Physical State

by Sven Nilsen

Introduction

I want to investigate the possibility of representing physical states as:

$$(\mathrm{Vec3} \circ \mathrm{Nil3}, \mathrm{Quad} \circ \mathrm{Nil3})$$

Together this is 21 real numbers, encoding information about position, velocity, acceleration, orientation, angular velocity and angular acceleration.

A nilpotent number takes the form:

$$a = a_0 \epsilon^0 + a_1 \epsilon^1 + a_2 \epsilon^2$$

Here, epsilon represents a small number such that:

$$\epsilon^3 = 0$$

Addition and subtraction of nilpotent numbers is done per component.

The multiplication rule of two nilpotent number is:

$$(a \cdot b)_i = \sum_{j=0}^{i} a_j b_{i-j}$$

For the case of nilpotent 3, we have:

$$a \cdot b = (a_0 \cdot b_0)\epsilon^0 + (a_0 \cdot b_1 + a_1 \cdot b_0)\epsilon^1 + (a_0 \cdot b_2 + a_1 \cdot b_1 + a_2 \cdot b_0)\epsilon^2$$

The division rule:

$$\frac{(a_0)}{(b_0)}\epsilon^0 + \frac{(a_1 b_0 - a_0 b_1)}{(b_0^2)}\epsilon^1 + \frac{(a_2 b_0^2 - a_1 b_1 b_0 - a_0 b_2 b_0 + a_0 b_1^2)}{(b_0^3)}\epsilon^2$$

Encoding Information in Nilpotent 3

The way we encode position, velocity and acceleration into nilpotent is as following:

$$f(s,v,a)=s\epsilon^{0}+v\epsilon^{1}+\left(v+\frac{1}{2}a\right)\epsilon^{2}$$

This can also be represented as a tuple:

$$f(s,v,a)=\left(s,v,v+\frac{1}{2}a\right)$$

This is based on the assumption that velocity and acceleration is for the next unit of time.

One way to read the nilpotent number is as a one dimensional particle having a current position, a current velocity and a next velocity. All operations on the nilpotent number should be in one unit of time, that is chosen so small that the physics simulation works approximately well.

If we need to convert the units of time we can use the following function:

$$f(p,t_a,t_b)=p_0\epsilon^{0}+p_1\epsilon^{1}+\left(p_1+(p_2-p_1)\left(\frac{t_b}{t_a}\right)^{2}\right)\epsilon^{2}$$

The rules of nilpotent are the same concerning the first element, which makes it possible to encode momentum if we want to:

$$f(s,v,a,m)=s\epsilon^{0}+mv\epsilon^{1}+m\left(v+\frac{1}{2}a\right)\epsilon^{2}$$

Physical Update

We assume the time unit is small enough to ignore forces that vary with velocity:

$$v_{\text{next}} = v + a\,dt$$
$$s_{\text{next}} = s + \frac{v + v_{\text{next}}}{2}\,dt$$

When we put this together we get:

$$s_{\text{next}} = s + \left(v + \frac{1}{2}a\,dt\right)dt$$

If we store velocity and acceleration for one time unit in a nilpotent number 'p', we have:

$$s_{\text{next}} = p_0 + p_2$$

To update the velocity we can use the formula:

$$v_{\text{next}} = 2\,p_2 - p_1$$

We also need to update the last part because the velocity is changing:

$$v_{\text{next}} + \frac{1}{2}a = 3\,p_2 - 2\,p_1$$

Putting this together we get a this formula:

$$p_{\text{next}} = (p_0 + p_2)\epsilon^0 + (2\,p_2 - p_1)\epsilon^1 + (3\,p_2 - 2\,p_1)\epsilon^2$$

We can write this using 5 operations without any multiplication:

$$p_{\text{next}} = (p_0 + p_2)\epsilon^0 + ((p_2 - p_1) + p_2)\epsilon^1 + ((p_2 - p_1) + (p_2 - p_1) + p_2)\epsilon^2$$

If we need to compute the state after 'n' time units, it can be done very fast using bit shift:

$$p_{\text{next(n)}} = (p_0 + (2^n + n - 2)\,p_2 - (2^n - 2)\,p_1)\epsilon^0 + (2^n\,p_2 - (2^n - 1)\,p_1)\epsilon^1 + ((2^n + 1)\,p_2 - 2^n\,p_1)\epsilon^2$$

From the formula above we see that we can give 'p' discrete coefficients.

