

# 仮想化した細胞間モデルの安定性解析

幸谷智紀\* 大梶弘順\*

本稿では、仮想化及び簡易化した細胞間モデルの安定性解析を行った結果を報告する。このモデルは常微分方程式として表現されるもので、いわゆる“Robust”であることが要求されている。我々は、このモデルが“Robust”であるということが、漸近安定 (asymptotic stable) な平衡点を持つことと同義であることを、多倍長数値計算ライブラリ BNCpack を用いた安定性解析を行うことによって確認することが出来た。

## Stability Analysis for a Virtualized Cell-cell Interface Model

Tomonori Kouya\* Kojun Ohsugi\*

In this paper, we describe the result of stability analysis for a simplified and virtualized cell-cell interface model. This model is expressed as a system of ODEs, which is also required to be so-called “robust.” In consequence of analyzing the stability of this model using BNCpack, our multiple precision numerical computation library, we can confirm that the robustness of this model is equivalent to having a asymptotically stable equilibrium.

### 1. 初めに

Segment Polarity Network Model (SPN モデル) は、ショウジョウバエの卵が孵化するまでの過程の一時期における、細胞内及び細胞間の mRNA (messenger RNA)・蛋白質の相互関係を常微分方程式 (ODE) としてモデル化したものである。一つの細胞 (cell) は六角形で表現され、平面状に並べられているものとする。この相互関係を図 1 に示す。mRNA は楕円、蛋白質は長方形として示されている。

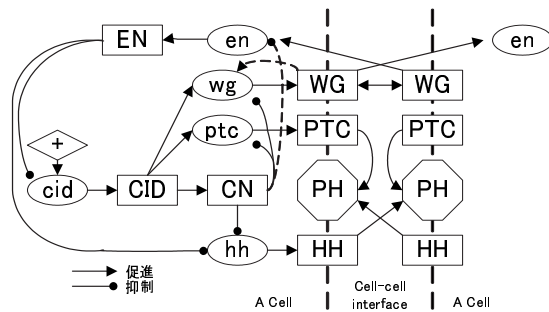


図 1: Segment Polarity Network Model

この SPN モデルにおける促進・抑制関係はフィードバック制御の形で表現される (図 2(左))。これを簡易化し、仮想的に mRNA とそれが生成する蛋白質を一体のものとして扱ったのが大梶らによる簡易化・仮想化した細胞間再生モデル [3] である (図 2(右))。

従来、これらのモデルにおいては“Robust”な系にするために、多数のパラメータを ODE の数値計算によって推定するという方法が取られてきた。しかし、この“Robust”という概念が数値計算によってし

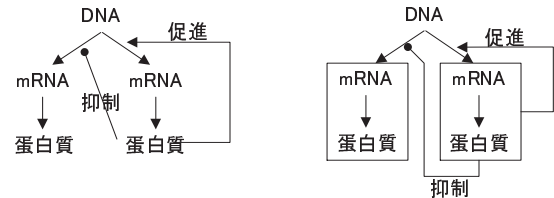


図 2: 細胞間再生モデル (左) と簡易化・仮想化モデル (右)

か推定されない以上は、その系が数学的な意味で安定 (Stable) であることは示されず、故に“Robust”という曖昧な用語を導入せざるを得なかったのではないかと、という疑いが残る。純粋に理論的な証明は難しくとも、数値計算である程度のレベルの安定性のチェックを行うことは出来ないものだろうか、というのが本研究の動機である。

### 2. 仮想化・簡易化モデルの構成

細胞間再生モデルの考え方を数値例も含めて説明する。

円環状に連なった  $N$  個の細胞にそれぞれにおいて蛋白質グループ  $x_i(t), y_i(t)$  が存在しているものとする。

これらの蛋白質グループの各要素は、他の要素によって生成を促進されたり抑制されたりする。もしある蛋白質  $z(t)$  が、別の複数の蛋白質の働きによって生成が促進される場合、これらを線型結合した  $u(t)$  によって

