音響旋律

すべての音は音響「旋律」である

音響「旋律」といっても、それは古典的な「メロディー」の概念とは異なる。その出発点は、最初に定義したある継続時間とその間のパラメータ変化を有したすべての音響要素を、ある持続をもった旋律であると捉えることにある。いかなる音も、その開始点(a)から終止点(b)に向かう運動を知覚することで、「この音は(a)から(b)に来た」というベクトルを、記憶をもとに意味のあるひとまとまりであると認識することができる。点(クリック)という最もシンプルな素材でさえも、1サンプル(1/44100秒)の長さの旋律であるとみなすことができる。

ある音の状態Aが状態Bに向かう変化を知覚し、AとBの関係性を聴き取ることで、旋律としての意味が生れる。最初に現れた状態Aをもとにして、そこから反復や緊張、解決(弛緩)といった機能をもつ状態B，C…が続いていく音の連鎖から生まれる相互の関係性が音響旋律である。音響旋律とはある音の状態Aが状態Bへ向う意識のベクトル(志向性)の連鎖であり、音Bに到達した時の音Aに対する記憶が、時間の中で(水平方向に)連鎖し蓄積されていくことで生まれる。

水平構造と垂直構造

音楽の共時的（synchronic）な構造のことを垂直構造、通時的（diachronic）な構造のことを水平構造と呼ぶ。この水平構造と垂直構造（という概念）は、音響構造における空間的なメタファーである。例えば五線譜の場合、横軸に時間、縦軸に音程が表現されているため、和音のような音の積み重ねが視覚的には垂直方向に表現され、旋律の時間的な流れが水平方向に表現されている。

「旋律」は音響の水平構造である

音をパラメータで表現した場合、この旋律ベクトルは音のパラメータの時間的な変化として一般化できる。旋律が単なる音の例なりではなく、「旋律」であり得るのは、列なっている音相互間の関係と聴き手の記憶や背景などによって、それらがフレーズ(楽句)という形にグルーピングされるからである。

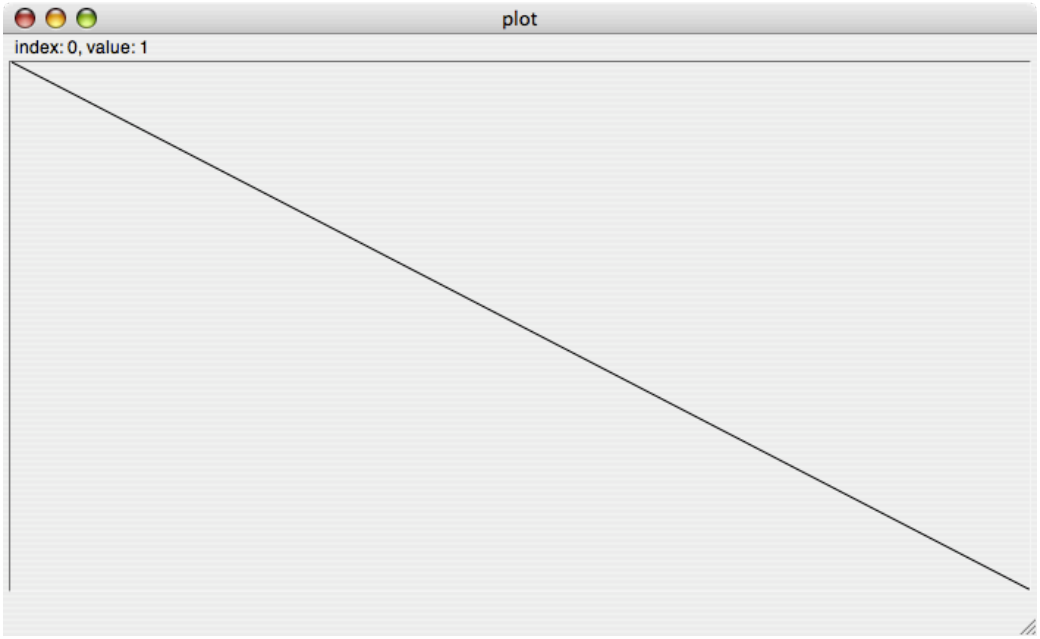
音響旋律は、従来の旋律が持っている関係性とグルーピングの概念をさらに掘り下げて、音そのものが内包しているミクロな状態とその関係性にも着目する。音響旋律は、音内部のより微小で精緻な構造を聴き取るための概念でもある。

「エンヴェロープ」はパラメータの時間変化である

音響素材の項で、音の開始時刻（初期値）と終了時刻（終値）に至る、パラメータの時間変化を考えた。ここでは音響旋律を記述するために、さらに広い意味でのパラメータの時間変化を考える。

音のパラメータの(音の周波数に比べて)比較的ゆっくりとした時間変化のことを「エンヴェロープ（envelope)」と呼ぶ。エンヴェロープは時間の関数であり、その値はパラメータの基本値に対する係数として表わされる。音響素材の項で定義した、音の開始時刻（初期値）から終了時刻（終値）までのパラメータの線形変化も、エンヴェロープのひとつである。エンヴェロープを用いて、さまざまな音響旋律を記述することができる。

