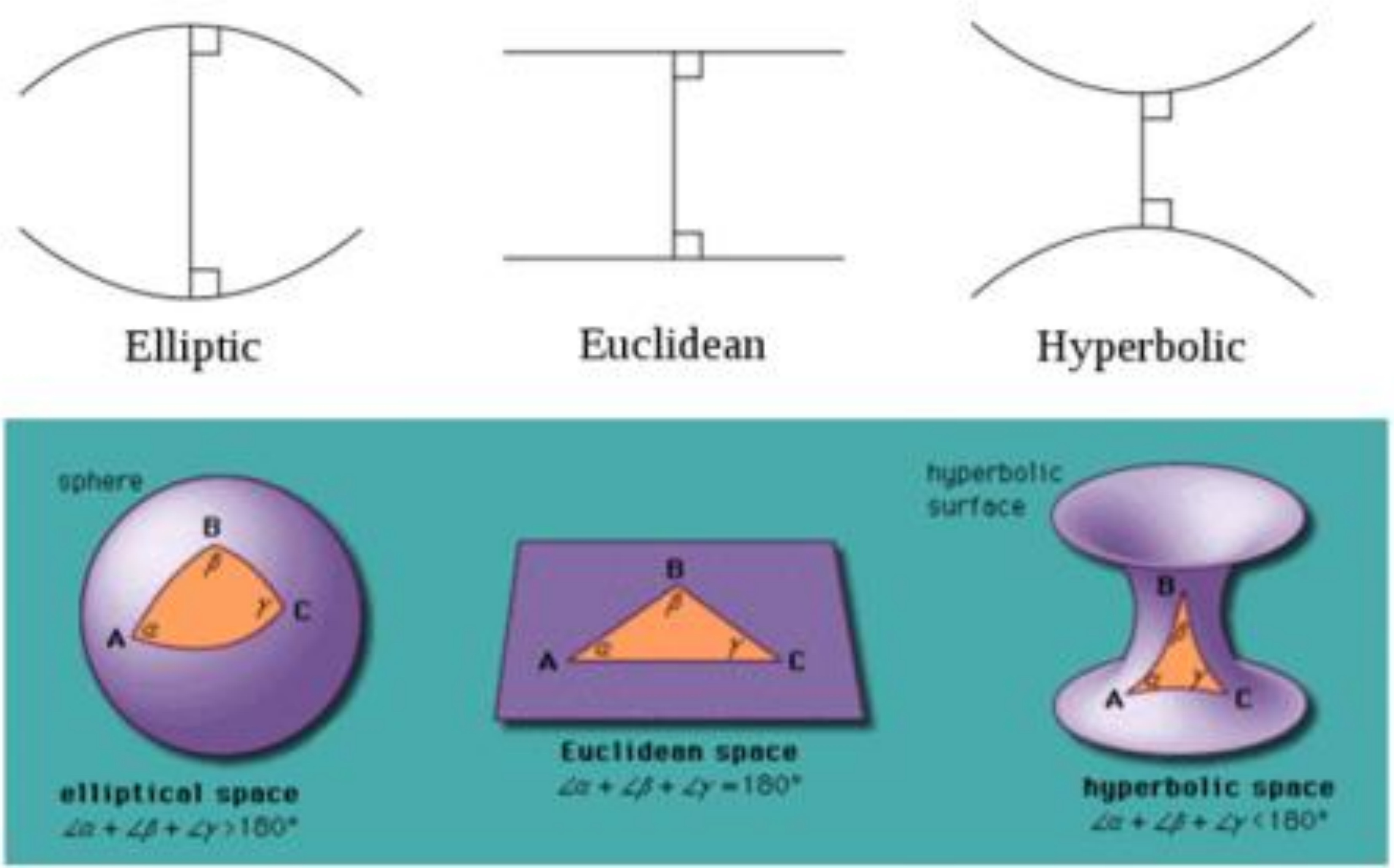


Dr. Ir. Pirouz Nourian, Ir. Shervin Azadi

https://github.com/Pirouz-Nourian/Dynamic_Relaxation
Chair of Design Informatics

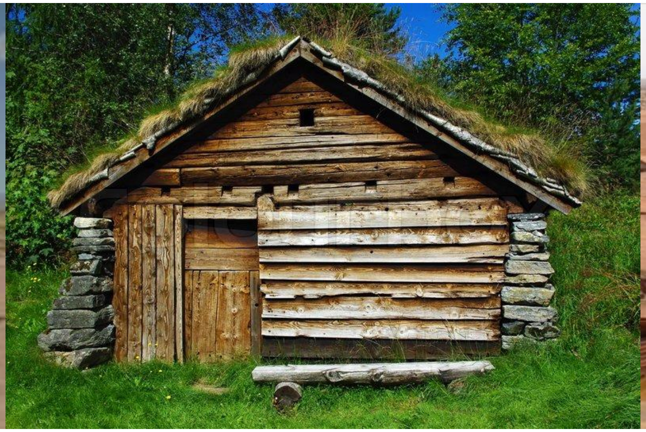
TU Delft

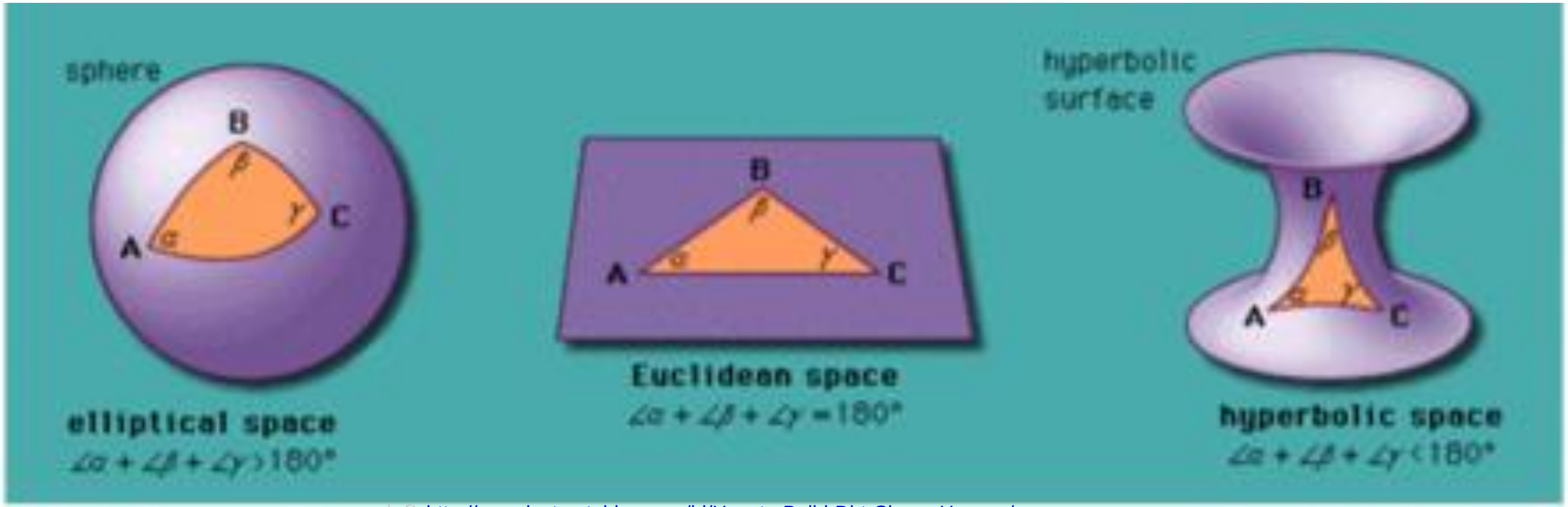
Material-Form-Structure



http://original.britannica.com/eb/art-322/Contrasting-triangles-in-Euclidean-elliptic-and-hyperbolic-spaces

Material-Form-Structure



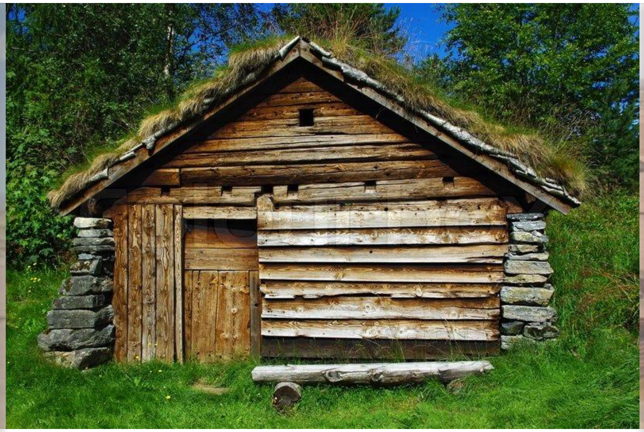
left http://www.instructables.com/id/How-to-Build-Dirt-Cheap-Houses/