$$\frac{d}{dt}z(t) = \frac{(u(t))^n}{b_u + (u(t))^n} - a_z z(t) \quad (1)$$

と表現される。逆に、生成が抑制される場合は、抑

\*静岡理科大学  
 \*Shizuoka Institute of Science and Technology

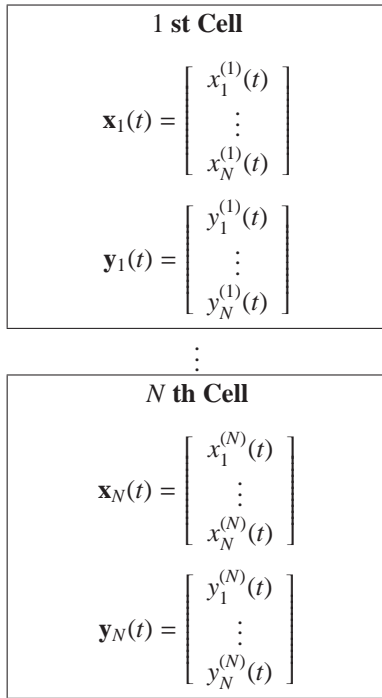


図 3: 蛋白質グループ

制効果のある要素の線型結合  $v(t)$  を用いて

$$\frac{d}{dt}z(t) = \frac{1}{b_v + (v(t))^m} - a_z z(t) \quad (2)$$

と表現される。ここで、 $a_z z$  は自壊する量、各パラメータ  $a_z, b_u, b_v \in \mathbb{R}, m, n \in \mathbb{N}$  は正定数である。

このような促進・抑制効果をまとめて記述すると(4)式のようになるが、これを一括して

$$\frac{d}{dt}\mathbf{Y}(t) = \mathbf{F}(\mathbf{Y}(t)) \quad (3)$$

と書くことにする。

ここで、 $f_j^{(i)}(t), g_j^{(i)}(t), h_j^{(i)}(t)$  は各蛋白質成分の線型結合で表現される関数である。今回は図 4( $N = 5$  の場合) に示されるような促進・抑制関係を使用するので、これらの関数は

$$f_j^{(i)}(t) = x_{j-2}^{(i-1)}(t) + x_{j+2}^{(i-1)}(t) + x_{j-2}^{(i+1)}(t) + x_{j+2}^{(i+1)}(t)$$

$$g_j^{(i)}(t) = \sum_{k=1, k \neq j}^N y_k^{(i)}(t)$$

$$h_j^{(i)}(t) = y_j^{(i-1)}(t) + y_j^{(i+1)}(t)$$

となる。但し、1 番目と  $N$  番目の細胞が繋がって円環状になっているため、この添字は

$$i - 1 < 1 \quad \text{の時は} \quad (i - 1) + N$$

$$i + 1 > N \quad \text{の時は} \quad (i + 1) - N$$

$$j - 2 < 1 \quad \text{の時は} \quad (j - 2) + N$$

$$j + 2 > N \quad \text{の時は} \quad (j + 2) - N$$

となる。

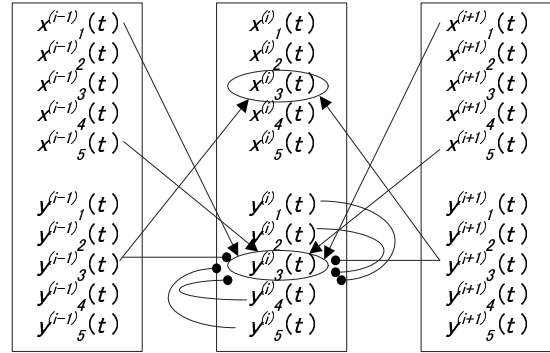


図 4:  $N = 5$  の場合の促進・抑制関係図

このモデルに必要なとされる性質は次のようになる。

1. 十分時間が経過した時点 ( $t > t_c$ ) において

$$\frac{y_i^{(i)}(t)}{y_j^{(i)}(t)} \geq 10 \quad (j \neq i)$$

となる。

2.  $t_d > t_c$  において、一定数以下の細胞が死滅 ( $\mathbf{x}_i(t_d) = 0, \mathbf{y}_i(t_d) = 0$ ) しても再生する(上記の条件を満足する)。