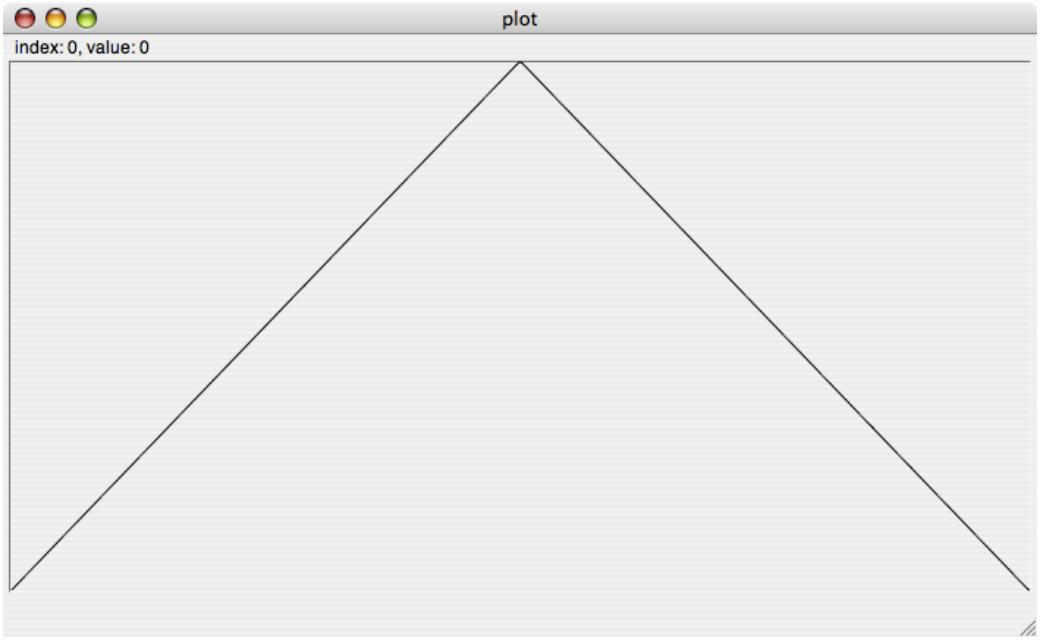
```
// 1秒間かけて値が1から0に変化するシンプルなエンヴェロープ
Env.new([1, 0], [1]).test.plot;
```



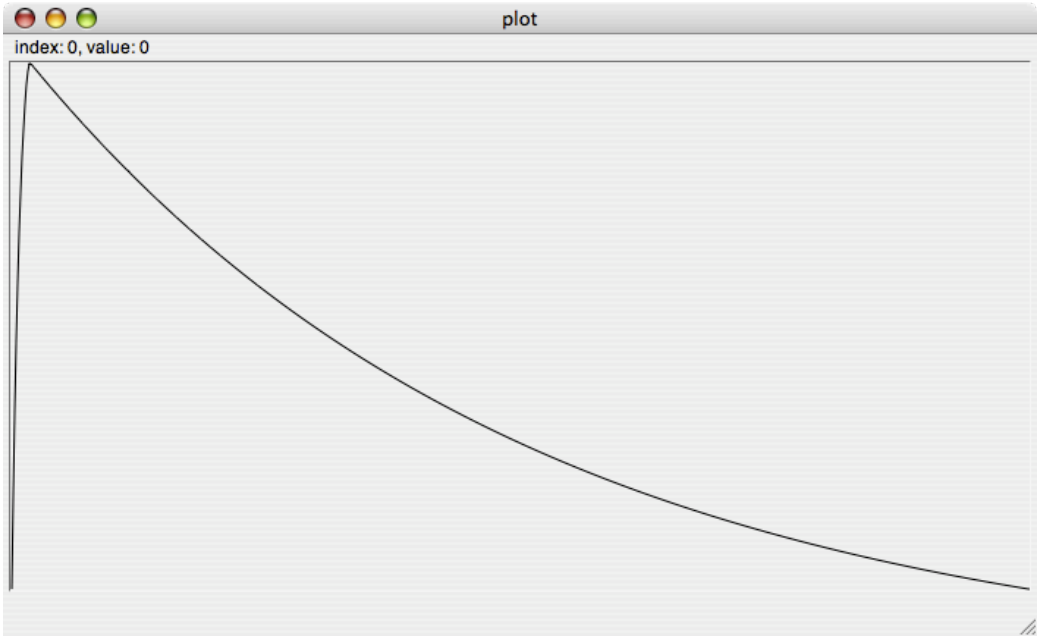
このエンヴェロープは以下のライン・ジェネレータと同義である

```
Line.kr(1, 0, 1);
```

