

成長する線分要素の挙動に関して

藤本將太郎

2016 年 7 月 7 日

1 研究背景

- 本田先生の研究
- 脇田先生の研究
- 2 次元平面内での界面の運動の解析
- 軸を定めて 1 つの関数で表すことのできないときの挙動の記述の可能性

2 研究目的

- 2 次元平面内でのひも状構造体の挙動に関する実際の物理を再現するモデルを作成し、現象の背後にある統計性などに注目できるフレームワークを作成する (?)

3 研究方法

本研究では、*Bacillus subtilis* が養分濃度が高く、固い寒天上で成長するとき、その成長初期段階に見られる座屈と折りたたみ現象の数理的解析を行う。また、簡単な数理モデルを作成してシミュレーションを行うことにより、現象の理解を試みる。

4 モデル化

4.1 ひも状細胞を弾性体としてとらえるモデル

4.1.1 概要

2 次元平面内でのひも状細胞の挙動の理解のために、この対象をモデル化して、 N 個の質点がそれぞれバネで 1 次的に繋がれているような状況を考える (図 1)。

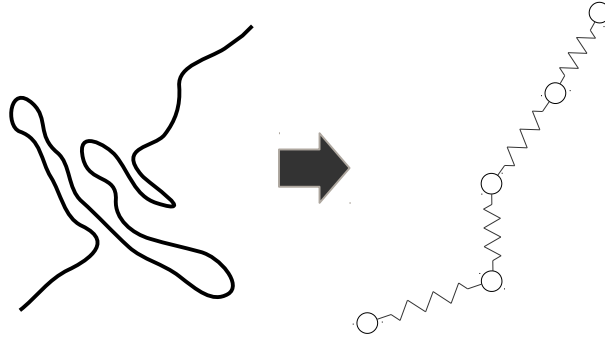


図1 ひも状細胞の質点とばねによるモデル化

このとき、これらの質点間に働く力は、バネの伸びによるフック則に従う力 F^s と、近接点との位置関係によって生じる曲げ弾性による力 F^b である。また、これらの力の他に、速度に比例する力として粘性抵抗 (粘性係数 γ) による力が働くとする。

ひも状細胞の成長を再現するために、各ばねの自然長が一定ではなく、時間の関数として増大していくとし、これがある閾値を超えた時にそのばねの間に新たに一つ質点を追加することによって成長過程を記述することにする。

4.1.2 各質点に対する運動方程式

i 番目 ($i \in [0, N-1]$) の質点に関する運動方程式は、以下のように表される。

$$m\ddot{\vec{x}}_i = F_i^s + F_i^b - \gamma\dot{\vec{x}}_i \quad (1)$$

$$F_i^s = -k_{i-1}(d_{i-1} - n_{i-1}) + k_i(d_i - n_i) \quad (1.a)$$

$$\begin{aligned} F_i^b = E_i & \left(\frac{\vec{x}_{i-1} + \vec{x}_{i+1}}{2} - \vec{x}_i \right) \\ & - \frac{1}{2}E_{i-1} \left(\frac{\vec{x}_{i-2} + \vec{x}_{i+2}}{2} - \vec{x}_{i-1} \right) \\ & - \frac{1}{2}E_{i+1} \left(\frac{\vec{x}_i + \vec{x}_{i+2}}{2} - \vec{x}_{i+1} \right) \end{aligned} \quad (1.b)$$

\vec{x}_i : i 番目の質点の位置ベクトル

k_i : i 番目と $i+1$ 番目の質点間のバネのバネ定数

n_i : i 番目と $i+1$ 番目の質点間のバネの自然長

d_i : i 番目と $i+1$ 番目の質点間の距離

E_i : $i-1, i, i+1$ 番目の3つの質点により決まる曲げ弾性係数

ここで、ひもの先端と終端がつながっている場合 (これを閉な場合と呼ぶことにする) には k_{-1} や n_{-1} , d_{-1} , E_0 , E_N などはラベル0の質点とラベル $N-1$ の質点の間でそれぞれ定義されるものである。一方で、ひもの先端と終端が繋がれていない場合 (これを開な場合と呼ぶことにする) には、これらの値は定義されず、 $k_{-1} = E_0 = E_{N-1} = 0$ と再定義することにする。