特に 2 の性質は重要であり、これが「細胞間再生モデル」という名前の由来となっている。一部の細胞が死滅すると、その部分はどの細胞にもなりうるオールマイティな幹細胞に入れ替わる。一定時間経つとこの幹細胞が元の細胞の性質を持つようになる(図 5)。数学的には  $i$  番目の細胞ならば、蛋白質  $y_i^{(i)}(t)$  がその他の  $y_i(t)$  の成分に比べて 10 倍以上優越するようになることを意味する。

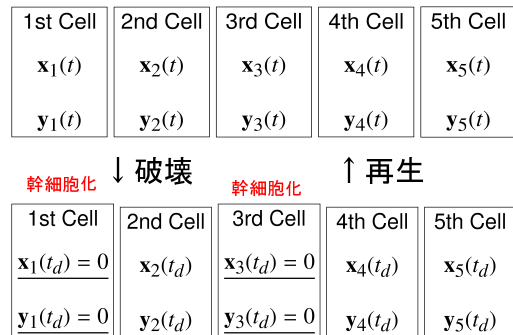


図 5: 細胞の再生

$$\frac{d}{dt} \begin{bmatrix} \vdots \\ \mathbf{x}_i(t) \\ \mathbf{y}_i(t) \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \frac{(h_j^{(i)}(t))^{p_x}}{b_x + (h_j^{(i)}(t))^{p_x}} - a_x x_j^{(i)}(t) \\ \vdots \\ \frac{(y_j^{(i)}(t))^{p_y} + c_f (f_j^{(i)}(t))^{p_f}}{b_y + (y_j^{(i)}(t))^{p_y} + c_f (f_j^{(i)}(t))^{p_f} + c_g (g_j^{(i)}(t))^{p_g} + c_h (h_j^{(i)}(t))^{p_h}} - a_y y_j^{(i)}(t) \\ \vdots \end{bmatrix} \quad (4)$$

(正定数 :  $a_x, a_y, b_x, b_y, c_f, c_g, c_h \in \mathbb{R}, p_x, p_y, p_f, p_g, p_h \in \mathbb{N}$ )

### 3. 簡易化・仮想化モデルの数学的性質

これらの現象を数値実験で示す。パラメータとしては大塚ら [3] が使用した

$$\begin{aligned} b_x &= 1000, a_x = 10, p_x = 3 \\ b_y &= 10^{-9}, a_y = 10^{-1}, p_y = 1 \\ c_f &= 10^{-1}, p_f = 3, c_g = 1, p_g = 3 \\ c_h &= 10^{-9}, p_h = 2 \end{aligned}$$

を用いる。  $t = 0$  における初期条件は各  $i$  に対して、

$$\begin{aligned} y_i^{(i)}(0) &> y_j^{(i)}(0) > 0 \quad (j \neq i) \\ 0 &< y_i^{(i)}(0) - y_j^{(i)}(0) < 100 \quad (j \neq i) \end{aligned}$$

となるようランダムに取った。

まず、1 の条件が満足されることを確認してみた。積分区間は  $[0, 300]$  である。図 6 は 1 番目の細胞の各成分をプロットしたものである。上が  $y_1(t)$  であり、確かに  $y_1^{(1)}(t)$  が他の成分に比べて優越して安定することが分かる。参考までにその下に  $x_1(t)$  のグラフを掲載しておく。スケールが小さくて分かりづらいが、幾つかの成分がある程度は優越して安定することが見て取れる。

次に、 $t = 100$  の時点で 1 番目と 3 番目の細胞を死滅させ、これらの細胞が再生されるかどうかを確認した計算結果を図 7 に示す。各細胞を特徴付ける成分のみを抜き出してプロットしたものを比較のために上に掲載してある。細胞が死滅したことは、この時点で  $[\mathbf{x}_1(100) \mathbf{y}_1(100)]^T = [\mathbf{x}_3(100) \mathbf{y}_3(100)]^T = 0$  と強制的に値を代入することで表現される。これを行った結果が下のグラフである。確かに一定時間後に再生していることが分かる。但し、再生に必要な時間は初期値に依存して決まり、一定ではない。