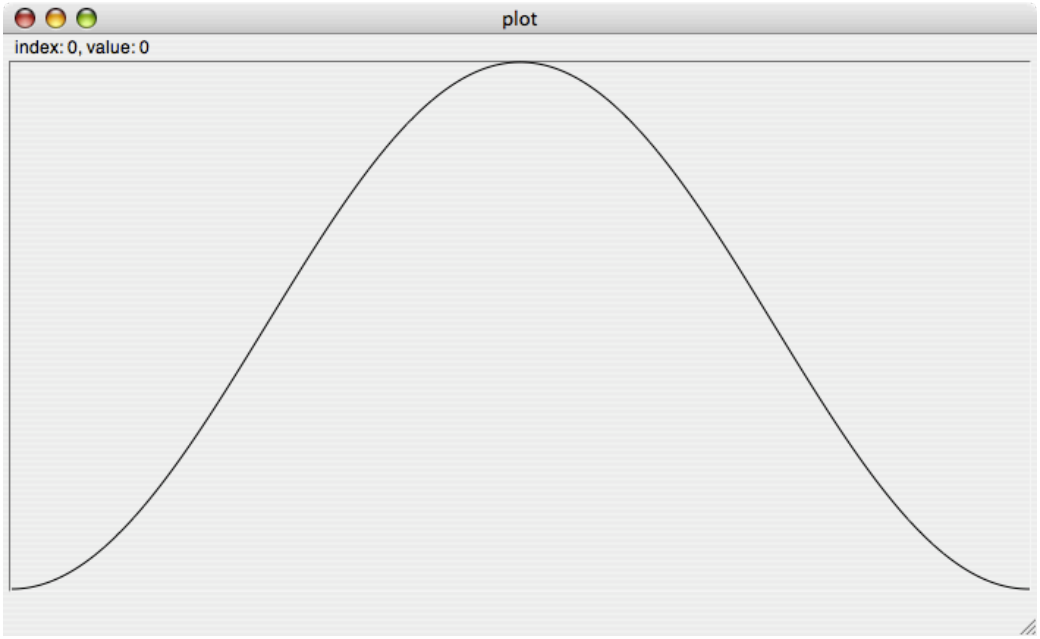
```
// 三角形のエンヴェロープ
Env.triangle(5, 0.5).test.plot;
```



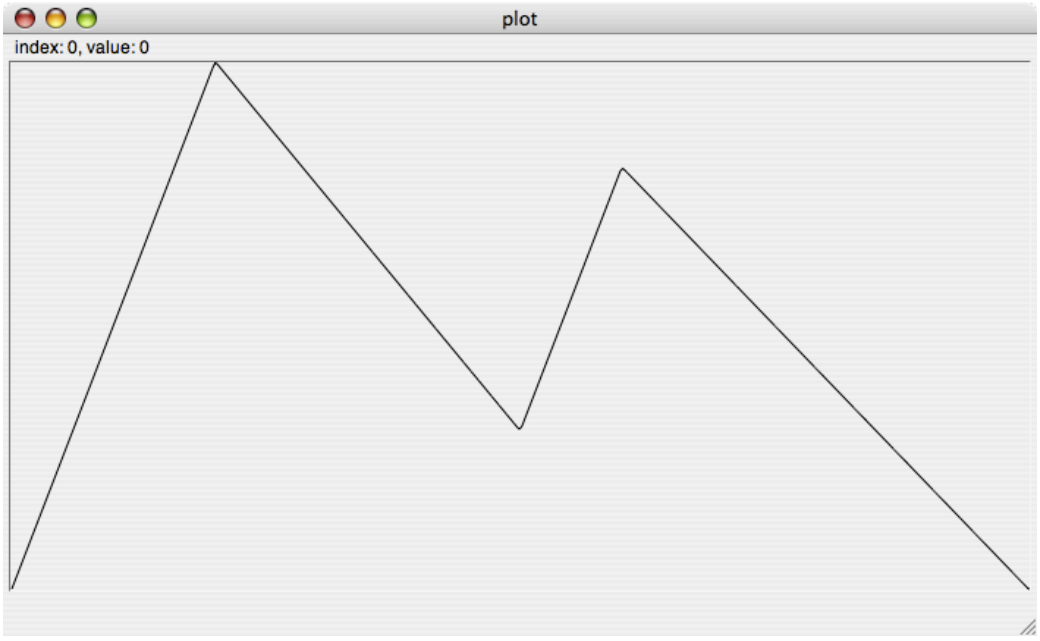

```
// パーカッシヴな(立ち上がりの鋭い)エンヴェロープ
Env.perc(0.05, 3, 0.5, -2).test.plot;
```