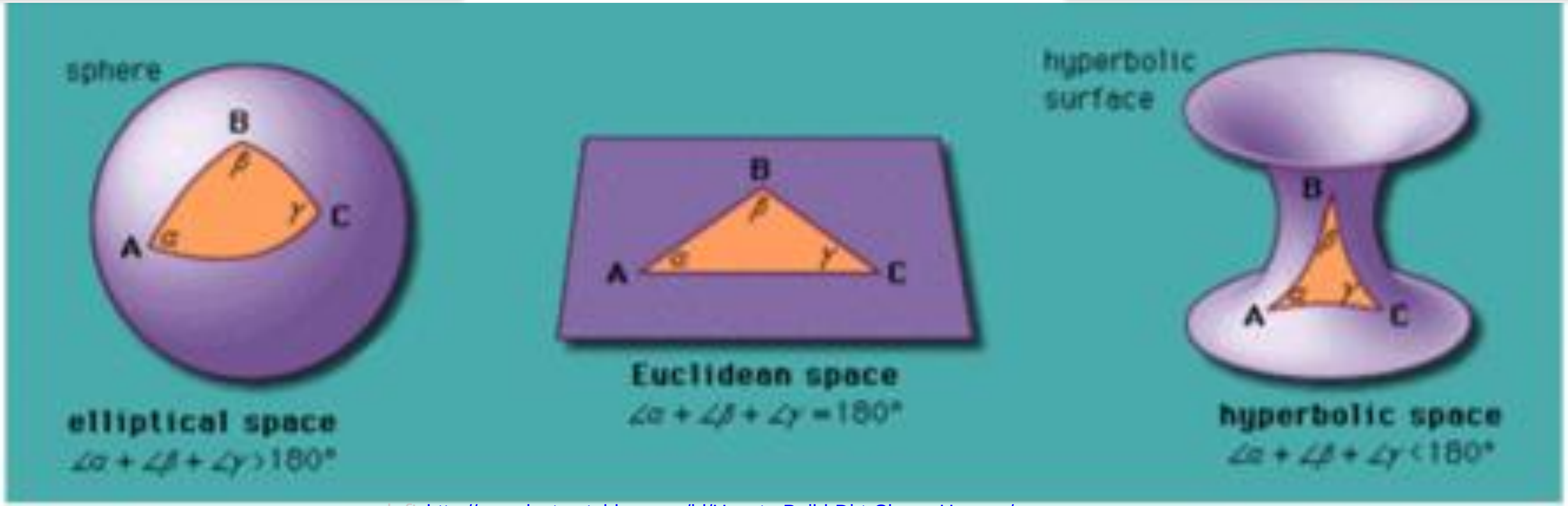
Middle: https://www.colourbox.com/image/ancient-fisherman-s-wooden-hut-in-ethnic-park-of-alesund-norway-image-1723627

Right: https://www.colourbox.com/image/ancient-fisherman-s-wooden-hut-in-ethnic-park-of-alesund-norway-image-1723627

bottom: http://original.britannica.com/eb/art-322/Contrasting-triangles-in-Euclidean-elliptic-and-hyperbolic-spaces

Material-Form-Structure





left http://www.instructables.com/id/How-to-Build-Dirt-Cheap-Houses/

Middle: https://www.colourbox.com/image/ancient-fisherman-s-wooden-hut-in-ethnic-park-of-alesund-norway-image-1723627

Right: https://www.colourbox.com/image/ancient-fisherman-s-wooden-hut-in-ethnic-park-of-alesund-norway-image-1723627

bottom: http://original.britannica.com/eb/art-322/Contrasting-triangles-in-Euclidean-elliptic-and-hyperbolic-spaces

Dynamic Relaxation

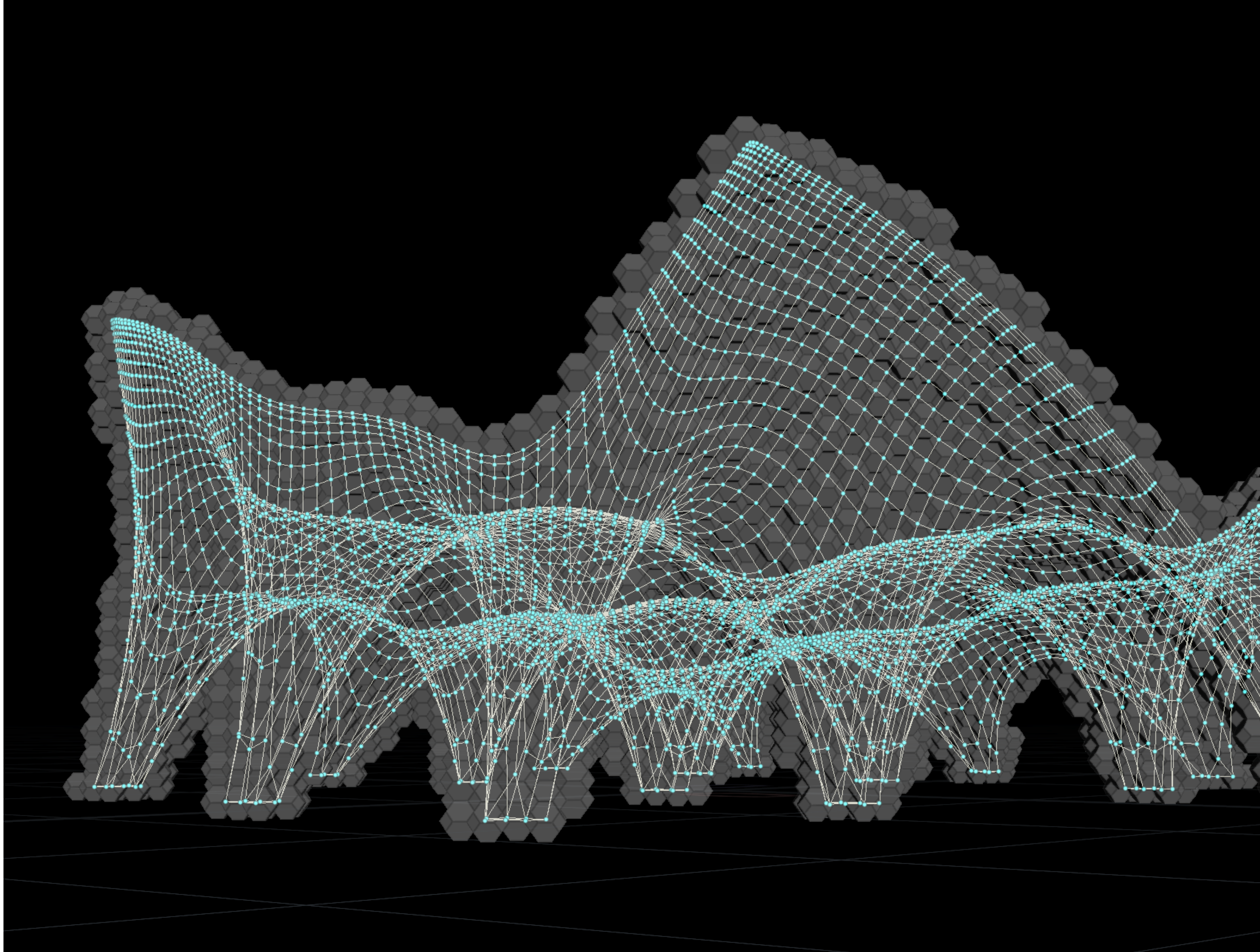
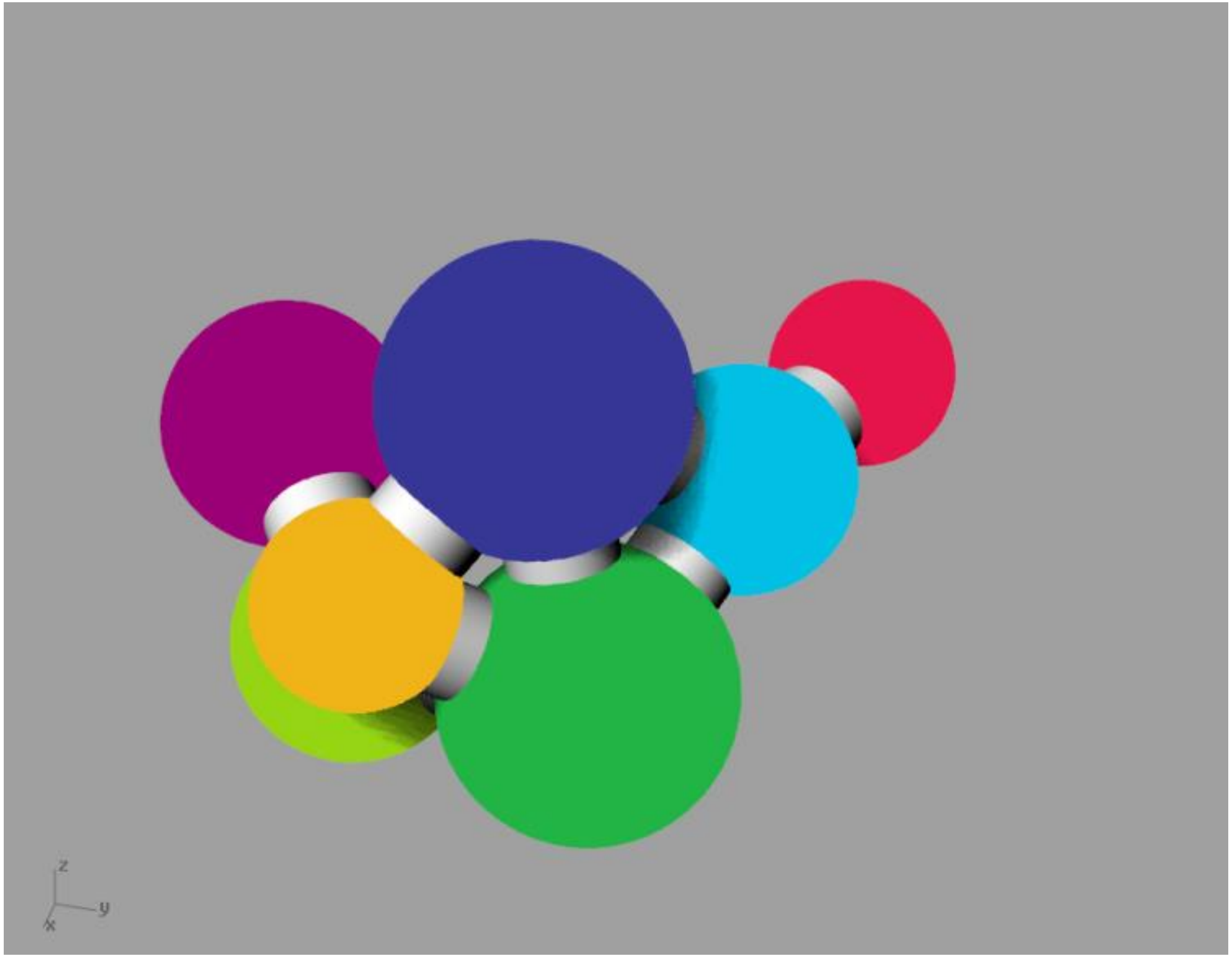


Image Credits: Earthy Honeycombs, Karim Daw, Shervin Azadi, Pirouz Nourian, Hans Hoogenboom