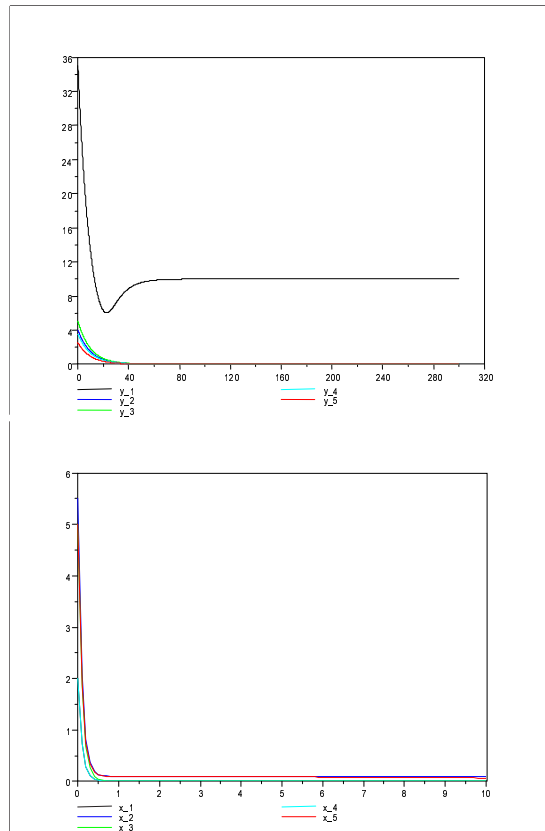


図 6:  $N = 5$  の時の  $y_1(t)$ (上) と  $x_1(t)$ (下)

$N = 2$  の場合，安定した解の値は

$$\begin{aligned}
 \mathbf{Y}(100) &= \begin{bmatrix} \mathbf{x}_1(100) \\ \mathbf{y}_1(100) \\ \mathbf{x}_2(100) \\ \mathbf{y}_2(100) \end{bmatrix} \\
 &= \begin{bmatrix} 1.01209790265282372e - 19 \\ 8.88895681936576754e - 02 \\ 1.00002269813700000e + 01 \\ 4.96988119049839271e - 06 \\ 8.88895681936576754e - 02 \\ 1.01209790265282372e - 19 \\ 4.96988119049839271e - 06 \\ 1.00002269813700000e + 01 \end{bmatrix} \quad (5)
 \end{aligned}$$

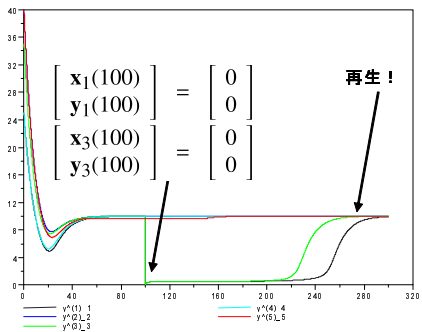
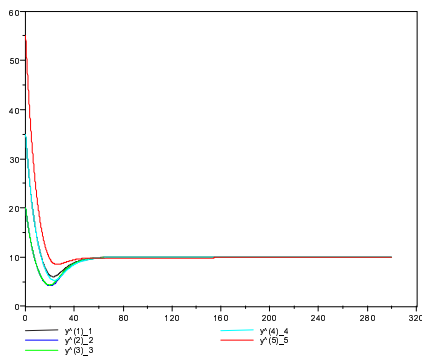


図 7: 安定した解 (上) と再生した解 (下)

となる。安定化の後，細胞の死滅があっても再生してこの値に戻ることを考えると，これが平衡点で，その周囲ではかなり強い漸近安定性を持つと考えられる。

#### 4. ODE の数値計算法によるベンチマーク

簡易化モデルは前述のように次元数が  $2N^2$  となり，細胞数  $N$  が増えると大次元な ODE を解く必要が出てくる。今回は比較的小規模なケースで予備的な数値実験とベンチマークを行う。

使用した環境は次の通りである。

#### 使用ソフトウェア

- BNCpack(IEEE754 double/mpfr)
- MPIBNCpack(IEEE754 double/mpfr)