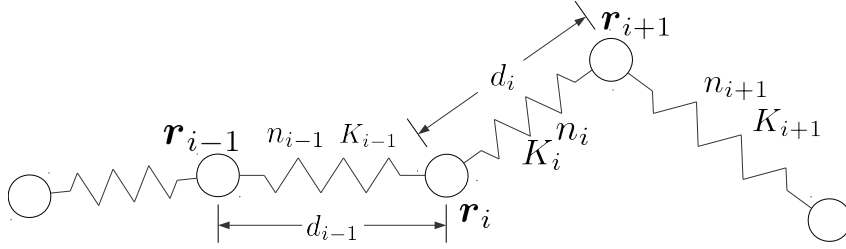


図2 質点とばねに関する物理量

シミュレーションでは実際にこの運動方程式を4次のルンゲクッタ法もしくはオイラー法で計算することによって時間発展を逐次計算していく。

ただ、それぞれの質点要素に関して同時に計算を行うことができるので、力の項の計算を行列として計算するほうが効率的である。

したがって、これから式(1), (1.a), (1.b)を行列による表記に書き直す。 x 座標に関する配列を \mathbf{x} , y 座標に関する配列を \mathbf{y} , x 方向の速度の配列を $\dot{\mathbf{x}}$, y 方向の速度の配列を $\dot{\mathbf{y}}$ としてこれらの列ベクトルに、ある行列が作用しているとする。

F_i^s を x 成分と y 成分で分けて考えると、

$$F_{x_i}^s = Z_i \cdot \mathbf{x}, \quad F_{y_i}^s = Z_i \cdot \mathbf{y}$$

このとき

$$Z_i = \begin{pmatrix} \cdots & 0 & -k_{i-1} \left(\frac{n_{i-1}}{d_{i-1}} - 1 \right) & k_{i-1} \left(\frac{n_{i-1}}{d_{i-1}} - 1 \right) + k_i \left(\frac{n_i}{d_i} - 1 \right) & -k_i \left(\frac{n_i}{d_i} - 1 \right) & 0 & \cdots \end{pmatrix}$$

よって

$$F_x^s = Z \cdot \mathbf{x}, \quad F_y^s = Z \cdot \mathbf{y}.$$

表記の省略のために、 $z_i \equiv k_i \left(\frac{n_i}{d_i} - 1 \right)$ とおくと

$$Z = \begin{pmatrix} z_{-1} + z_0 & -z_0 & 0 & \cdots & \cdots & 0 & -z_{-1} \\ -z_0 & z_0 + z_1 & -z_1 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & -z_{N-3} & z_{N-3} + z_{N-2} & -z_{N-2} \\ -z_{-1} & 0 & \cdots & \cdots & 0 & -z_{N-2} & z_{N-2} + z_{N-1} \end{pmatrix}.$$

ここで、

$$z = \begin{pmatrix} z_{-1} & 0 & \cdots & \cdots & 0 \\ 0 & z_0 & 0 & \cdots & 0 \\ 0 & \cdots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & z_{N-3} & 0 \\ 0 & \cdots & \cdots & 0 & z_{N-2} \end{pmatrix}, \quad z^u = \begin{pmatrix} 0 & z_0 & 0 & \cdots & 0 \\ 0 & \cdots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & z_{N-3} & 0 \\ 0 & \cdots & \cdots & 0 & z_{N-2} \\ z_{-1} & 0 & \cdots & \cdots & 0 \end{pmatrix}$$

