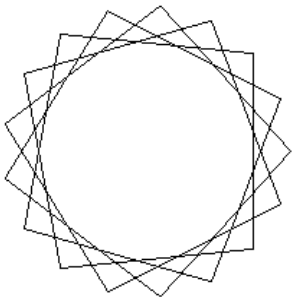
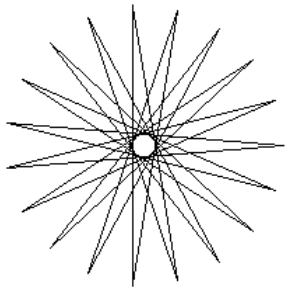
Processing 練習（基本編）

1. 星型多角形

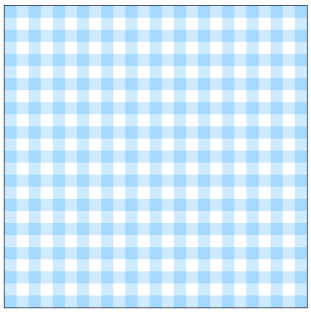
   for 文の中で
   sin, cos を使うだけである。

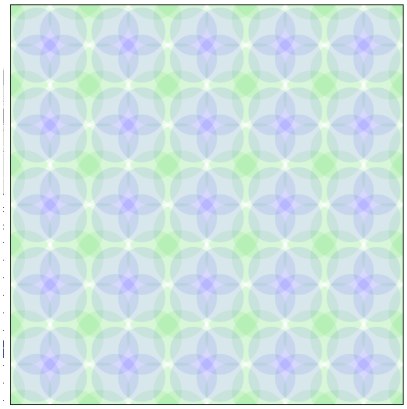




2. 規則的な模様

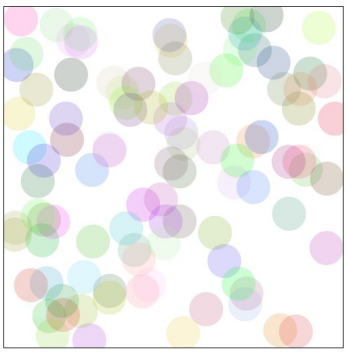
   透明度をうまく使う
   左：簡単
   右：単位格子を考える

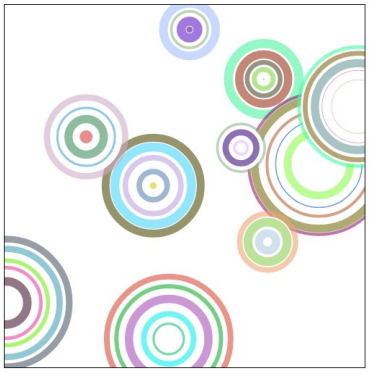




3. 乱数

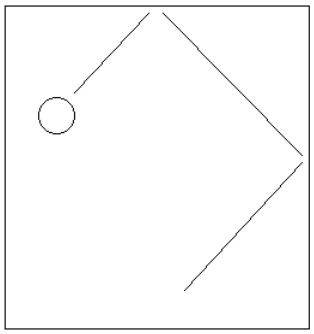
   乱数 random(a, b)で
   a から b までの
   ランダムな小数を得る

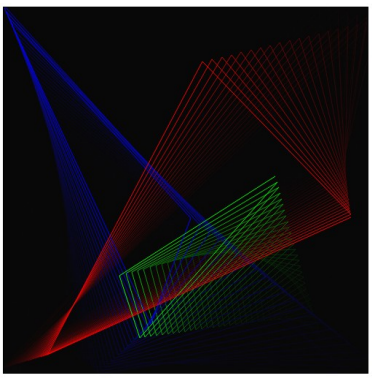




4. アニメーション

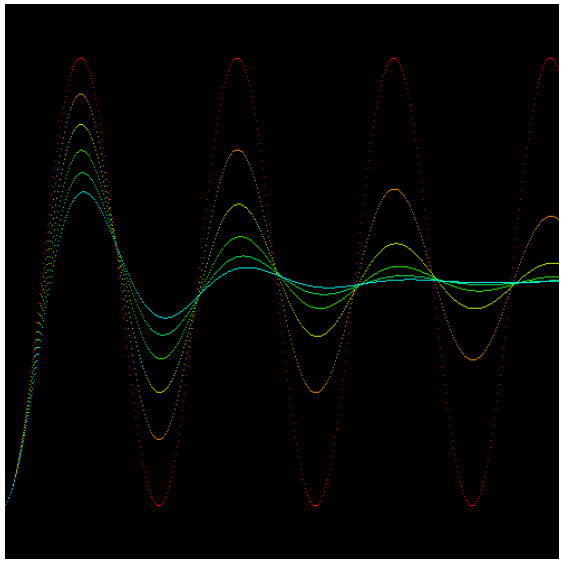
   左：端で跳ね返るボール
   右：ラインアート
   （端で跳ね返るボールを
   いくつか用意して、
   間を直線でつなぐと描ける）