#### コンピュータ環境

- Pentium IV 2.8GHz/512MB RAM
- Vine Linux 2.6r4

#### PC Cluster

- Pentium IV 2.8GHz × 11 nodes
- Vine Linux 2.6r4
- mpich 1.2.5

ハードウェアは Pentium IV 2.8GHz の PC である。並列計算に際してはこれらを 1 Node とした PC cluster(MPI) を用いた。BNCpack 及び MPIBNCpack によるベンチマークでは，4 段 (8 次相当) の GBS アルゴリズム [1] を用いている。

ベンチマーク結果を表 1 に示す。予想通り Scilab が低速であったが，多倍長計算よりも遅いという結果は意外であった。これは被積分関数に引数のチェックや配列の値抽出などオーバーヘッドが大きくなる要素が多く入っているためと思われる。

BNCpack を用いた計算は，関数評価回数の多い補外法を用いているにも関わらず，かなり高速であ

表 1: ベンチマーク結果 ( $N = 5$ (上) と  $N = 10$ (下) の場合)

N=5, DIM=50					
BNCpack/MPIBNCpack					
	double	50 桁	100	200	500
1CPU	0.82	7.21	10.15	21.78	70.12
5CPUs	6.50	5.53	5.97	8.10	16.03

N=10, DIM=200					
BNCpack/MPIBNCpack					
	double	50 桁	100	200	500
1PE	3.28	30.68	39.70	85.98	284.74
10PEs	22.82	24.10	24.90	31.96	59.99

る。しかし、IEEE754 double の場合、この程度の次元数では並列計算の効果は全くなく、むしろ計算時間は増大している。逆に多倍長計算では低い次元数でもそれなりに効果があることが分かる。

但し、今回の問題は安定性に優れているもので、ステップ幅制御機構のあるソルバを使用したにもかかわらず、0.1 程度の初期ステップ幅を与えておけば、殆どの部分でこのステップ幅での計算が実行された。

## 5. 多倍長計算による安定性解析

数値例に示されるように、今回対象とした系では漸近安定な平衡点が存在していることが期待される。そこで、 $F(\mathbf{Y}^*) = 0$  となる平衡点  $\mathbf{Y}^*$  を求め、必要があれば平衡点における関数  $F$  の Jacobi 行列  $\partial F(\mathbf{Y}^*)/\partial \mathbf{Y} = J(\mathbf{Y}^*)$  の固有値も求め、漸近安定性を調べることにする。これらの性質を調べるには慎重な精度チェックが欠かせないため、全ての計算は多倍長浮動小数点数を用いて行う必要がある。

平衡点、及び Newton 法と固有値計算、全てにわたって (3) 式の右辺の関数  $F(\mathbf{Y})$  の Jacobi 行列が必要となるので、今回は中心差分商と補外法 (最大 10 段) に基づく数値微分 [5] を使用して求めることにする。一般には差分商に基づく数値微分は桁落ちによる精度低下があると思われるが、補外法によって大きい刻みから徐々に打ち切り誤差を落としていくことでかなりの精度を確保でき、丸め誤差の増大も抑えることが出来ることが知られている [6]。しかし、安定性解析においては高い精度で平衡点を求める必要があるため、Jacobi 行列の導出及び平衡点を求める Newton 法まで、すべて BNCpack/MPFR を用いた多倍長計算を行って精度の確保に努めた。また、Newton 法の反復に必要な連立一次方程式の計算は LU 分解を用いて行っている。

その結果、 $N = 2, 5$  の場合は Newton 法によってそれぞれ 4 回、7 回反復後に平衡点が求められた。

また、この平衡点における Jacobi 行列の固有値は実際に計算をする必要もなく、すべての固有値の実部が負になることも確認された。例として図 8 に Newton 法の初期値  $\mathbf{Y}_0$  と、4 回反復後に収束した平衡点の近似値  $\mathbf{Y}_4$  における Jacobi 行列成分をそれぞれ示す。これらの計算は 2 進 128bit で実行されたものであるが、256bit 計算の結果と差が殆どないことが確認されている。初期値においても殆ど対角優位な sparse 行列になっており、平衡点においてはその傾向が一段と強まることが分かる。 $N = 5$  の場合も同様であることが確認されている。また、平衡点  $\mathbf{Y}_4$  の値は、(5) 式の値と極めて近いことも確認できる。よって、今回対象とした系は漸近安定な平衡点を持つ、ということが結論付けられる。