$$z^l = \begin{pmatrix} 0 & \cdots & \cdots & 0 & z_{-1} \\ z_0 & 0 & \cdots & 0 & 0 \\ \cdots & \ddots & \cdots & 0 & 0 \\ \cdots & 0 & z_{N-3} & 0 & 0 \\ \cdots & \cdots & 0 & z_{N-2} & 0 \end{pmatrix}, \quad z^{ul} = \begin{pmatrix} z_0 & 0 & \cdots & \cdots & 0 \\ 0 & z_1 & 0 & \cdots & 0 \\ 0 & \cdots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & z_{N-2} & 0 \\ 0 & \cdots & \cdots & 0 & z_{N-1} \end{pmatrix}$$

のように、上方向に要素をずらしたものを z^u 、左方向にずらしたものを z^l 、上と左に一回ずつずらしたものを z^{ul} と表すことにすると、

$$Z = z + z^{ul} - z^u - z^l$$

である。

同じように、曲げ弾性による力 F_i^b を x 成分と y 成分で分けて考えると、

$$F_{x_i}^s = B_i \cdot \mathbf{x}, \quad F_{y_i}^s = B_i \cdot \mathbf{y}$$

$$B_i = \begin{pmatrix} \cdots & 0 & -\frac{1}{4}E_{i-1} & \frac{1}{2}E_i + \frac{1}{2}E_{i-1} & -E_i - \frac{1}{4}E_{i-1} - \frac{1}{4}E_{i+1} & \frac{1}{2}E_i + \frac{1}{2}E_{i+1} & -\frac{1}{4}E_{i+1} & 0 & \cdots \end{pmatrix}$$

であり、これを z と同様に行列 e

$$e = \begin{pmatrix} E_{-1} & 0 & \cdots & \cdots & 0 \\ 0 & E_0 & 0 & \cdots & 0 \\ 0 & \cdots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & E_{N-3} & 0 \\ 0 & \cdots & \cdots & 0 & E_{N-2} \end{pmatrix}$$

と、その上下左右へ要素を平行移動した行列 $e^l, e^d, e^r, e^u, e^{ld}, e^{rd}, e^{lu}, e^{ru}$ を考えると、

$$B = -\frac{1}{4}(e^{ld} + e^{rd} + e^{lu} + e^{ru}) + \frac{1}{2}(e^l + e^d + e^r + e^u) - e$$

となる。

4.1.3 自然長の増大とばねの分割規則

また、このシミュレーションではひも状構造体の成長を記述したいので、これを実現する方法として単位時間ごとにそれぞれのバネの自然長が増大するようにし、自然長がある閾値を超えた際には、その線分要素を内分する新たな点を追加し、2つの線分要素に分裂するようにする。

ここで、 n_k を時刻 t_k , ($k = 0, 1, \dots, n$) における自然長とし、単位時間あたりの自然長の成長率 α として以下のように線形に成長していくものとする:

$$n_{k+1} = n_k + \alpha \Delta t_k$$

(実際のバクテリアの成長の場合、細胞が連なってできたファイバーの長さは指数関数的に増大していくことが知られている (Honda, Wakita and Katori, 2015)。今の場合は線分の長さの成長率を規定してはおらず、その自然長を指定するものとなっているため、自然長も指数関数的に増大するという必然性はない。しかし、

もし緩和時間よりも遅く成長しているような場合を考えると、自然長の成長は線分の成長とみなしても良いだろう。また、離散時刻を連続時間に変換する際に、適当なスケール変換 ($\Delta t_k \rightarrow n_k \Delta t$) を行えば、時間に対して指数関数的に成長しているとみなすことができる。))