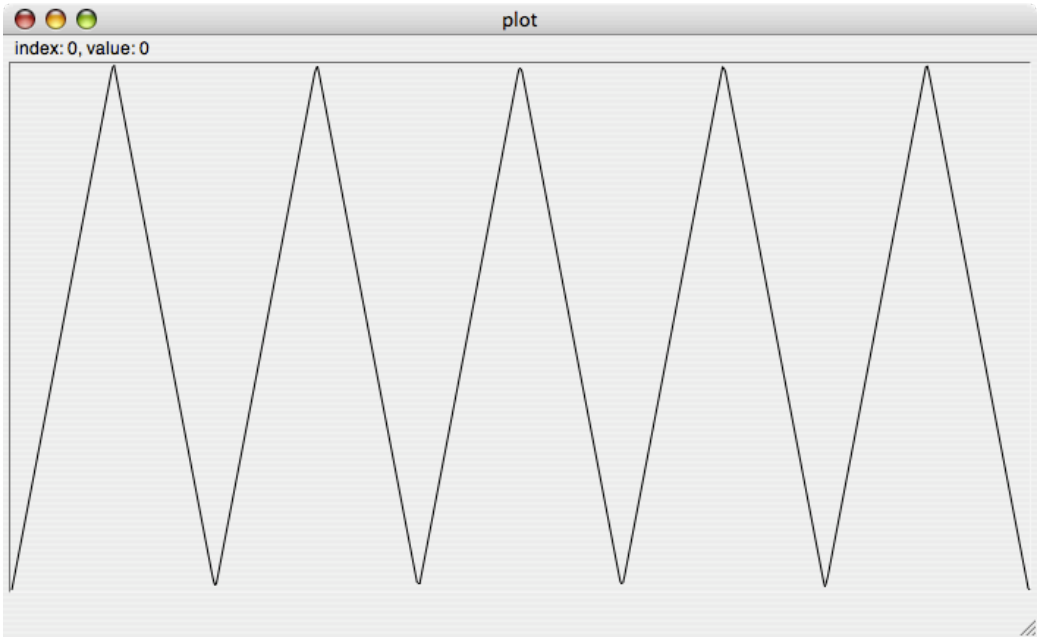

```
// 正弦波形のエンヴェロープ
Env.sine(3, 0.5).test.plot;
```



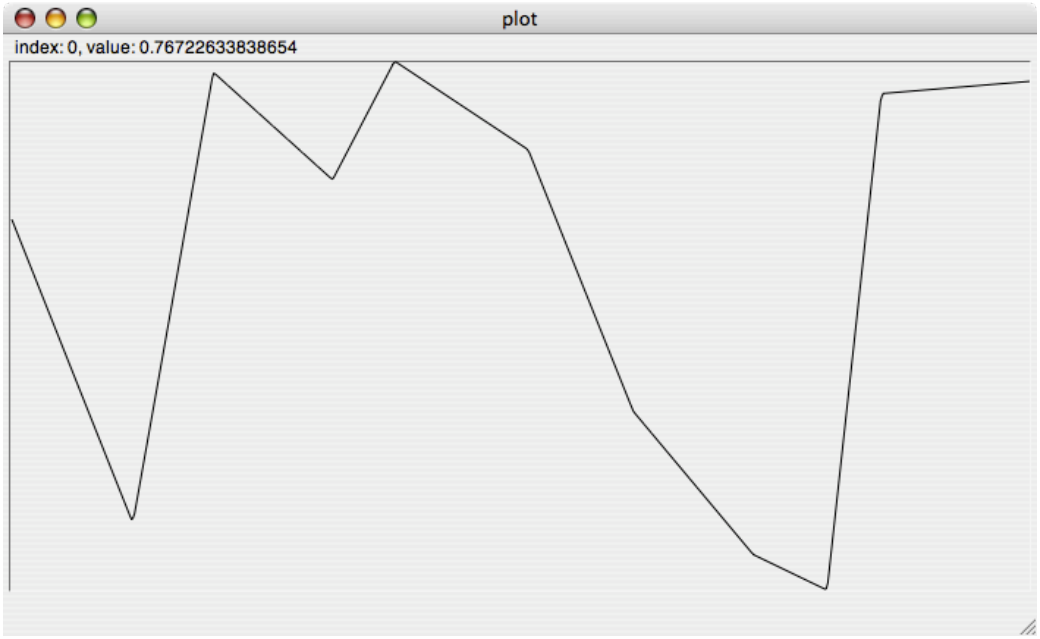

```
// エンヴェロープ任意の折れ線として一般化できる。

Env.new([0, 1, 0.3, 0.8, 0], [2, 3, 1, 4]).test.plot;
```



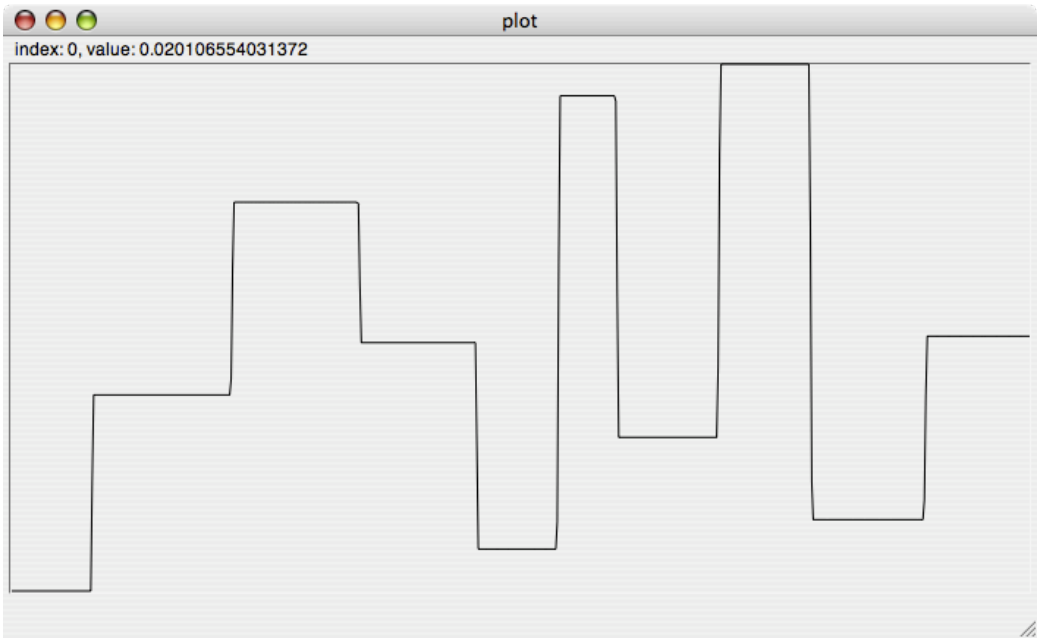

```
// 10秒間のうちに値が11回上下するエンヴェロープ
Env.new([0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 0], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]).test.plot;
```