TU Delft

Force-Directed Graph-Drawing



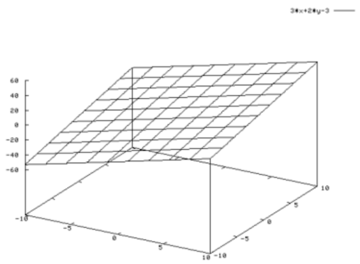
Minimal Surfaces

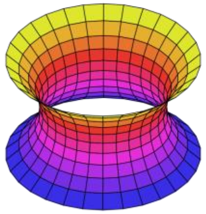
- A surface that minimizes total area subject to some constraint

TU Delft

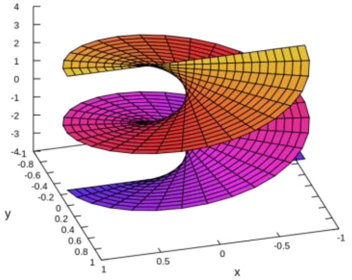
Minimal Surfaces

Classic



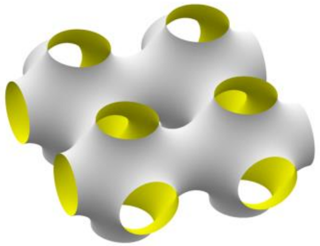


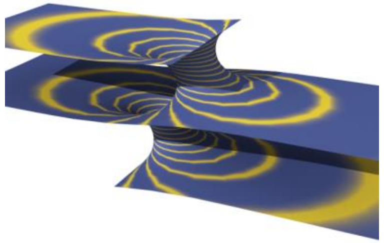
Catenoid



Helicoid

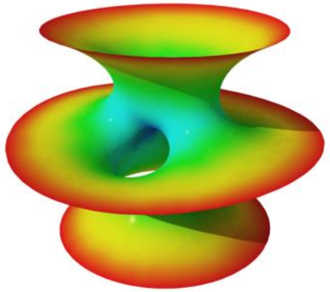
19th Century

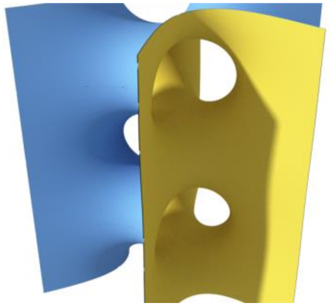




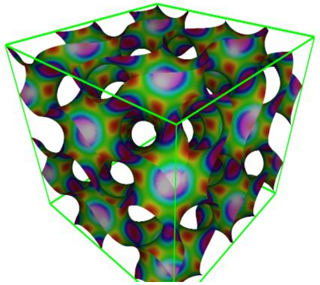
Riemann

Contemporary





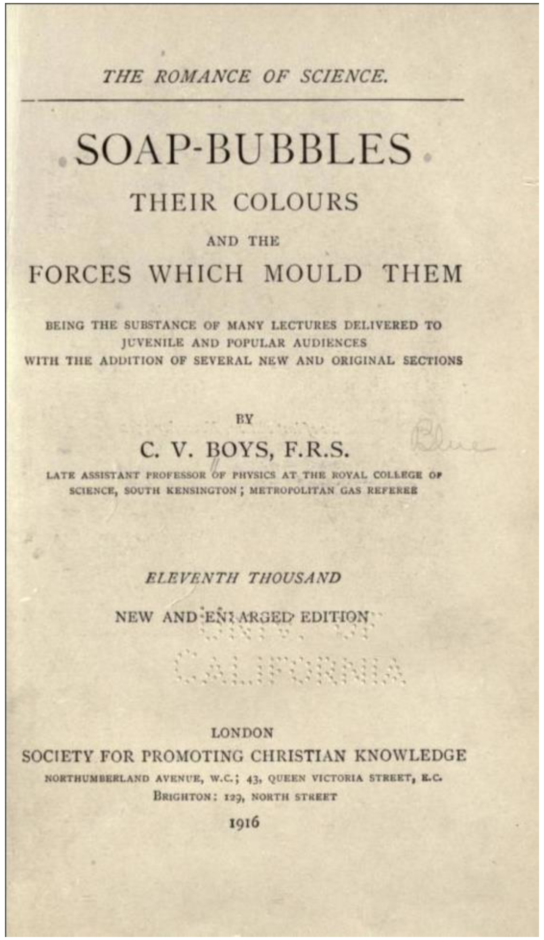
Saddle Tower



Gyroid

WikiPedia

Minimal Surfaces



THE ROMANCE OF SCIENCE.

SOAP-BUBBLES

THEIR COLOURS

AND THE

FORCES WHICH MOULD THEM

BEING THE SUBSTANCE OF MANY LECTURES DELIVERED TO
JUVENILE AND POPULAR AUDIENCES
WITH THE ADDITION OF SEVERAL NEW AND ORIGINAL SECTIONS

BY

C. V. BOYS, F.R.S.

LATE ASSISTANT PROFESSOR OF PHYSICS AT THE ROYAL COLLEGE OF
SCIENCE, SOUTH KENSINGTON; METROPOLITAN GAS REFEREE

ELEVENTH THOUSAND

NEW AND ENLARGED EDITION

LONDON
SOCIETY FOR PROMOTING CHRISTIAN KNOWLEDGE
NORTHUMBERLAND AVENUE, W.C.; 43, QUEEN VICTORIA STREET, E.C.
BRIGHTON: 129, NORTH STREET
1916

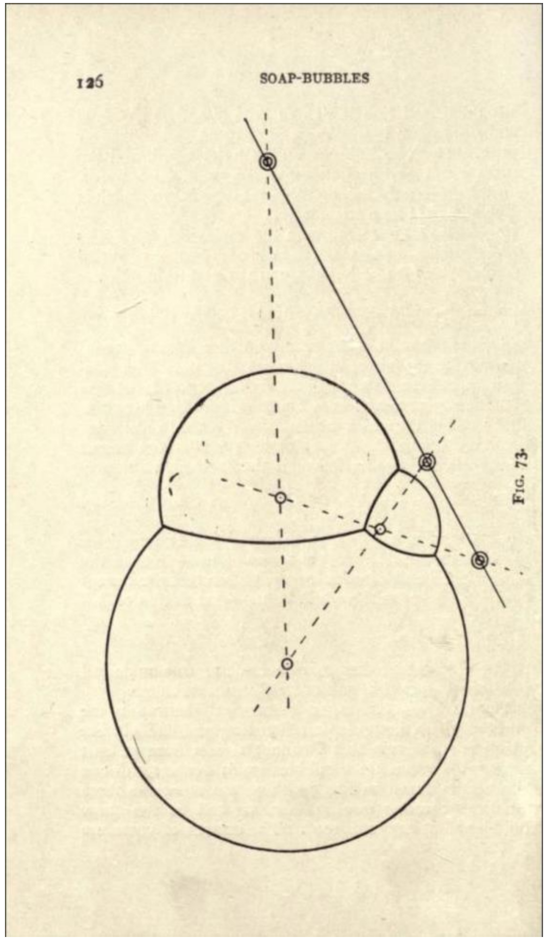


FIG. 74.

Minimal Surfaces

It is easy to show that this is heavy; it is only necessary to drop into the jar a bubble, and so soon as the bubble meets the heavy vapour it stops falling and remains floating upon the surface as a cork does upon water (Fig. 53). Now let me test the bubble and see whether any of the vapour has passed to the inside. I pick it up

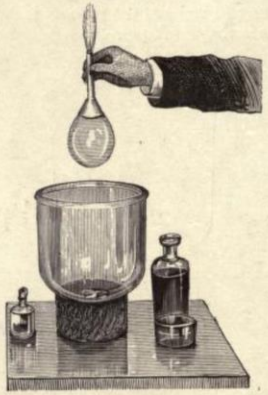


FIG. 54.

out of the jar with a wire ring and carry it to a light, and at once there is a burst of flame. But this is not sufficient to show that the ether vapour has passed to the inside, because it might have condensed in sufficient quantity upon the bubble to make it inflammable. You remember that when I poured some of this vapour upon water (see p. 34), sufficient condensed to so weaken the water-skin that the frame of wire could get

through to the other side. However, I can see whether this is the true explanation or not by blowing a bubble on a wide pipe, and holding it in the vapour for a moment. Now on removing it you notice that the bubble hangs like a heavy drop; it has lost the perfect roundness that it had at first, and this looks as if the vapour had found its way in, but this is made certain by bringing a light to the mouth of the tube, when the vapour, forced out by the elasticity of the bubble, catches fire and burns with a flame five or six inches long (Fig. 54). You might also have noticed that when the bubble was re-

FIG. 55.

moved, the vapour inside it began to pass out again and fell away in a heavy stream, but this you could only see by looking at the shadow upon the screen.

Experiments with Soap-bubbles

You may have noticed when I made the drops of oil in the mixture of alcohol and water, that when they were brought together they did not at once unite; they pressed against one another and pushed each other away if allowed, just as the water-drops did in the fountain of which I showed you a photograph. You also may have noticed that the drops of water in the paraffin

mixture bounced against one another, or if filled with the paraffin, formed bubbles in which often other small drops, both of water and paraffin, remained floating.

In all these cases there was a thin film of something between the drops which they were unable to squeeze out, namely, water, paraffin, or air, as the case might be.

FIG. 56.