また、自然長を増大させていく時、線分の単位長さあたりのエネルギーが保存されるように、各時刻におけるバネ定数 K_k を自然長の長さ n_k を用いて

$$K_{k+1} = K_k n_k / n_{k+1}$$

のように補正することにする。

自然長がある一定の閾値 (n_c) を超えたとき、そのばねを内分する点を新たに一つ追加する。点を追加する際に更新される物理量は以下の3つである。

- 質点の速度
- 自然長
- バネ定数

今、一番単純な例として二つの質点の中点の座標に新たな質点を追加することにする。即ち、 $\mathbf{r}_i = (x_i, y_i)$ と $\mathbf{r}_{i+1} = (x_{i+1}, y_{i+1})$ の間に追加される点の座標を $\mathbf{r}_j = (x_j, y_j)$ とすると

$$\mathbf{r}_j = \frac{\mathbf{r}_{i+1} + \mathbf{r}_i}{2}$$

質点の速度に関しては、新たに追加する点の速度は、点が追加される両端の2点の速度の平均とする。また、このままでは運動量の保存が成り立たなくなってしまうため、新たに点が追加された両端の点は、追加した点に与えた分だけ運動量を失うとして、速度の補正を行う。即ち両端の2点の(ステップ k での)速度を $\mathbf{v}_i^k, \mathbf{v}_{i+1}^k$ 、挿入される点の速度を \mathbf{v}_j^{k+1} として

$$\begin{cases} \mathbf{v}_j^{k+1} &= \frac{\mathbf{v}_i^k + \mathbf{v}_{i+1}^k}{2} \\ \mathbf{v}_i^{k+1} &= \mathbf{v}_i^k - \frac{1}{2} \mathbf{v}_{i+1}^k \\ \mathbf{v}_{i+1}^{k+1} &= \mathbf{v}_{i+1}^k - \frac{1}{2} \mathbf{v}_i^k \end{cases}$$

自然長の更新に関しては、追加された点によって分断された二つのばねの長さがそのまま自然長となるように、ちょうど中点で分裂する今の場合分割されてできる二つのばねの自然長は、分裂前のばねの自然長の $1/2$ となる。即ち分裂前の自然長を n_i^k 、分裂によってできた二つのばねの自然長を n_j^{k+1}, n_{j+1}^{k+1} のようにすると、

$$n_j^{k+1} = n_{j+1}^{k+1} = \frac{n_i^k}{2}$$

バネ定数については、分裂の前後でエネルギーの増減があってはならないという仮定の下では分裂後のそれぞれのバネ定数 (K_j^{k+1}, K_{j+1}^{k+1}) は、分裂前のそれ (K_i^k) の2倍となる。

$$K_j^{k+1} = K_{j+1}^{k+1} = 2K_i^k$$

4.1.4 結果 1

これまでにあげた条件で、2次元平面上に質点をばねでつないだ鎖状オブジェクトを配置し、パラメータを与えて実際にシミュレーションを行った。

適切なバネ定数や曲げ弾性係数を設定しないと有限時間内に観測している領域から勢い良く飛び出していくような発散が見られた。

初期状態で各ばねに与えているバネ定数や自然長が一定なとき、分裂のルールから必ず同じタイミングで分裂が起こるため、ばねの本数は倍々に増えていく様子が観察される。これは実験系においてひも状細胞の全長が指数関数的に増大していることと対応すると言えるかもしれない。

しかしながら、適当なパラメータを設定できていると思われるときであっても、このモデルではばねの重なりを許しているため、実際の実験系に見られる、1 回折れ曲がってできた 2 本のひも状構造がくっつきながら成長していくような振る舞いは観察されなかった。また、シミュレーションにおけるスケールの小ささの問題もあるかもしれないが、実験のように折れ曲がるときの曲率半径が小さくなく、より緩やかに弧を描いて折れ曲がっている状況が観察される。このような振る舞いが見られるのは実際の系に比べて曲げ弾性の大きさが大きいからだと言えるかもしれない。

4.1.5 重なりを許さない自己排他的モデル

前述のモデルは、結果 1 でも確かめられたように、重なりを許すモデルだったために、折れ曲がりの際の曲率半径が実験よりも相対的に大きくなり、折れ曲がった 2 本のひも構造が互いに広い範囲で接触したまま成長していくような振る舞いは観察できなかった。したがって、重なりを許さないようにルールを変更した時の振る舞いがどうなるか、というのは当然興味もたれる。

