

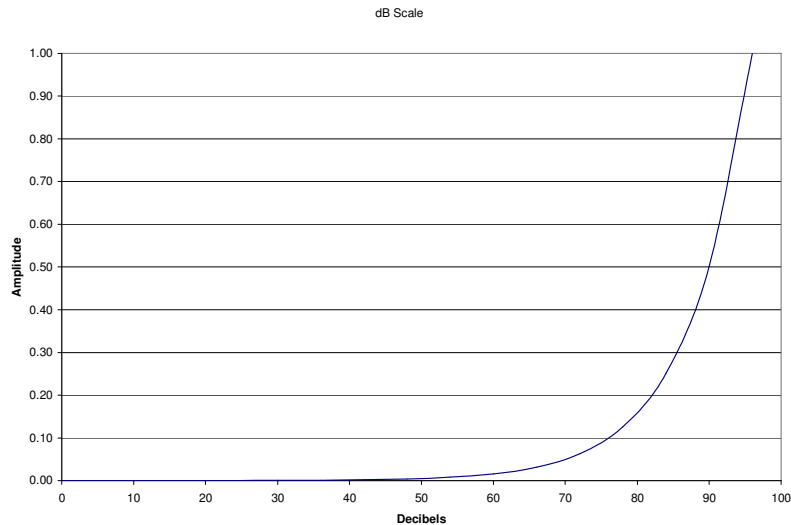
Peak Amplitude in Decibels

The peak amplitude of the envelope determines the volume level for the signal. Since we multiply the volume level with the signal, we will usually normalize the range of the envelope from [0,1] for convenience. A value of zero produces silence and a value of 1.0 produces maximum amplitude. Values in between result in attenuation of the signal.

When specifying the peak amplitude it is helpful if relative amplitude values produce relative changes in loudness. However, a simple linear change in amplitude does not produce a linear change in loudness. Our ears hear on a logarithmic scale such that a doubling of intensity is needed to result in a perception that the sound is twice as loud. For this reason, volume levels are usually indicated using a decibel (dB) scale. The decibel level is calculated from a linear amplitude ratio as a logarithm.

$$dB = 20\log(A_2/A_1) \quad (6.4)$$

When describing peak amplitude, we can set A_1 to 1. A linear change in decibels produces an exponential change in amplitude.



To specify volume in dB we must calculate the equivalent amplitude level by the inverse of the equation (6.4). To keep the amplitude normalized, we divide by the maximum amplitude value.

$$A = \frac{10^{dB/20}}{10^{dB_{\max}/20}} \quad (6.5)$$

Applying the laws of exponents, we can reduce the equation to:

$$A = 10^{dB/20} \cdot 10^{-dB_{\max}/20} = 10^{dB - dB_{\max}/20} \quad (6.6)$$

This equation specifies volume as decibels of gain. A value of 0 for dB results in dB_{\max} attenuation and a value of dB_{\max} results in 0dB attenuation. Alternatively, we can specify volume as attenuation, in which case 0 represents maximum volume and dB_{\max} represents the minimum, and we need to invert the sign of the dB value. We set the peak amplitude variable by one of the two following statements.

```
peakAmp = pow(10, (dbAmp - maxDB)/20) // gain
peakAmp = pow(10, -dbAmp/20); // attenuation
```

The maximum dB level needs to be chosen so that we can effectively cover the entire amplitude range of the final signal. For a 16-bit system, we have an amplitude range of $\pm 2^{15}$ or 96dB if we round the output. For a 24-bit system, we could use a range of 144dB.

No value for dB can produce an amplitude value of 0 and thus cannot directly represent a peak amplitude for silence. For example, using a maximum of 96dB and setting db for 96dB attenuation in the equations above will set *peakAmp* to 0.0000158. To convert to 16-bit integer output, we multiply by 2^{15} (32767) and round the result, producing a final value of 1. If we truncate the result, 96dB will represent an amplitude of 0. In either case, a attenuation greater than 96dB will produce silence. However, when used internally with single precision float, many of the amplitude values will be very small numbers close to 0 and could potentially cause problems. A good solution is to treat any attenuation level greater than the maximum as "infinite" attenuation and set the linear amplitude to 0. A gain level of 0 is also treated as infinite attenuation.

```

// Attenuation:
if (dbAmp >= maxDB)
    peakAmp = 0;
else
    peakAmp = pow(10, -dbAmp/20.0);

// Gain:
if (dbAmp <= 0)
    peakAmp = 0;
else
    peakAmp = pow(10, (dbAmp - maxDB)/20.0);

```

In addition to applying the envelope to the signal, we will usually multiply by an overall volume level.

```
sample = OSC() * EG() * masterVol;
```

If we use dB for both envelope and master volume, we can sum the dB values and multiply once by the result.

```

db = EG() + masterVol;
amp = pow(10, -db/20);
sample = OSC() * amp;

```

This technique is especially useful when several values are combined to produce the volume level. The exponential function can be replaced with a lookup table for conversion of dB to amplitude, an additional savings in computation time. Hardware synthesizers typically use this technique since digital circuits can perform addition and table lookup faster and easier than multiplication.

For software synthesizers, it is generally simpler and more effective to multiply the normalized amplitude levels. However, specifying volume levels in dB is often used for inputs from the user since it is the common way engineers describe volume levels, and musicians who use synthesizers are familiar with it.