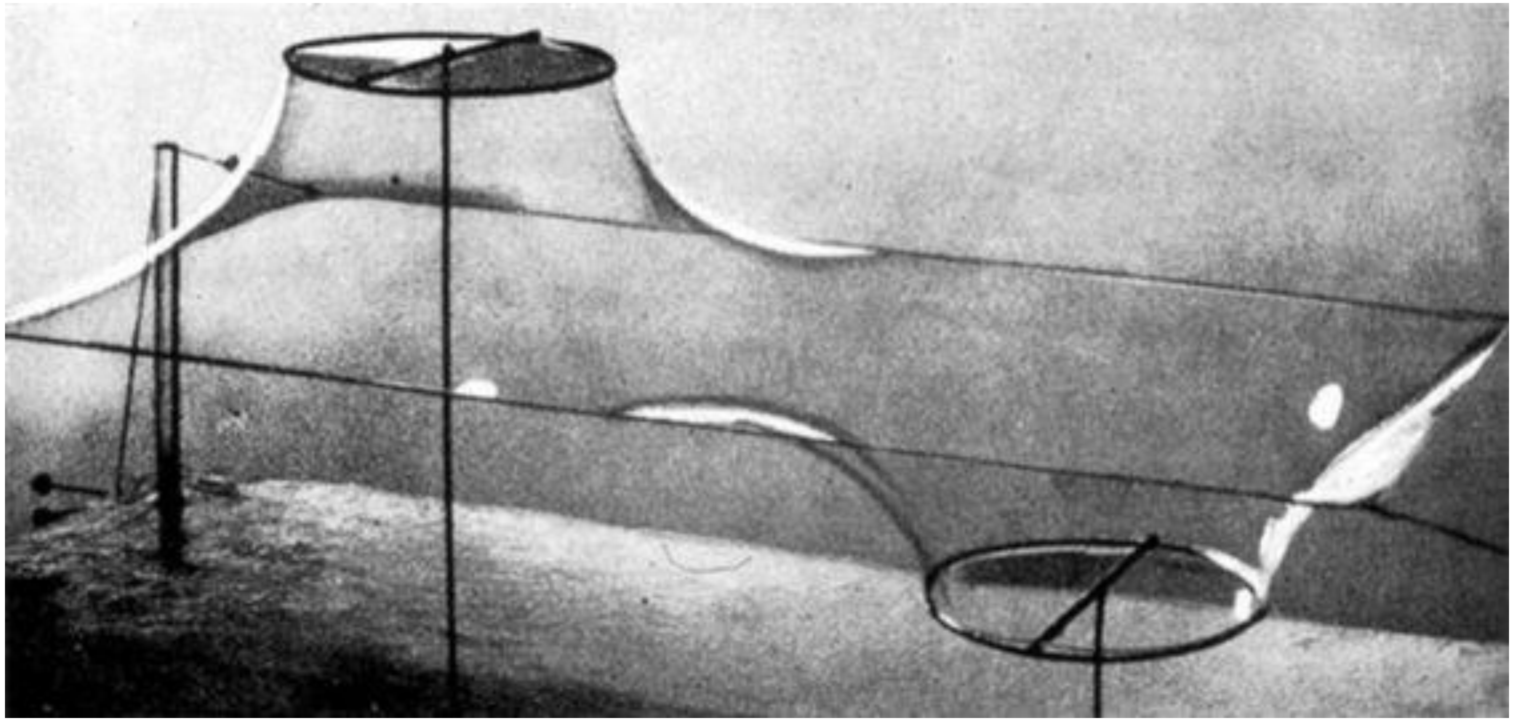
Will two soap-bubbles also when knocked together be unable to squeeze out the air between them? This you can try at home just as well as I can here, but I will perform the experiment at once. I have blown a pair of bubbles, and now when I hit them together they remain distinct and separate (Fig. 55).

I shall next place a bubble on a ring, which it is just too large to get through. In my hand I hold a ring, on

Minimal Surfaces

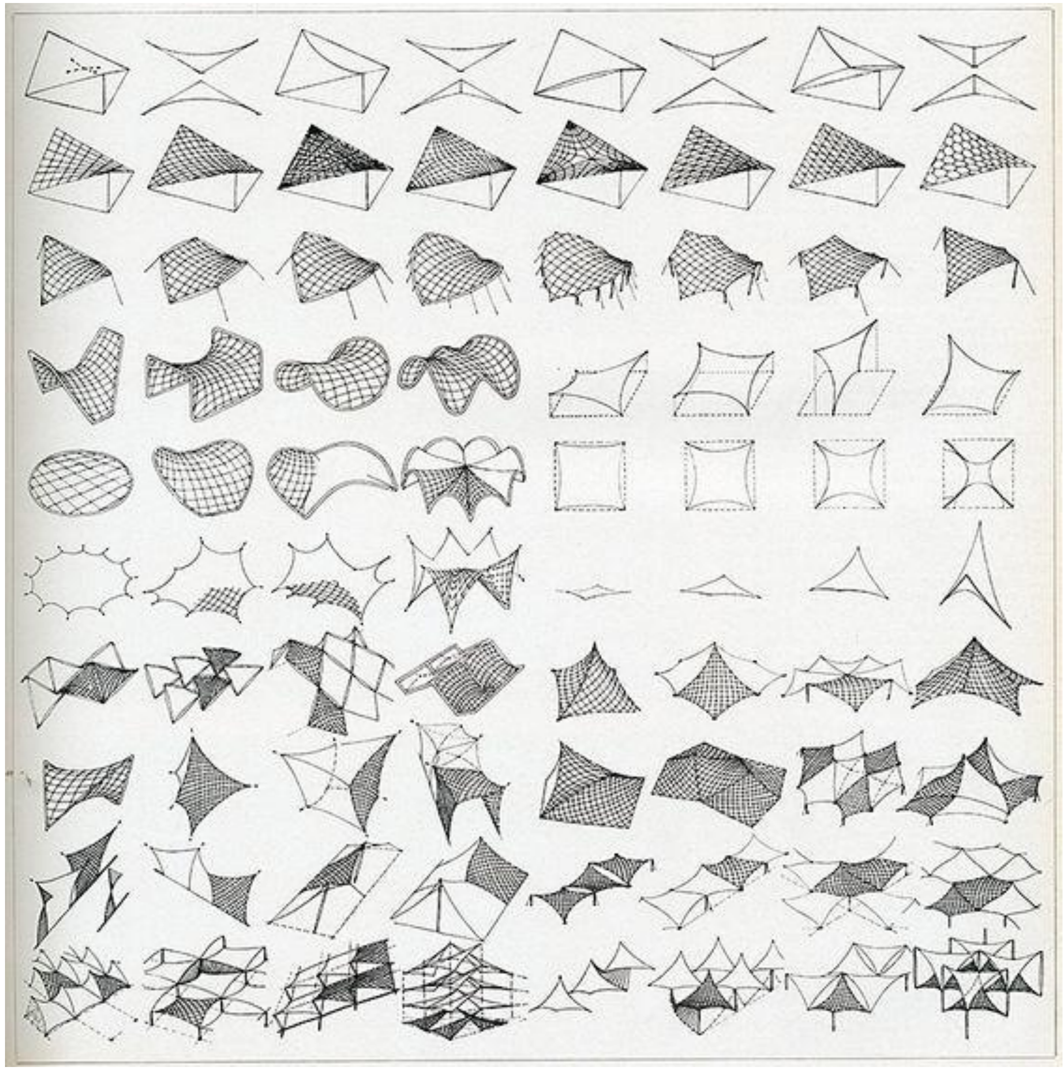
ArchDaily

Minimal Surfaces



ArchDaily

Minimal Surfaces



ArchDaily

Newton’s Laws of Motion

And what they actually mean…

1. Inertness: movement does not require force, in the absence of forces, an object is inert, i.e. it keeps its velocity, whether zero or non-zero
2. Acceleration: the vector sum of forces on an object cause an acceleration inversely proportional to the mass of the object
3. Action and Reaction: When one body exerts a force on a second body, the second body simultaneously exerts a force equal in magnitude and opposite in direction on the first body.
4. Superposition: Forces add up like vectors

TU Delft

Newton’s Laws of Motion

And what they actually mean…

Why do we say velocity and not speed? Is this just a fancy word?

1. Inertness: movement does not require force, in the absence of forces, an object is inert, i.e. it keeps its velocity, whether zero or non-zero.
2. Acceleration: the vector sum of forces on an object cause an acceleration inversely proportional to the mass of the object.
3. Action and Reaction: When one body exerts a force on a second body, the second body simultaneously exerts a force equal in magnitude and opposite in direction on the first body.
4. Superposition: Forces add up like vectors.

Yes sir Newton, but what does this mean for structures? Where does the so-called surface reaction force come from?

Free-Body Diagrams and the mysterious N

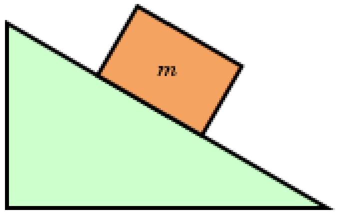
Where does this force N come from in reality?

Observe that if N did not exist, then the body could have accelerated towards the inside of the ramp, because its weight that is equal to mg has a component in the direction perpendicular to the ramp's surface. In fact, in real life if the ramp is not rigid enough, the surface reaction might not be sufficient to keep the body atop.

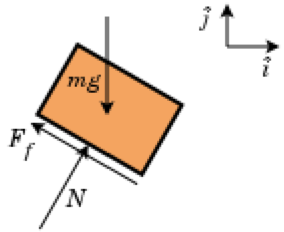
Unrelated to this, or our lecture indeed, is the F_f or the force of friction on the surface, reacting to the component of weight parallel to the ramp's surface. However, what is philosophically related to our subject here is that F_f is a reactive force, i.e. it only appears in reaction to $mg\cos\left(\frac{\pi}{2}-\theta\right)=mg\sin\theta$, where θ is the slope of the ramp.