ここからは、各ばねの交差が起らないようにルールを追加し、その結果を見てみることにする。

まず、質点の位置の更新は同時に行われるので、そのタイミングで交差判定を行い、交差している箇所について、その交差を解消するように新しく移動後の点を取り直す。一箇所この操作を行った後、再度交差判定を行い、最終的に交差した箇所がなくなるまでこの過程を続ける。また、この間に時間ステップは進めず、最終的な配置が決定してから時間ステップを 1 進めることにする。

4.2 自然長、バネ定数を一定に保つモデル

4.2.1 概要

自然長を増大させるモデルにおいて、線分の交差判定を行う際に困難が生じた。

成長させているのは自然長であるため、一般に剛体のシミュレーションとして行うときのような剛体球ポテンシャルは有効でなく、したがって、実際に点を動かす前にその次のステップで交差するかどうかを判定した上で、もし交差しそうなら交差しないように次のステップで移動する点の位置を変更する作業が必要だった。しかしながら、(もう少し頑張れるかもしれないが) この試みはうまくいかなかったので、今度は発想を変えて自然長、バネ定数 K は一定に保ったまま、ある時間間隔ごとにランダムに点を 1 つずつ追加していくことによって線素群の成長を記述することを試みる。また、自己回避的な挙動を示して交差を防ぐために、各点ごとに斥力ポテンシャルを設け、交差が起らないようにする。

4.2.2 結果

実際にシミュレーションとして作動させてみたが、うまいパラメータの設定ができていないのか、期待通りに動かなかった。

また、質点間距離はある程度抑えられているとはいえ、互いの距離を一定に保ったまま動的に動かすことは難しい。

(要確認 & 再挑戦)

4.3 三角格子上で動くひも状オブジェクト

4.3.1 概要

一番初めのモデルで、自己排他性をうまく取り入れることができなかったために、2つ目のモデルでは初めから質点間の距離を一定に保つようなルールを取り入れておいて、その上で各質点に斥力ポテンシャルを導入することで自己排他性を持たせつつ、点をランダムに追加していくという方法で線分の成長を表現することとした。

この考えをさらに推し進めてみると、質点の位置を2次元のユークリッド空間における点ではなく、各格子点同士の距離が等しくなるような格子における格子点とすれば、隣接する質点とは必ず等しい距離に保つことができる。また、同じ格子点を占めることのできる質点の数を1つに限定することによって、自己排他性も担保されていることから、2つめのモデルで考慮しなかった条件は常に満たされるという意味で、格子状の点を考えることには意味があるように思われる。

ここでは、4回対称性を持つ正方格子よりも、6回対称性をもつ三角格子の方が取りうる線素のパターンの数は多いということから、三角格子上でひも状構造を作り、その成長をモデル化することを目的とする。

4.3.2 三角格子の作成

プログラム上で三角格子を自然に定義しようとするならば、扱いやすい正方格子をもとに、それを変形したものとして三角格子を扱うのが良いだろう。即ち、図3のように、正方格子の隣接関係にプラスしてさらに二つの格子点と隣接関係にあり、それが図の番号のように対応しているとみなすことである。このようにしておけば実際のデータ構造としては2次元の行列上に情報を保存しておきつつ、各格子点の隣接関係を参照できるようにしておくことで、描画などの際には三角格子として表現することが可能となる。