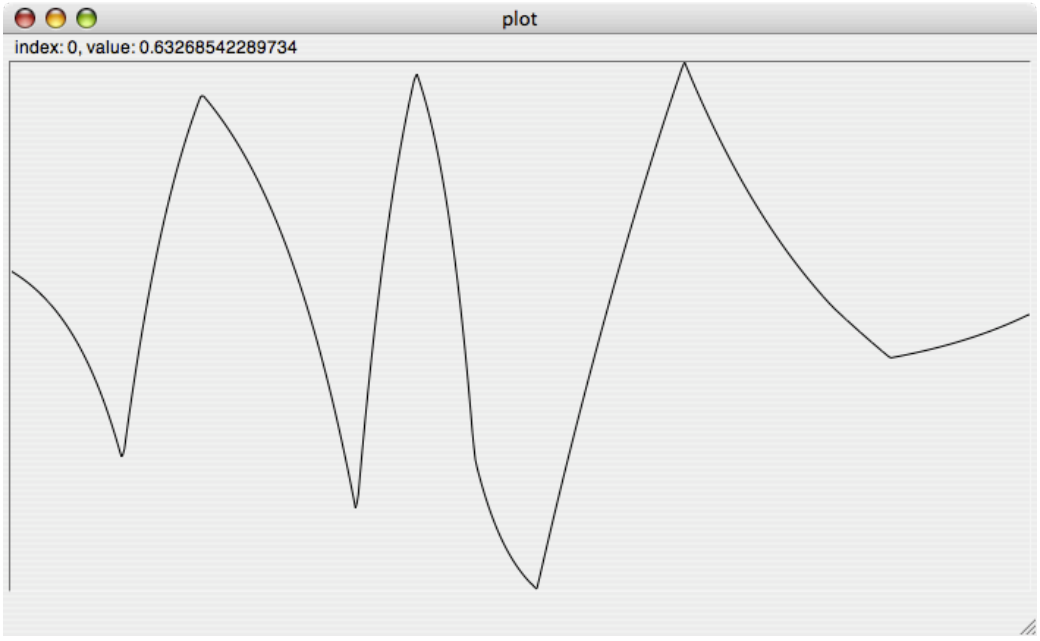

```
// 同じものをこのように書くこともできる
Env.new(Array.fill(11, {|i| i % 2}), Array.series(10, 1, 0)).test.plot;


// ランダムなエンヴェロープ
Env.new(Array.rand(11, 0.0, 1.0), Array.rand(10, 0.5, 1.5)).test.plot;
```



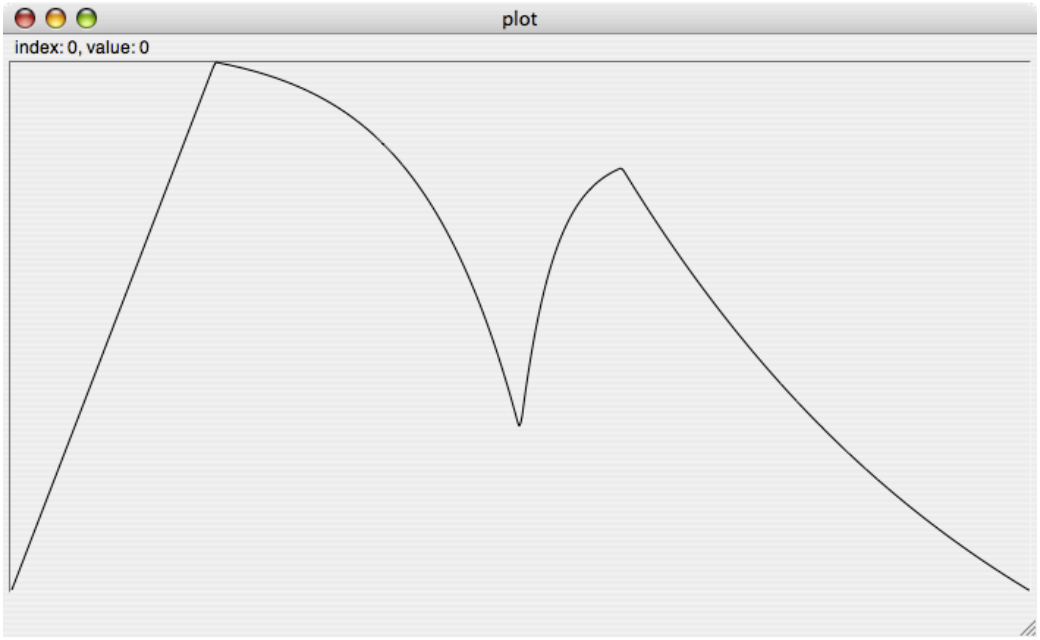

```
// ノードのつなぎ方(セグメントの形状)にはいろいろある

// ステップ状
Env.new(Array.rand(11, 0.0, 1.0), Array.rand(10, 0.5, 1.5), \step).test.plot;
```