https://en.wikipedia.org/wiki/Free_body_diagram

A block on a ramp

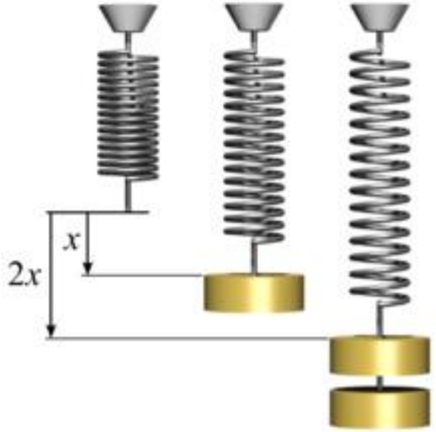


Free body diagram
of just the block



Hooke’s Law

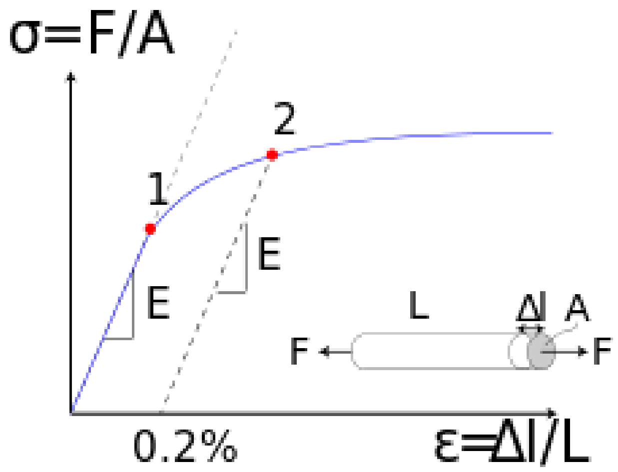
$$f = k\Delta x$$

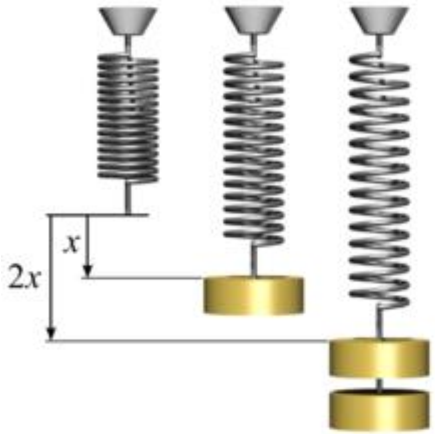


https://en.wikipedia.org/wiki/Hooke%27s_law

TU Delft

Hooke’s Law and the Stress-Strain Relation



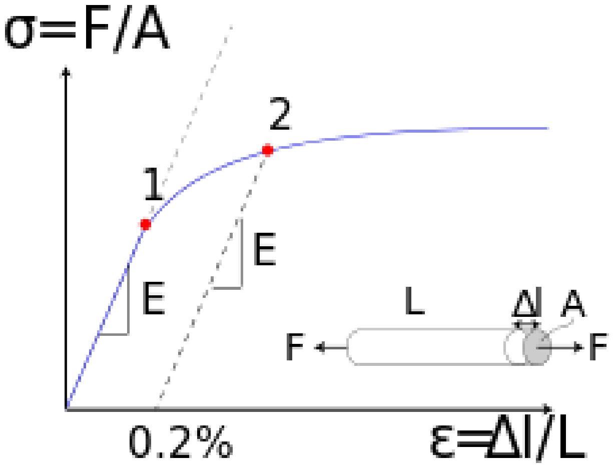


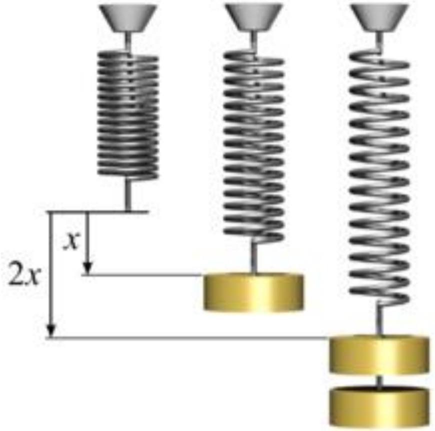
Stress–strain curve showing typical yield behavior for nonferrous alloys. Stress (σ) is shown as a function of strain (ϵ)
1: Elastic (proportionality) limit
2: Offset yield strength (0.2% proof strength)

https://en.wikipedia.org/wiki/Hooke%27s_law

https://en.wikipedia.org/wiki/Stress%E2%80%93strain_curve

Hooke's Law and the Stress-Strain Relation





Stress–strain curve showing typical yield behavior for nonferrous alloys. Stress (σ) is shown as a function of strain (ϵ)
1: Elastic (proportionality) limit
2: Offset yield strength (0.2% proof strength)

https://en.wikipedia.org/wiki/Hooke%27s_law

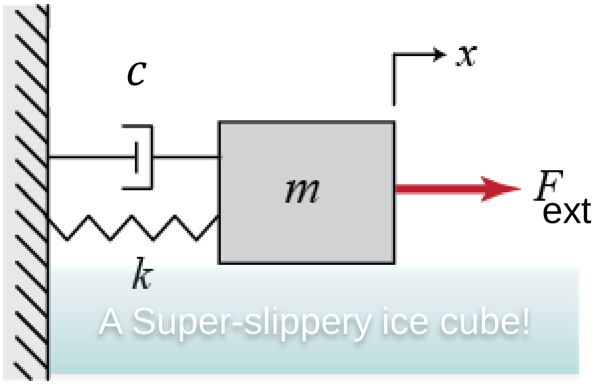
https://en.wikipedia.org/wiki/Stress%E2%80%93strain_curve

TU Delft

Newton's Equation of Motion (free-body)

Effectively: $m\ddot{u} + c\dot{u} + ku = f_{ext}$

Because: $f_r = f_{ext} - kx - c\dfrac{dx}{dt} = ma = m\dfrac{d^2x}{dt^2}$



For a deeper understanding only

This is the archetypical example of a dynamical system, showing:

- *'Time Lead'* (velocity differentiator (w.r.t. time), whose coefficient is the mass of the object
- *'Time Lag'* (velocity integrator (w.r.t. time), whose coefficient is the elasticity of the spring
- *'Proportional Gain'* (velocity dependant damping that is in fact viscose damping, like a honey spoon in a honey jar, the faster you try to oscillate it the harder it resists the motion; the damping principle used in the suspension system of cars)

Optional text, only for further information!

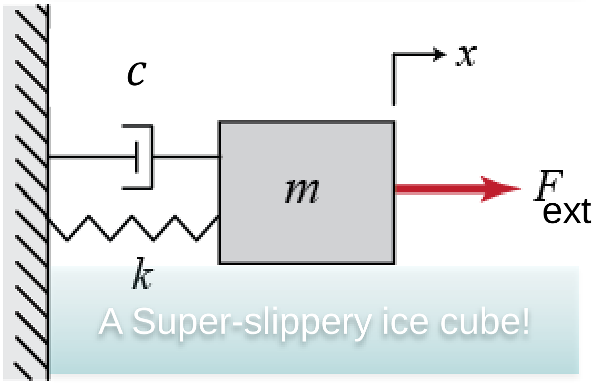
TUDelft

Newton's Equation of Motion (free-body)

Effectively: $m\ddot{u} + c\dot{u} + ku = f_{ext}$

Because: $f_r = f_{ext} - kx - c\dfrac{dx}{dt} = ma = m\dfrac{d^2x}{dt^2}$

Which means the result of active and reactive forces will be the cause of acceleration



For a deeper understanding only

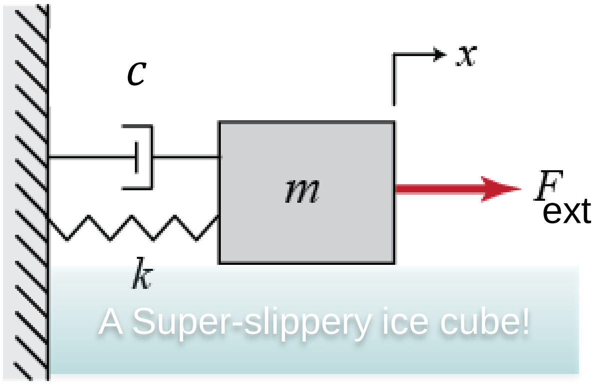
This is the archetypical example of a dynamical system, showing: a system consisted of a solid mass, a viscose damper, and an ideal spring. An external force is exerted to the mass. Contrary to the cases often discussed in statics, here we need to also look at the dynamics (of velocity, that is). However, here we have a simpler question with regards to the displacement u, its time derivative $\dot{u}$ (velocity), and its second time derivative $\ddot{u}$ (acceleration), The important thing here is that the exerted external active force f_{ext} must be the only reason any other force appears in this system. The sum of active and reactive forces on the system is dubbed f which will be in fact equal to $f_{ext} - kx - c\frac{dx}{dt}$ because the spring and the damper force are both reactive and resisting motion in the positive rightward direction. When the outcome f_r is positive, then the system accelerates inversely proportional to its mass and decelerates when the outcome is negative.

TU Delft

Newton's Equation of Motion (free-body)

$$m\ddot{u} + c\dot{u} + ku = f_{ext}$$

$$f_r = f_{ext} - kx - c\frac{dx}{dt} = ma = m\frac{d^2x}{dt^2}$$



No external forces➔ $m\frac{d^2x}{dt^2} + c\frac{dx}{dt} + kx(t) = 0$

Now, we are looking for a solution to this differential equation. In other words, we are looking for a function that is somehow proportionate to its derivatives, because they have to cancel each other out with positive coefficients. There are two kinds of functions with such properties, namely the sinusoidal and exponential functions that are related to one another through Euler's equation $e^{\sigma t + i\omega t} = \mathrm{e}^{\sigma t}(\cos\omega t + i\sin\omega\, t)$

For a deeper understanding only

If $x(t) = e^{\lambda t}$

Then $\mathrm{m}\lambda^2 e^{\lambda t} + c\lambda e^{\lambda t} + ke^{\lambda t} = 0$

Then $\mathrm{m}\lambda^2 + c\lambda + k = 0 \Rightarrow \lambda = \frac{-c \pm \sqrt{c^2 - 4mk}}{2m}$

$for\ c < 2\sqrt{mk}, c^2 - 4mk < 0\ the\ system\ has\ imaginary\ eigevalues\ and\ the\ response\ is\ somewhat\ oscillatory$

$for\ c = 0, c^2 - 4mk < 0\ the\ system\ has\ only\ imaginary\ eigevalues\ and\ the\ response\ is\ strictly\ oscillatory$

$for\ c > 2\sqrt{mk}, c^2 - 4mk > 0\ and\ so, the\ system\ has\ strictly\ real\ eigevalues\ and\ the\ response\ is\ steadily\ convergent\ to\ zero$