Processing 練習（常微分方程式編）

1. ルンゲ・クッタ法

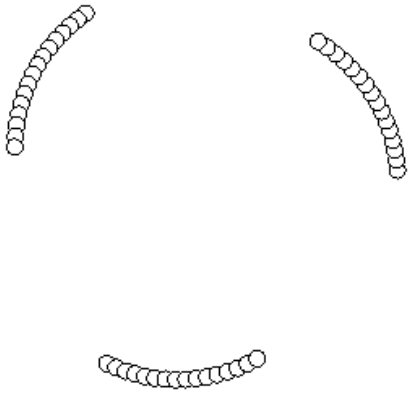


一次元減衰振動の運動方程式を
4 次ルンゲ・クッタ法で解き、
時間発展をグラフにプロットせよ。

差分法が初めての人は
最初は「オイラー法」で書いて良い。

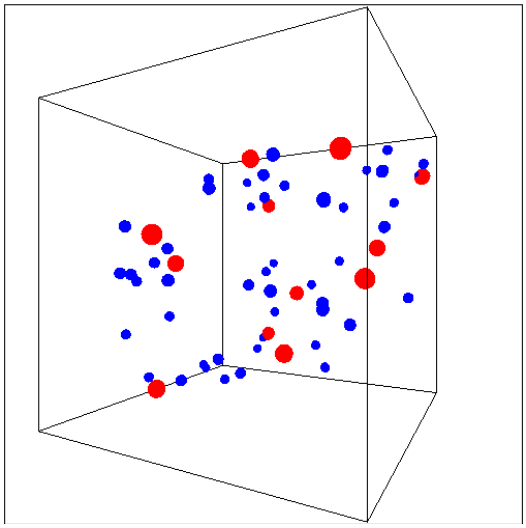
摩擦係数をいくつか変えながら
シミュレーションしてそれぞれ別の色で
同じ画面に表示せよ。

2. 重力多体問題



重力相互作用する3質点系の
運動を「ベルレ法」で解け。
初期位置を正三角形の頂点上、
速度を対称にしてやると、
（しばらくは）同じ軌道を回り続ける
はずである。
（3体問題の定常解のひとつ）

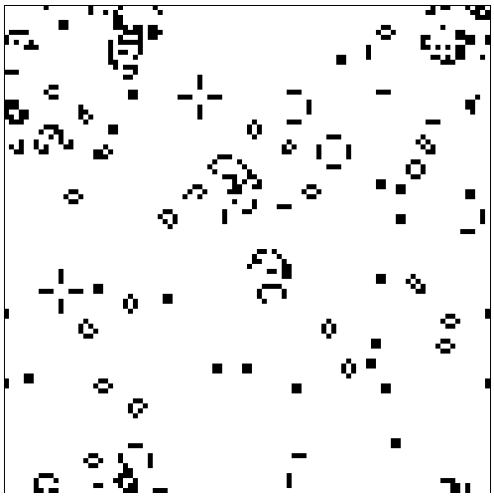
3. 分子動力学法



2.を応用して、相互作用する N 質点系の
運動をシミュレーションせよ。
相互作用ポテンシャルは
クーロン、ファンデルワールスなど
いろいろ試してみよ。
（最初は 2 次元でやろう）