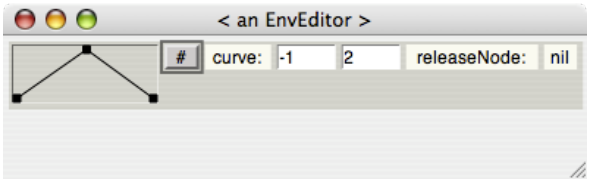

```
// 線形(デフォルト)
Env.new(Array.rand(11, 0.0, 1.0), Array.rand(10, 0.5, 1.5), \linear).plot;

// 指数関数形
Env.new(Array.rand(11, 0.0, 1.0), Array.rand(10, 0.5, 1.5), \exponential).plot;

// ランダムな曲率
Env.new(Array.rand(11, 0.0, 1.0), Array.rand(10, 0.5, 1.5), Array.rand(10, -2.0, 2.0)).plot;
```




```
// セグメント毎に曲率を指定する

Env.new([0, 1, 0.3, 0.8, 0], [2, 3, 1, 4]).test.plot;
```




```
// エンヴェロープのエディット
EnvEditor(Env.new([0, 1, 0], [1.0, 0.5], [-1.0, 1.0])).gui
```




```
Env([ 0, 1.5, 0 ], [ 1.1491935483871, 1.1008064516129 ], [ -1, 2 ], nil, nil)
```

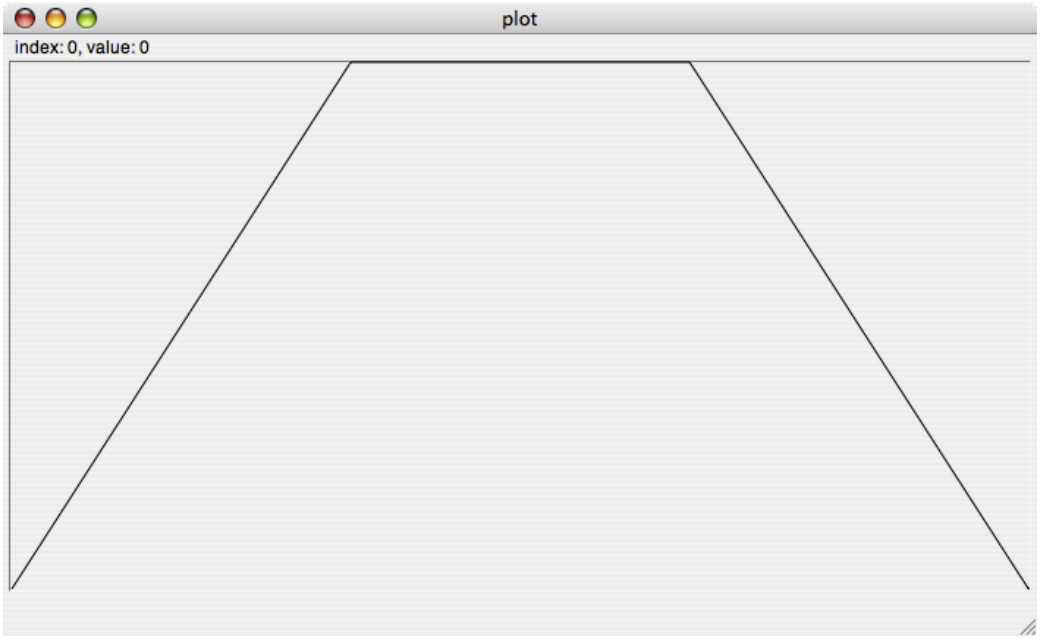
台形のエンヴェロープ

直感的にわかりやすく、また実際に使い易いのが「台形のエンヴェロープ」である。デジタルサウンド特有の鳴り始めと鳴り終りのプチッというクリックノイズ(波形の不連続点)を避けるためにも用いられる。

台形エンヴェロープでは、パラメータの時間変化を、アタック、サステイン、リリースの3つの部分に分けて表現する。
・アタック・タイム(attack time):音の鳴り始めからパラメータが最大値に達するまでの時間
・サスティン・タイム(sustain time):パラメータが最大値を取り続けている時間
・リリース・タイム(release time):パラメータが最大値から0に達するまでの時間

