Ten $\leftrightarrow$ Square: A Non-Standard Universe

William Dracopoulos   Bartholomé Hubert

February 2026

What if physics was based on different rules?

What if the distance of a point to the origin wasn't given by

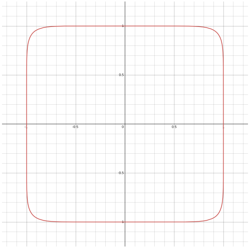
$$d = \sqrt{x^2 + y^2}$$

What if the distance of a point to the origin was given instead by

$$d = \sqrt[10]{x^{10} + y^{10}}$$

In such a universe, everything would likely behave differently. But how differently? Could one even imagine it? Could we make sense of it if we see it?

Here is what a "circle" would look like



Its area can be approximated using a Taylor series:

$$4\int_0^1 \sqrt{1-x^{10}}dx = 4\left(1-\sum_{n=1}^{\infty}\binom{1/10}{n}(-x)^{10n}\right) \approx 3.9429449613$$

In such a universe, π could be defined to be equal to that area.

However, if π is defined to be the ratio of the circumference to the diameter, we reach a contradiction. π would be equal to 3.7887. The number π doesn't have the same properties in such a universe as these two definition to not agree on the same value.

How would the universe look?

Would stars form? If so, what would they look like?

The physics

Gravity: $\mathbf{F_g} = -\frac{Gm_1 m_2}{||\mathbf{r}||^2}\hat{\mathbf{r}}$
Friction: $\mathbf{F_f} = -\beta\dot{\mathbf{r}}$
(Heuristic) Collision force: $\mathbf{F_c} = \frac{Am_1 m_2}{||\mathbf{r}||^3}\hat{\mathbf{r}}$

$$\boxed{\text{But } ||(r_1, r_2)|| = \left(r_1^{10} + r_2^{10}\right)^{\frac{1}{10}}!}$$

Positions and velocities calculated by Euler Integration.
-An invisible body at the center of the simulation keeps everything in the view area

Implementation: avoiding value explosions

Computing powers of 10 creates extremely large values which computers can't capture.
Computing tenth powers is extremely slow.
Solution: A pre-computed look up table with high-precision floats.

$$\left(x^{10} + y^{10}\right)^{\frac{1}{10}} = d \tag{1}$$

$$\implies x\left[1 + \left(\frac{y}{x}\right)^{10}\right]^{\frac{1}{10}} = d \tag{2}$$

$$L(r) = (1 + r^{10})^{\frac{1}{10}} \tag{3}$$

$$a := \max(x, y) \quad b := \min(x, y) \tag{4}$$

$$|a| \cdot L\left(\left|\frac{b}{a}\right|\right) = d \tag{5}$$

Implementation: Optimizing an N-body simulation

An N-body simulation involves the computation of N^2 forces. With thousands of bodies this number gets large.
Solution: hardware acceleration using WebGPU. N^2 threads can compute forces in parallel for an $O(1)$ time complexity. A few more passes sums them to get acceleration.

The simulation

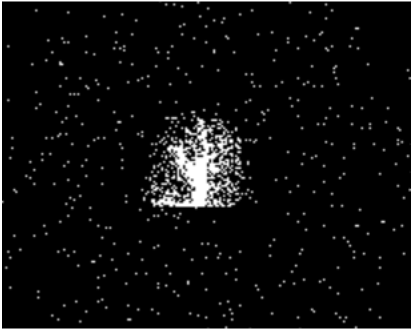


Figure: A forming star

Click here to check out the simulation!