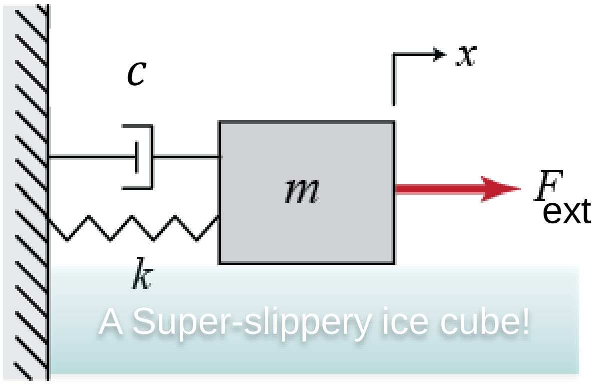
Image Credits:
http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemModeling
https://en.wikipedia.org/wiki/Harmonic_oscillator

Newton's Equation of Motion (free-body)

At the state of the equilibrium, shortly reached after the system is 'excited', the exerted force f_{ext} will be zero, and thus:

Effectively: $$m\ddot{u} + c\dot{u} + ku = f_{ext}$$

Because: $$f_{ext} - kx - c\frac{dx}{dt} = m\frac{d^2x}{dt^2}$$



No external forces, after we let go, f_{ext}=0➔

$$m\frac{d^2x}{dt^2} + 2\zeta\omega_0\frac{dx}{dt} + \omega_0^2 x = 0$$

$$\omega_0 = \sqrt{\frac{k}{m}} \qquad \zeta = \frac{c}{2\sqrt{mk}}$$

For a deeper understanding only

Which means the system is going to start move to balance itself

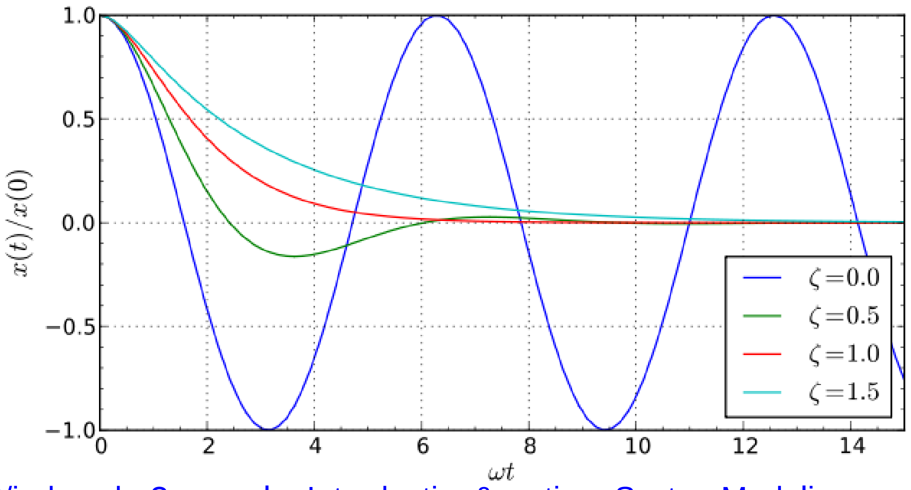


Image Credits:
http://ctms.engin.umich.edu/CTMS/index.php?example=Introduction&section=SystemModeling
https://en.wikipedia.org/wiki/Harmonic_oscillator

TU Delft

Dynamic Relaxation (with vector forces)

Computational Modelling

$$m\ddot{u} + c\dot{u} + ku = f$$

Physical Modelling

Image Credit: https://www.dong.world/2017/10/works-of-antoni-gaudi-677-gaudis-crypt-casa-vicens/

adopted from: Configraphics: Graph Theoretical Methods for Design and Analysis of Spatial ConfigurationsP Nourian
A+ BE: Architecture and the Built Environment 6 (14), 1-348

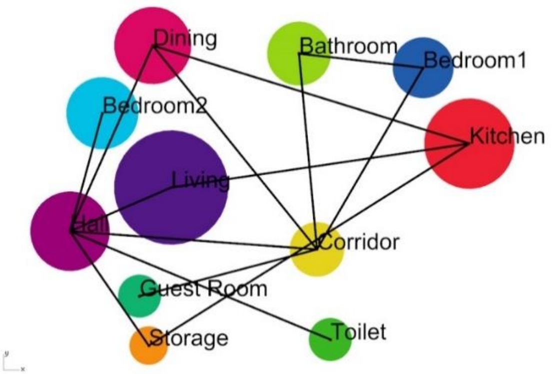
Force-Directed Graph Drawing

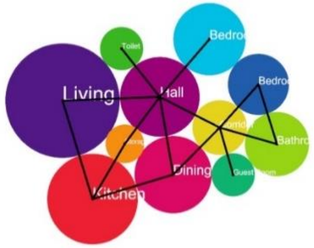
Given the graph $\Gamma = (\mathrm{V}, \mathrm{E})$, $\mathrm{E} = (\mathrm{V_i}, \mathrm{V}_j)$ if $\mathrm{V_i}$ is linked to $\mathrm{V_j}$

Do

- **For Each vertex** $u \in V$
  - **Resulting_Forces=**$\sum Attraction\_Forces(u)$ **+** $\sum Repulsion\_Forces(u)$
  - u**=**u **moved by the Resulting_Forces**
- **Next**
- **Recompute Continuance_Condition:** $\forall (i,j) \in E, x_{ij} \neq (R_i + R_j) \mp ErrorTolerance$
- **Iteration_Count=Iteration_Count+1**

Until (Continuance_Condition=False Or Iteration_Count>MaximumIterations)





Attraction_Forces= $\mathrm{AF_{ij}} = \mathrm{k_a} \Delta \mathrm{x_{ij}}$, if $(\mathrm{I}, \mathrm{j}) \in \mathrm{E}$,

$$\mathrm{k_a} = \text{attraction strength factor},$$

$$\Delta \mathrm{x_{ij}} = Distance\ V_i\ to\ V_j - RestLength(i,j)$$

$$RestLength(i,j) = R_i + R_j$$

Repulsion_Forces= $\mathrm{RF_{ij}} = \frac{\mathrm{k_r}}{\mathrm{x_{ij}}}$, for all (I, j) $if\ x_{ij} < RestLength(i,j)$

$$\mathrm{k_r} = \text{repulsion strength factor}$$

$$x_{ij} = Distance\ V_i\ to\ V_j$$

$$RestLength(i,j) = R_i + R_j$$

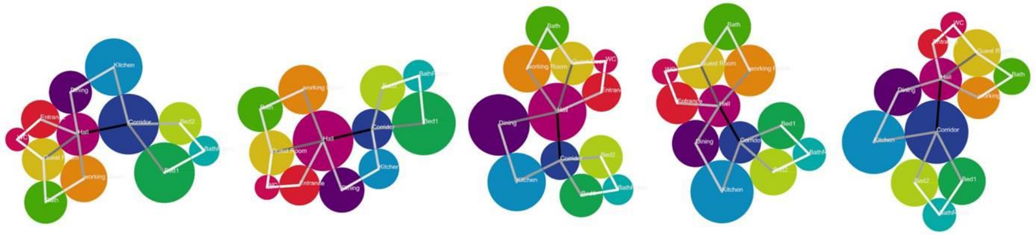


Image Credit:

Dynamic Relaxation (with vector forces)

$$m\ddot{u} + c\dot{u} + ku = f_{ext}$$

For each coordinate x,y, or z:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

Image Credit: https://www.dong.world/2017/10/works-of-antoni-gaudi-677-gaudis-crypt-casa-vicens/

Dynamic Relaxation (with vector forces)

For each coordinate x,y, or z:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

P_{ix}	Applied force at node i in direction x
K_{ix}	Stiffness term at node i in direction x
δ_{ix}^{t}	Total displacement of node i in direction x at time t
C_i	Viscous Damping constant at node i
v_{ix}^{t}	Velocity of node i in direction x at time t
M_i	Lumped fictitious mass at node i chosen to optimise convergence
$\dot{v}_{ix}^{t}$	Acceleration at node i in direction x at time t

Dynamic Relaxation (with vector forces)

For each coordinate x,y, or z:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

$$R_{ix}^{t} = P_{ix} - K_{ix}\delta_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t}$$

How far a node has been displaced from its relaxed position

R_{ix}^{t} Residual (or resultant) of the applied and structural member forces at node i in direction x at time t

These are the forces that are dependant on the position of a node at the t_{th} iteration time.

TUDelft

Dynamic Relaxation (with vector forces)

For each coordinate x,y, or z we do the followings:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

$$R_{ix}^{t} = P_{ix} - K_{ix}\delta_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t}$$

$acceleration\ as\ an\ apaproximate\ derivative\ of\ velocity:$

$$\dot{v}_{ix}^{t} = \frac{v_{ix}^{t+\Delta t/2} - v_{ix}^{t-\Delta t/2}}{\Delta t}$$

$velocity\ of\ a\ moment\ in\ time\ as\ an\ average\ of\ two\ half\ moments\ before\ and\ after:$

$$v_{ix}^{t} = \frac{v_{ix}^{t+\Delta t/2} + v_{ix}^{t-\Delta t/2}}{2}$$

$$R_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t} = M_i\left(\frac{v_{ix}^{t+\Delta t/2} - v_{ix}^{t-\Delta t/2}}{\Delta t}\right) + C_i\left(\frac{v_{ix}^{t+\Delta t/2} + v_{ix}^{t-\Delta t/2}}{2}\right)$$

TU Delft

Dynamic Relaxation (with vector forces)

For each coordinate x,y, or z we do the followings:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

$$R_{ix}^{t} = P_{ix} - K_{ix}\delta_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t}$$

$acceleration\ as\ an\ apaproximate\ derivative\ of\ velocity$:

$$\dot{v}_{ix}^{t} = \frac{v_{ix}^{t+\Delta t/2} - v_{ix}^{t-\Delta t/2}}{\Delta t}$$

$velocity\ of\ a\ moment\ in\ time\ as\ an\ average\ of\ two\ half\ moments\ before\ and\ after$:

$$v_{ix}^{t} = \frac{v_{ix}^{t+\Delta t/2} + v_{ix}^{t-\Delta t/2}}{2}$$

$$R_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t} = M_i\left(\frac{v_{ix}^{t+\Delta t/2} - v_{ix}^{t-\Delta t/2}}{\Delta t}\right) + C_i\left(\frac{v_{ix}^{t+\Delta t/2} + v_{ix}^{t-\Delta t/2}}{2}\right)$$