台形エンヴェロープの例を示す。エンヴェロープで線の大きさ(サイン波の音量)を変化させながら、エンヴェロープ曲線を示す。

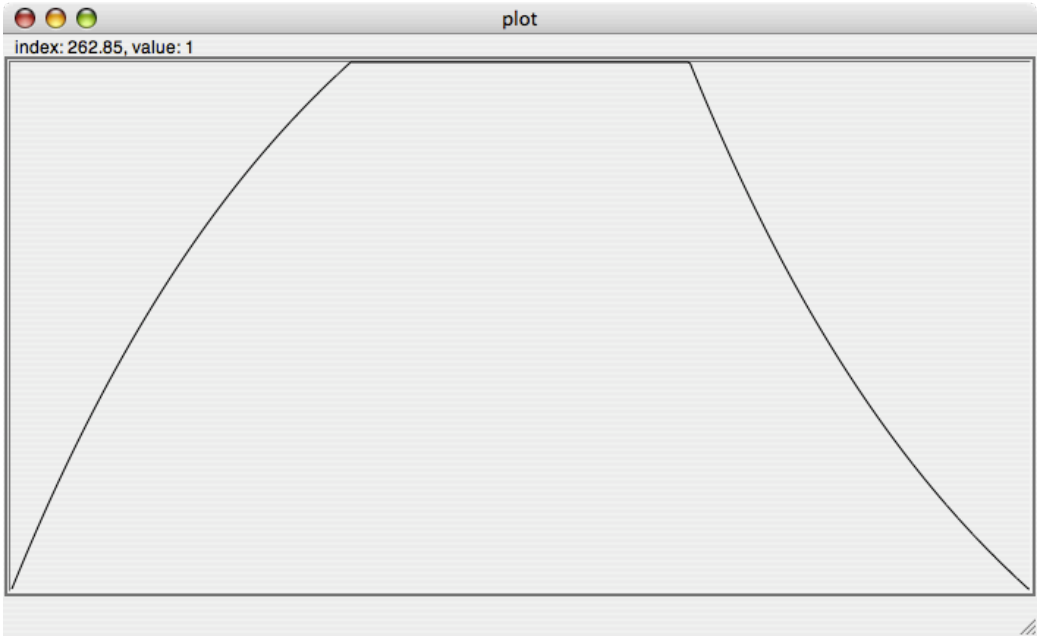
```
// アタック、サスティン、リリース・タイムのいずれもが1秒のエンヴェロープ

Env.linen(1, 1, 1).test.plot;
```




```
Env.linen(0, 1, 2).test.plot;
Env.linen(2, 1, 0).test.plot;
Env.linen(0.001, 3, 0.001).test.plot;

// サステイン・タイムを0にすることで三角形のエンヴェロープも表現できる
Env.linen(3, 0, 0).test.plot;
Env.linen(0, 0, 3).test.plot;
Env.linen(1.5, 0, 1.5).test.plot;
```

変化の曲率(湾曲の度合い)を調整する。

```
Env.linen(1, 1, 1, 1, -10).test.plot;
Env.linen(1, 1, 1, 1, -1).test.plot;
```




```
Env.linen(1, 1, 1, 1, 1).test.plot;
Env.linen(1, 1, 1, 1, 10).test.plot;

Env.linen(1, 1, 1, 1, 'sine').test.plot;
Env.linen(1, 1, 1, 1, 'welch').test.plot;
```

これらのエンヴェロープを用いて、1つの音響素材によるさまざまな音響旋律をつくってみる。

エンヴェロープによる音の分類

エンヴェロープという観点から、音の特徴を大きく以下の5つに分類すると見通しが良くなる。

短音：一瞬の内に鳴り終ってしまい。ピッチや長さの知覚が難しい音(クリック音〜クリックピッチ〜トーンピッチ)。

```
// 短音のエンヴェロープの例
Env.linen(0.001, 0.01, 0.001).test.plot;
```

定常音：継続時間が十分あり、ピッチや音量にあまり変化のない(知覚しやすい)音。

```
// 定常音のエンヴェロープの例
Env.linen(0.001, 3, 0.001).test.plot;
```

減衰音：立ちあがりが鋭く、ゆっくりとした減衰のエンヴェロープを持った音。減衰の過程によってさまざまな変化が生まれる。

```
// 減衰音のエンヴェロープの例
Env.linen(0.01, 0, 5, 1, -10).test.plot;
```

トレモロ音：全体として音が持続しながら、ピッチや音量が周期的に変化する音

```
// トレモロ音のエンヴェロープの例
Env.new(Array.fill(101, {|i| 0.5 + ((i % 2) * 0.2)}), Array.series(100, 0.1, 0)).test.plot;
```

ランダム音：全体として音が持続しながら、音量やピッチが不規則(非周期的)に変化する音。

```
// ランダム音のエンヴェロープの例
Env.new(Array.rand(101, 0.3, 0.8), Array.series(100, 0.1, 0)).test.plot;
```