Dry Friction

This friction is proportional to the normal force:

$$F = \mu N$$

For example in the case of a moving car we have:

$$F = \mu m g$$

If the car is driven forward with an engine the net force is:

$$F = F_e - \operatorname{sign}(F_e) \mu m g$$
$$F = \operatorname{sign}(F_e)(\operatorname{abs}(F_e) - \mu m g)$$

If the force from the engine is less the normal friction, the car will not move:

$$a = \frac{F}{m} = \operatorname{sign}(F_e) \max\left(\operatorname{abs}\left(\frac{F_e}{m}\right) - \mu g, 0\right)$$

Putting this in a function makes it possible to compute the next velocity of the nilpotent:

$$f(p, F_e, m, \mu, g) = p_0 \epsilon^0 + p_1 \epsilon^1 + \left(p_2 + \operatorname{sign}(F_e) \max\left(\operatorname{abs}\left(\frac{F_e}{m}\right) - \mu g, 0\right)\right) \epsilon^2$$

Skin Friction

This friction is proportional to velocity, such as the air drag around a car:

$$a=\frac{dv}{dt}=-k\,v$$
$$v(t)=v_0\,e^{-k\,t}$$

From this we have two alternative functions:

$$f\,(p\,,k)=p_0\epsilon^0+p_1\epsilon^1+\left(p_2-\frac{1}{2}k\,p_1\right)\epsilon^2$$

$$f\,(p\,,k)=p_0\epsilon^0+p_1\epsilon^1+p_2\,e^{-k}\epsilon^2$$

Rotation Matrix

Euler's rotation theorem states that any two Cartesian coordinate systems with a common origin is related by a rotation about some fixed axis. If we represent the rotation as a rotation matrix we can transform the first coordinate system to the second:

$$R\,A = A_{\text{next}}$$

If we want to reverse the operation we can use the inverse of the rotation matrix:

$$A = R^{-1}\,A_{\text{next}}$$

The product of two rotation matrices is a rotation matrix:

$$mul(R, R) \rightarrow R$$

For a non-identity rotation matrix the eigenvalues are (1, -1, -1) or (1).
The eigenvector corresponds to the axis of rotation.

Absolute Function

We try to construct an absolute function for nilpotent 2:

$$f(x_0 \epsilon^0 + x_1 \epsilon^1) = \mathrm{abs}(x_0)\epsilon^0 + \mathrm{sign}(x_0) x_1 \epsilon^1$$

This is because the 'sign' function is the derivative of the 'abs' function.

With real numbers, we have:

$$y = \mathrm{abs}(x)$$
$$x = \pm y$$

With nilpotent numbers, this is not true in general:

$$y = \mathrm{abs}(x)$$
$$x \neq \pm y$$

This is because the 'sign' function returns 0 if the argument is 0, so we get:

$$f(0\epsilon^0 - 1\epsilon^1) = 0\epsilon^0 + 0\epsilon^1$$
$$0\epsilon^0 - 1\epsilon^1 \neq \pm(0\epsilon^0 + 0\epsilon^1)$$

Which is bad because the way we define functions should behave separately for rational and nilpotent numbers. To fix this we need to treat 0 in first term as a special case:

$$f(0\epsilon^0 + x_1 \epsilon^1) = 0\epsilon^0 + \mathrm{abs}(x_1)\epsilon^1$$
$$f(x_0 \epsilon^0 + x_1 \epsilon^1) = \mathrm{abs}(x_0)\epsilon^0 + \mathrm{sign}(x_0) x_1 \epsilon^1$$

Extended for nilpotent 3 we have:

$$f(0\epsilon^0 + 0\epsilon^1 + x_2 \epsilon^2) = 0\epsilon^0 + 0\epsilon^1 + \mathrm{abs}(x_2)\epsilon^2$$
$$f(0\epsilon^0 + x_1 \epsilon^1 + x_2 \epsilon^2) = 0\epsilon^0 + \mathrm{abs}(x_1)\epsilon^1 + \mathrm{sign}(x_1) x_2 \epsilon^2$$
$$f(x_0 \epsilon^0 + x_1 \epsilon^1 + x_2 \epsilon^2) = \mathrm{abs}(x_0)\epsilon^0 + \mathrm{sign}(x_0) x_1 \epsilon^1 + \mathrm{sign}(x_0) x_2 \epsilon^2$$