Processing 練習（偏微分方程式編）

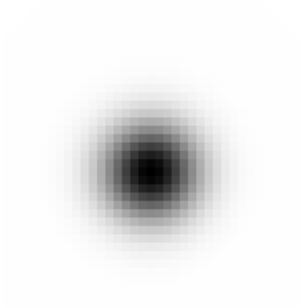
1. ライフゲーム



ライフゲームは偏微分方程式でも
何でもないが、プログラム的には
かなり類似しているので
最初はこれを書いてみよう。

境界は次の2つを試せ。
「固定境界」:端は全部0
「周期境界」:右と左、上と下がつながる

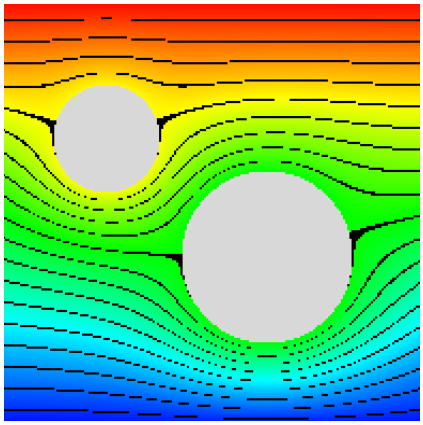
2. 拡散方程式



にじんでいくだけなので面白くないっちゃ面白くないが、
純粋な拡散方程式を解くのはあらゆる偏微分方程式の
なかで一番簡単（問題が起こりにくい）な気がするので、
次はこれを書いてみよう。

まず方程式を差分式に書き換えて、プログラムできる形にする。
拡散係数を変数として d とでもおいておく。
空間、時間刻み幅 dx, dt も適当に決める。
慣れればライフゲームより簡単である。

3. ポアソン方程式



拡散方程式の定常状態は、

$$\partial\Phi/\partial t \propto \triangle\Phi = 0$$

となるので、これはラプラス方程式を満たしている。
この性質より、ラプラス方程式を拡散方程式に見立て、
変化がなくなるまで反復して収束させてやると、
ラプラス方程式の解が求まるのである。
ポアソン方程式

$$\triangle\Phi = f$$

も、右辺におまけがついてるだけなので、
同様の手法で解くことが出来る。

ポアソン方程式は物理のいろんな所に出てくるが、わかりやすい例は、
導体と電場、物体中の完全流体の流れ、などである。
図は上端、下端、灰色の円内で Φ が一定という境界条件のもとで拡散方程式を
収束するまで解いてやったものである。流れっぽい。

Processing 練習（3D 編）

1. 斜めの四角形？

   3D に入る前に、「座標変換」の方法を学んでおいたほうがいい。
   なぜなら、3D では「物体を(ax, ay, az)の位置に置く」ということはあまりせず、
   「物体を(0, 0, 0)において、(ax, ay, az)平行移動する」手法を取ることが多いからである。

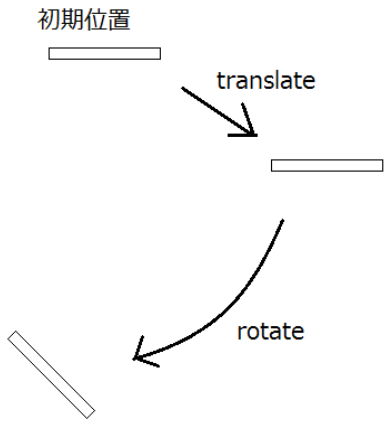
   変換に使うのは、次の3つの命令である。

   translate(x, y)　…　平行移動
   rotate(t)　　…　原点周りの回転（角度 t ラジアン）
   scale(s)　　…　等比拡大（s 倍）

   これらの命令は、命令が書かれた**後のすべての文**に影響する。
   また、**後に書いた方から先**に適用される。
   なぜなら、こいつらは「**変換行列**」だからである。

   次のプログラムを考える。

```
rotate(45.0*2*(float)Math.PI/360);
translate(200, 100);
rect(100, 100, 100, 10);
```



   これは右図のような挙動により、
   右図での下の位置に表示される。
   実際に書いてためしてみるべし。

   さて、さっき「変換命令は
   後に書いた文全てに適用される」と書いたが、
   もちろんこのままでは不便極まりない。
   そこで、次の2つの命令がある。

   pushMatrix();
   popMatrix();

   これらの命令は常に対で使い、この2つに挟まれた行に書かれた命令は、
   この2つの間だけで効力を発揮するようになる。これらの外側には影響を及ぼさない。

2. 3D 教材について

   某塾で教えた時のテキストをアップしておいたのでこれにそって勉強してみてください。
   中学生対象なので易しいですが悪しからず…
   http://www18.atwiki.jp/physlab_keisanki/pages/30.html