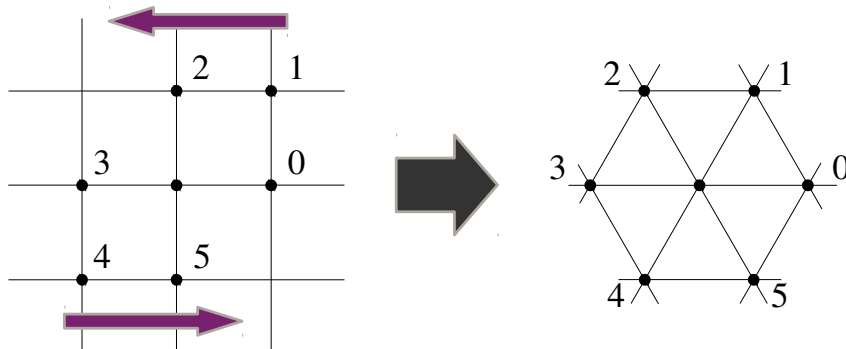


図3 2次元正方格子を三角格子に対応させるための隣接関係の再定義

またこのとき、三角格子は周期境界条件を満たすようにする。

4.3.3 三角格子状で定義されるひも状構造

上のようにして作成された三角格子上で定義されるひも状オブジェクトの表し方を以下のように定める。

まず、始点(頭)となる格子点の座標(正方格子における(行, 列))を定める。次に、始点から次の格子点へ

のベクトルを、図 4 に示したような 6 方向の単位ベクトル \mathbf{u}_α ($\alpha \in \{0, 1, \dots, 5\}$) のうちのひとつで表現する。同様に、2 つ目の点に向かうベクトルを指定する。このようなベクトルをオブジェクトの構成要素数だけつなげれば、全体を表すことができる。また、この方法の利点は、格子を占有している点がどのひも状オブジェクトに属するものなのか、ベクトルを遡ることで特定できるということである。

したがって、一つのひも状オブジェクト S は以下で決定される。

$$S = ((x, y), (\alpha_1, \alpha_2, \dots, \alpha_{N-1})), \alpha_i \in \{0, 1, \dots, 5\}$$

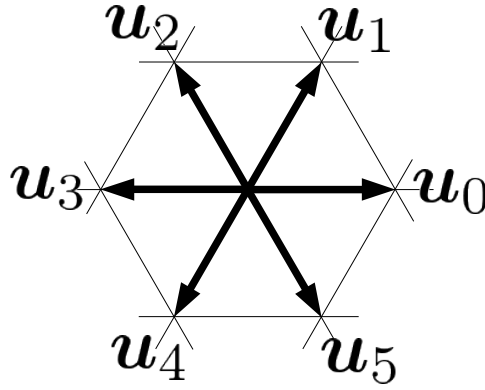


図 4 6 つの基本ベクトル

4.3.4 ひも状構造の配置

次に、ひも状構造を実際に三角格子の中に配置することを考える。これは一見何でもないのであるが、実際はオブジェクトの要素数が増大するにつれて、重なりなく配置することは難しくなってくる。

これは、三角格子中の自己回避型ランダムウォーク (SAW) の問題として捉えることができる。まず、三角格子全体の中から占有されていない点をランダムに一つ選び出す。次に、選択された点の周囲の 6 つの点のうち占有されていないものへ向かうベクトルを一つランダムに決定する。この過程を繰り返していくと、自己排他的なランダムウォークパターンを形成することができる。しかしながら、構成要素の数が一定 (格子中にひとつのひも状オブジェクトがあるとして、7 個) 以上になると、周りの点すべてが占有されており、それ以上点を増やすことができないという状態 (この状態のことを deadlock 状態と呼ぶことにする) となることがある。今は指定した個数の要素でオブジェクトを構成したいので、deadlock となった時には、その一つ前の要素を配置する時点までさかのぼり、deadlock した方向のベクトル以外で占有されていない点に向かうベクトルを選ぶ。この過程自体も再帰的であり、もしある点から進んだすべての点で deadlock となるようであれば、その状態もまた deadlock 状態であるとし、さらに一つ前のステップまで遡る。もしすべてのパターンを試行しても deadlock となる場合 (即ち初期点で deadlock となる場合)、初期点を選ぶところからやり直すことにする。

4.3.5