$Damping\ proportionate\ to\ masses$:

$$C_i = M_i C$$

$$R_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t} = M_i\left(\frac{v_{ix}^{t+\Delta t/2} - v_{ix}^{t-\Delta t/2}}{\Delta t}\right) + CM_i\left(\frac{v_{ix}^{t+\Delta t/2} + v_{ix}^{t-\Delta t/2}}{2}\right)$$

Dynamic Relaxation (with vector forces)

For each coordinate x,y, or z we do the followings:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

$$R_{ix}^{t} = P_{ix} - K_{ix}\delta_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t}$$

https://gitlab.com/Pirouz-Nourian/Earthy/blob/master/Intensive%20Programming%20Workshops/Dynamic%20Relaxation/DynamicRelaxation.py

Damping proportionate to masses:

$$C_i = M_i C$$

$$R_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t} = M_i \left(\frac{v_{ix}^{t+\Delta t/2} - v_{ix}^{t-\Delta t/2}}{\Delta t} \right) + C M_i \left(\frac{v_{ix}^{t+\Delta t/2} + v_{ix}^{t-\Delta t/2}}{2} \right)$$

Long story short:

$$A = \frac{1}{1 + C\Delta t} \quad B = \frac{1 - C\Delta t}{1 + C\Delta t} \qquad v_{ix}^{t+\Delta t/2} = A \frac{\Delta t}{M_i} R_{ix_i}^{t} + B v_{ix}^{t-\Delta t/2}$$

Dynamic Relaxation (with vector forces)

For each coordinate x,y, or z we do the followings:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

$$R_{ix}^{t} = P_{ix} - K_{ix}\delta_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t}$$

https://gitlab.com/Pirouz-Nourian/Earthy/blob/master/Intensive%20Programming%20Workshops/Dynamic%20Relaxation/DynamicRelaxation.py

$$A = \frac{1}{1 + C\Delta t} \quad B = \frac{1 - C\Delta t}{1 + C\Delta t} \quad \boxed{v_{ix}^{t+\Delta t/2} = A\frac{\Delta t}{M_i} R_{ix_i}^{t} + B v_{ix}^{t-\Delta t/2}}$$

Something we can compute based on the current configuration of the system

Now, from the definition of velocity:

Now we can predict the next position of the nodes in the system, using their velocity

$$v_{ix}^{t+\Delta t/2} = \frac{x_i^{t+\Delta t} - x_i^{t}}{\Delta t} \rightarrow \boxed{x_i^{t+\Delta t} = x_i^{t} + v_{ix}^{t+\Delta t/2}\Delta t}$$

Dynamic Relaxation (with vector forces)

For each coordinate x,y, or z we do the followings:

$$P_{ix} - K_{ix}\delta_{ix}^{t} - C_i v_{ix}^{t} = M_i \dot{v}_{ix}^{t}$$

$$R_{ix}^{t} = P_{ix} - K_{ix}\delta_{ix}^{t} = M_i \dot{v}_{ix}^{t} + C_i v_{ix}^{t}$$

https://gitlab.com/Pirouz-Nourian/Earthy/blob/master/Intensive%20Programming%20Workshops/Dynamic%20Relaxation/DynamicRelaxation.py

$$A = \frac{1}{1 + C\Delta t} \quad B = \frac{1 - C\Delta t}{1 + C\Delta t} \quad \boxed{v_{ix}^{t+\Delta t/2} = A\frac{\Delta t}{M_i} R_{ix}^{t} {}_{i} + B v_{ix}^{t-\Delta t/2}}$$

$$v_{ix}^{t+\Delta t/2} = \frac{x_i^{t+\Delta t} - x_i^{t}}{\Delta t} \Rightarrow \boxed{x_i^{t+\Delta t} = x_i^{t} + v_{ix}^{t+\Delta t/2}\Delta t}$$

Longer story: Hooke's law, graph theory, and cosines, projecting stiffness forces along the edges onto principal axes:

Where t is not an exponent but an indicator of time step, $f_{\{i,j\}}$ is the elasticity force along the edge (i,j), and l_{ij} is the length of that edge. The division by this length in every direction gives the relevant shadow.

$$\boxed{R_{ix}^{t} = P_{ix} + \sum_{i\sim j}\left(\frac{f_{i,j}}{l_{i,j}}\right)^{t}\left(x_i - x_j\right)^{t}}$$

The forces along the edges of the mesh is computed with respect to their rest lengths: $f_{i,j} = K\Delta l$

TU Delft

Dynamic Relaxation (pseudocode)

1) $$\boxed{R_{ix}^{t} = P_{ix} + \sum_{i\sim j}\left(\frac{f_{i,j}}{l_{i,j}}\right)^{t}\left(x_i - x_j\right)^{t}}$$

2) $$\boxed{v_{ix}^{t+\Delta t/2} = A\frac{\Delta t}{M_i}R_{ix_i}^{t} + Bv_{ix}^{t-\Delta t/2}}$$

3) $$\boxed{x_i^{t+\Delta t} = x_i^{t} + v_{ix}^{t+\Delta t/2}\Delta t}$$

$$A = \frac{1}{1 + C\Delta t} \quad B = \frac{1 - C\Delta t}{1 + C\Delta t}$$

https://gitlab.com/Pirouz-Nourian/Earthy/blob/master/Intensive%20Programming%20Workshops/Dynamic%20Relaxation/DynamicRelaxation.py

TUDelft

Dynamic Relaxation (pseudocode)

1) $$\boxed{R_{ix}^{t} = P_{ix} + \sum_{i\sim j}\left(\frac{f_{i,j}}{l_{i,j}}\right)^{t}\left(x_i - x_j\right)^{t}}$$

2) $$\boxed{v_{ix}^{t+\Delta t/2} = A\frac{\Delta t}{M_i}{R_{ix}^{t}}_{i} + Bv_{ix}^{t-\Delta t/2}}$$

Well, this is problematic for an algorithm with a discrete time set to Δt

3) $$\boxed{x_i^{t+\Delta t} = x_i^{t} + v_{ix}^{t+\Delta t/2}\Delta t}$$

$$A = \frac{1}{1 + C\Delta t} \qquad B = \frac{1 - C\Delta t}{1 + C\Delta t}$$

https://gitlab.com/Pirouz-Nourian/Earthy/blob/master/Intensive%20Programming%20Workshops/Dynamic%20Relaxation/DynamicRelaxation.py

TU Delft

Dynamic Relaxation (pseudocode)

1) $$\boxed{R_{ix}^{t} = P_{ix} + \sum_{i\sim j}\left(\frac{f_{i,j}}{l_{i,j}}\right)^{t}\left(x_i - x_j\right)^{t}}$$

We, therefore, consider the discrete time of the algorithm as $\Delta t' = 2\Delta t$ or $\Delta t = 0.5\Delta t'$ in all formulae, where $\Delta t'$ is the algorithm's iteration time interval:

2) $$\boxed{v_{ix}^{t+0.25\Delta t'} = 0.5A\frac{\Delta t'}{M_i} R_{ix_i}^{t} + Bv_{ix}^{t-0.25\Delta t'}}$$

3) $$\boxed{x_i^{t+0.5\Delta t'} = x_i^{t} + 0.5v_{ix}^{t+0.25\Delta t'}\Delta t'}$$

$$A = \frac{1}{1+0.5C\Delta t'} \quad B = \frac{1-0.5C\Delta t'}{1+0.5C\Delta t'}$$

https://gitlab.com/Pirouz-Nourian/Earthy/blob/master/Intensive%20Programming%20Workshops/Dynamic%20Relaxation/DynamicRelaxation.py

Dynamic Relaxation (pseudocode)

1) $$\boxed{R_{ix}^{t} = P_{ix} + \sum_{i\sim j}\left(\frac{f_{i,j}}{l_{i,j}}\right)^{t}\left(x_i - x_j\right)^{t}}$$

2) $$\boxed{v_{ix}^{t+\Delta t'} = 2A\frac{\Delta t'}{M_i}R_{ix}^{t}{}_{i} + Bv_{ix}^{t-\Delta t'}}$$

3) $$\boxed{x_i^{t+2\Delta t'} = x_i^{t} + 2v_{ix}^{t+\Delta t'}\Delta t'}$$

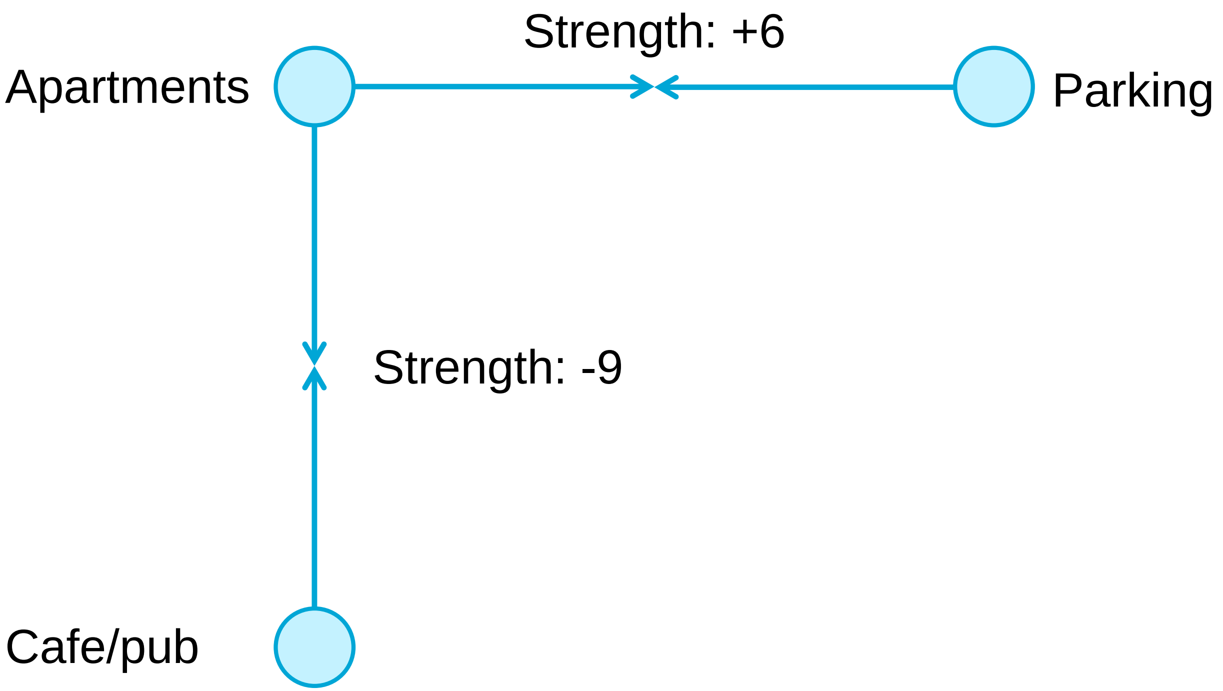
We could, in principle, also consider the discrete time of the algorithm as $\Delta t' = 0.5\Delta t$ or $\Delta t = 2\Delta t'$ in all formulae, where $\Delta t'$ is the algorithm's iteration time interval. However, using these formulae, the velocities will be large and thus likely to entangle the graph. Thus we implement the algorithm as in previous page.