点による旋律

```
Env.new([0, 1, 0.3, 0.8, 0], [2, 3, 1, 4], [3, -5, 5, -10]).plot;

// 振幅が変化する点列(ゆっくりとしたトレモロ)
(
~out = {
    var env = Env.new([0, 1, 0.3, 0.8, 0], [2, 3, 1, 4], [3, -5, 5, -10]);
    Impulse.ar([Rand(40, 80), Rand(40, 80)], 0, 0.2) * EnvGen.kr(env, doneAction:2)}
)

// 周波数が変化する点列
(
~out = {
    var env = Env.new([0, 1, 0.3, 0.8, 0], [2, 3, 1, 4], [3, -5, 5, -10]);
    Impulse.ar([Rand(40, 80), Rand(40, 80)] * EnvGen.kr(env, doneAction:2), 0, 0.2)}
)
```

線による旋律

```
Env.new([0.9, 0.001, 0.9, 0], [2, 5, 0.1], [-10, 10, -5]).plot;

// 振幅が変化する線(減衰音とその反転)
(
~out = {
    var env = Env.new([0.9, 0.001, 0.9, 0], [2, 5, 0.1], [-10, 10, -5]);
    SinOsc.ar([Rand(4000, 8000), Rand(4000, 8000)], 0, 0.2) * EnvGen.kr(env, doneAction:2)}
)

// 周波数が変化する線
(
~out = {
    var env = Env.new([0.9, 0.001, 0.02, 0], [2, 5, 3], [-10, 10, -5]);
    SinOsc.ar([Rand(4000, 8000), Rand(4000, 8000)] * EnvGen.kr(env, doneAction:2), 0, 0.2)}
)
```

面による旋律

```
Env.new([0, 1, 0.3, 0.8, 0], [1, 4, 0.1, 4], 'welch').plot;

// 振幅が変化する面(ゆるやかな2つの減衰音)
(
~out = {
    var env = Env.new([0, 1, 0.3, 0.8, 0], [1, 4, 0.1, 4], 'welch');
    WhiteNoise.ar(0.2 * EnvGen.kr(env, doneAction:2)).dup }
)

// 振幅が変化する面(速いトレモロ)
(
~out = {
    var env = Env.new(Array.fill(201, {|i| i % 2}), Array.series(200, 0.01, 0));
    WhiteNoise.ar(0.2 * EnvGen.kr(env, doneAction:2)).dup }
)
```

「3秒」は聴覚のチャンクであり、「聴覚的現在」である。

聴覚の基本は時間的な差違である。いいかえれば、聴覚が聴くのは本質的に時間である。

「3秒」は聴覚における「経験のひとかたまり(チャンク)」である。3秒は、人間における現在の長さであり、生物学的には、人間の聴覚バッファーの長さに対応している。

▼エクササイズ：3秒の旋律(音響デッサン)

3秒間のエンヴェロープによってパラメータを時間変化させることで、点(列)、面、線による「3秒の旋律」をデザインせよ。点(列)、

面、線の3つの素材と、定常音、減衰音、トレモロ音、ランダム音などのエンヴェロープを組み合わせ、一筆書きによる素描のような断片的旋律の可能性を探索せよ。

```
// 振幅とパンが0.3秒毎にランダムに変化する面による3秒の旋律(ランダム・リスニング)
(
~out = {
    var enva = Env.new([0]++Array.rand(9, 0.0, 1.0)++[0], Array.fill(10, 0.3));
    var envp = Env.new(Array.rand(11, -1.0, 1.0), Array.fill(10, 0.3));
    Pan2.ar(
        WhiteNoise.ar(0.5), EnvGen.kr(envp, doneAction:2), EnvGen.kr(enva, doneAction:2)
        )
    }
)
```

デジタルサウンドにおける時間の最小単位はサンプリング時間(1/44100秒)である。3秒の音は、44100(サンプル/秒)x3秒x2(チャンネル)=264600個のサンプルに相当する。3秒の旋律を作曲するということは、この264600個のサンプル(数)をデザインする、ということに他ならない。サンプルを意識することで、次の「純正律点列の旋律」のような可能性が生まれる。

▼エクササイズ：点列の旋律

非常に密な(高い周波数の)純正律点列をもとに旋律をデザインせよ。「非常に高い周波数の純正律点列」とは、周波数が22050[Hz](1：2)、14700[Hz](1：3)、11050[Hz](1：4)、8820[Hz](1：5)…の点列である。

```
// 周波数が0.3秒毎に微妙に上昇する点列による3秒の旋律(基本周波数=11050[Hz])
(
~out = {
    var envf = Env.new(Array.rand(11, 0.99, 1.01).sort.postln, Array.fill(10, 0.3));
    Pan2.ar(Impulse.ar(11050 * EnvGen.kr(envf, doneAction:2), 0, 0.5), 0.0, 0.7)}
)
```

非常に疎らな(低い周波数の)点列をもとに旋律をデザインせよ。

```
// 周波数が3秒毎に微妙に上昇する点列による30秒の旋律(基本周波数=1[Hz])
(
~out = {
    var envf = Env.new(Array.rand(11, 1.0, 2.0).sort.postln, Array.fill(10, 3));
    Pan2.ar(Impulse.ar(EnvGen.kr(envf, doneAction:2), 0, 0.5), 0.0, 0.7)}
)
```