Absoid Functions

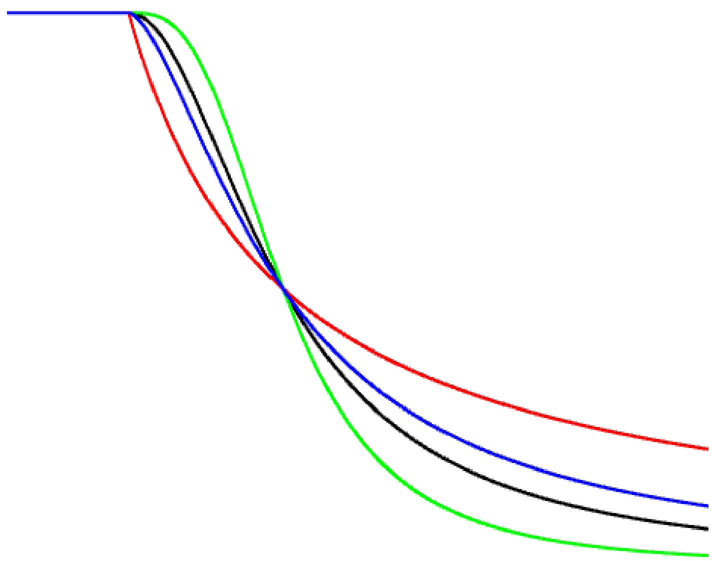
by Sven Nilsen, 2020

In this paper I describe a useful family of open-ended interpolation functions for absolute inputs.

In utility programming, it is sometimes desirable to control when some behavior becomes more important than others. For example, when an AI agent is approaching some target, it is important for safety analysis to know at what distance the AI agent's behavior is changing from one kind to another.

Although many interpolation functions exists, I did not manage to find a suitable formula. As a consequence, I ended up developing my own. It is a simple modification of commonly used formulas.

An \`absoid\` function can be thought of as similar to smooth step or sigmoid functions:



The name \`absoid\` was inspired by combiing \`abs\` and \`sigmoid\`.

An absoid function is constructed using the following higher order function:

$$\text{absoid}(z : \mathbb{R}, n : \mathbb{R} \wedge (> 0)) = \backslash(x : \mathbb{R}) = \text{if } x <= 0 \ \{ \ 1 \ \} \text{ else } \{ \ 1 \ / \ ((x \ / \ z)^n + 1) \ \}$$

This family of functions have the following properties:

$\forall$ z, n { absoid(z, n)(z) = 0.5 }   \`z\` is a turning point

$\forall$ z, n { lim x $\rightarrow$ ∞ { absoid(z, n)(x) } == 0 }   converges to zero in the limit

The parameter \`n\` controls how rapidly the transition is at the turning point. A higher value gives a more rapidly transition, but also smoothes out the function around origo.