$$A = \frac{1}{1 + 2C\Delta t'} \qquad B = \frac{1 - 2C\Delta t'}{1 + 2C\Delta t'}$$

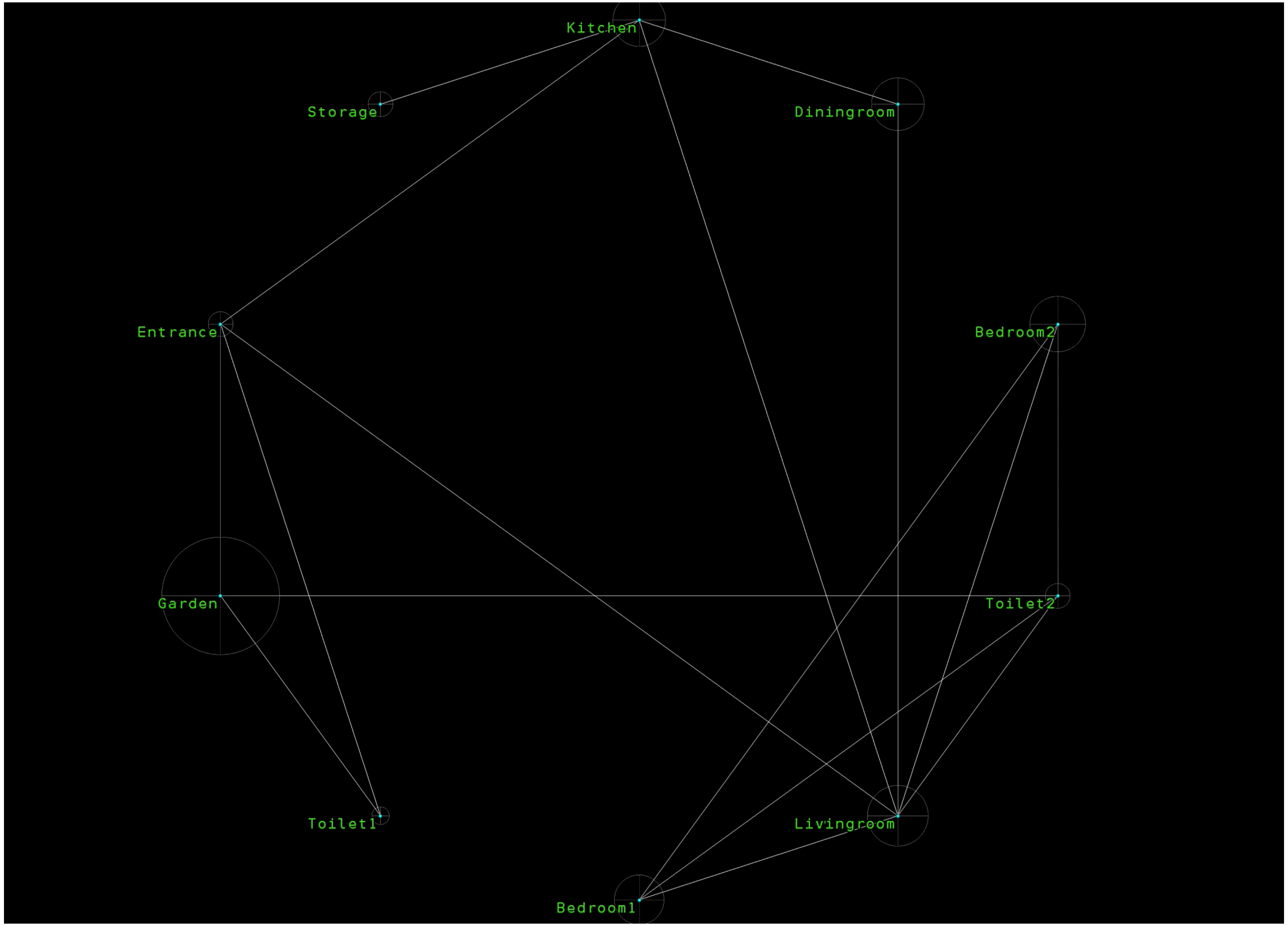
https://gitlab.com/Pirouz-Nourian/Earthy/blob/master/Intensive%20Programming%20Workshops/Dynamic%20Relaxation/DynamicRelaxation.py

Force-directed graph drawing

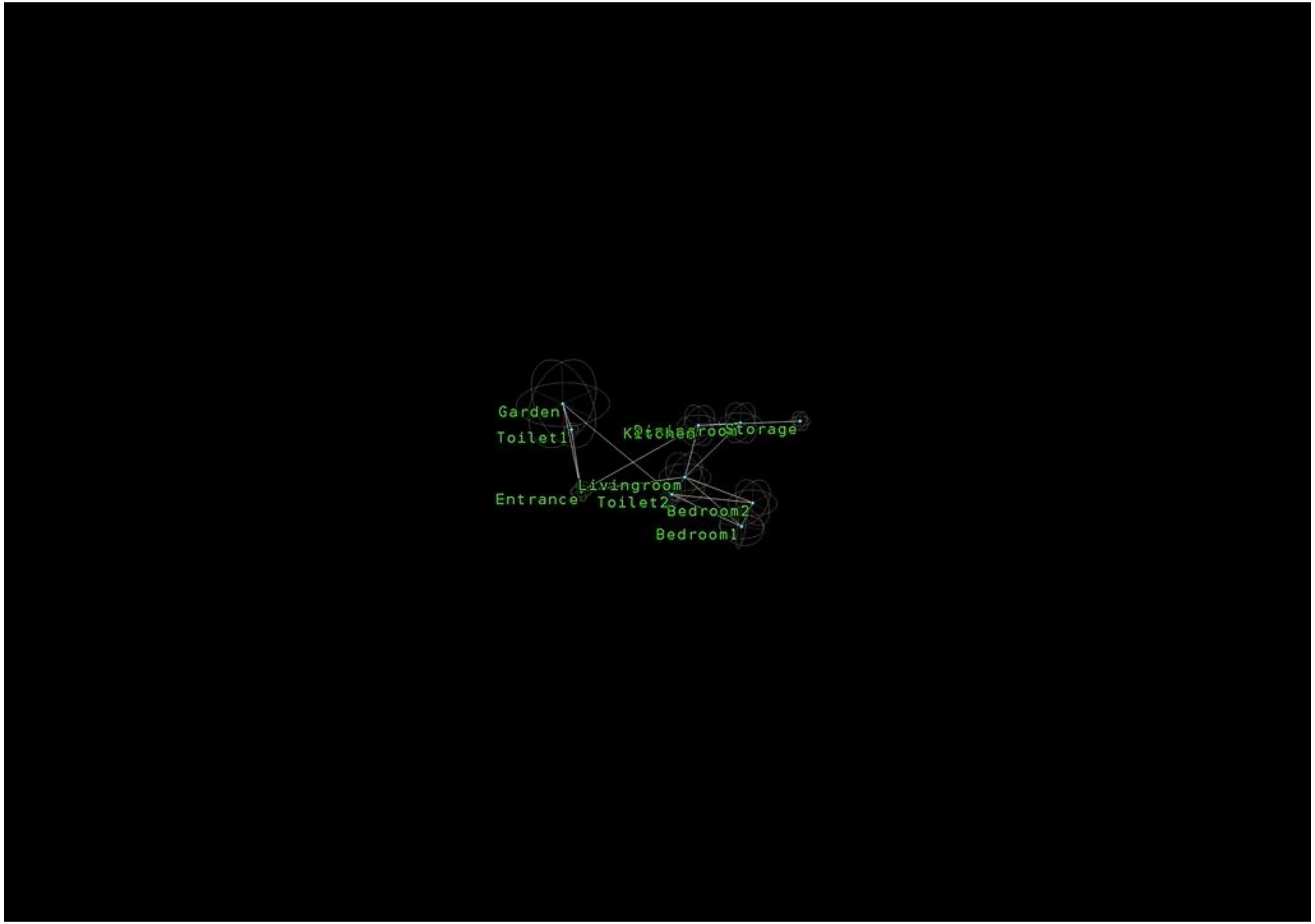
- Same principle as dynamic relaxation, but then applied to a graph
- Connected nodes in a graph are attracted to each other with a certain strength



2D Force-Directed Graph Drawing



3D Force-Directed graph drawing (houdini)



Example input requirements (graph)

- All nodes are given a size based on area
- All relationships are assigned a strength (attracting or repelling)

Nodes

Node	Name	Area (m2)
1	Entrance	4
2	Bedroom	12
3	Garden	12
4	Bathroom	9
5	Kitchen	9

Relationships

Node1	Node2	Strength
2	4	6
3	5	4
1	2	-4

TU Delft

Tasks

- Change .csv input files with your data
- Try to understand what is happening in the Python code
- Extend Python scripts in Grasshopper:
  - How to deal with different strengths between 2 nodes?
  - Make it work in 3D as well
  - Make it work in a voxel grid (nodes can only be located at the centers of voxels, and balls change to groups of voxels)

TU Delft

Questions:

p.nourian@tudelft.nl