## 6. 結論と今後の課題

以上によって次の結論を得た。

- 今回の解析対象としたモデルは、漸近安定な平衡点を必ず持つ問題となっていることが確認できた。
- 安定しているモデルであるため、多倍長 ODE ソルバを用いる必要はない。またもっと低次の解法でも問題なく解けるものと思われる。また今回使用した 0.1 刻みではステップ幅制御も殆ど必要がなかった。

今後の課題は、今回使用した安定性解析プログラムを使って、安定な ODE になるパラメータ・蛋白質の促進/抑制効果の組み合わせを見つけことが挙げられる。その際には Jacobi 行列の導出と、それを用いて計算を行う Newton 法・固有値解法の高速度化が欠かせないが、これは MPIBNCpack[7] を用いることで容易に実現できる。

## 参考文献

- [1] E.Hairer, S.P.Nørsett, G.Wanner, Solving Ordinary Differential Equations I (2nd ed.), Springer-Verlag, 1993.
- [2] G. von Dassow et al., "The segment polarity network is robust developmental module", Nature Vol.406, pp.188-192, 2000.
- [3] 大楯弘順・他, 単純化した相互作用ルールによる擬似細胞のパターン形成と再生, 静岡理工科大学紀要 Vol.11, 2003.
- [4] 幸谷智紀・大楯弘順, Segment Polarity Network Model に基づく細胞間再生モデル, 研究集会「常微分方程式とその周辺」, 2005.
- [5] 永坂秀子・福井義成, "数値微分の誤差", 情報処理学会論文誌, vol.22, No.5, pp.411-416.
- [6] 福井義成, "数値微分の補外法の収束判定について", 数値解析研究会報告 Vol.44, No.4, pp.21-28.
- [7] BNCpack/MPIBNCpack, <http://na-inet.jp/na/bnc/>
- [8] GMP, <http://swox.com/gmp/>
- [9] MPFR, <http://www.mpfr.org/>

$$J(\mathbf{Y}_0) = \begin{bmatrix} -1.0e+1 & 0 & 0 & 0 & 0 & 0 & 2.4e-4 & 0 \\ 0 & -1.0e+1 & 0 & 0 & 0 & 0 & 0 & 6.9e-3 \\ 0 & 0 & -1.0e-1 & -1.4e-3 & 4.2e-5 & 0 & -3.7e-11 & 0 \\ 0 & 0 & -3.8e-4 & -1.0e-1 & 0 & 5.7e-3 & 0 & -6.8e-14 \\ 0 & 0 & 6.9e-3 & 0 & -1.0e+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2.4e-4 & 0 & -1.0e+1 & 0 & 0 \\ 5.7e-3 & 0 & -6.8e-14 & 0 & 0 & 0 & -1.0e-1 & -3.8e-4 \\ 0 & 4.2e-5 & 0 & -3.7e-11 & 0 & 0 & -1.4e-3 & -1.0e-1 \end{bmatrix}$$

$$J(\mathbf{Y}_4) = \begin{bmatrix} -1.0e+1 & 0 & 0 & 0 & 0 & 0 & -4.1e-47 & 0 \\ 0 & -1.0e+1 & 0 & 0 & 0 & 0 & 0 & 3.0e-2 \\ 0 & 0 & -1.0e-1 & 5.2e-38 & 1.5e-12 & 0 & 0 & 0 \\ 0 & 0 & -6.4e-39 & -9.9e-2 & 0 & -2.9e-49 & 0 & -1.7e-48 \\ 0 & 0 & 3.0e-2 & 0 & -1.0e+1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -4.1e-47 & 0 & -1.0e+1 & 0 & 0 \\ -2.9e-49 & 0 & -1.7e-48 & 0 & 0 & 0 & -9.9e-2 & -6.4e-39 \\ 0 & 1.5e-12 & 0 & 0 & 0 & 0 & 5.2e-38 & -1.0e-1 \end{bmatrix}$$

$$\text{ここで } \mathbf{Y}_4 = \begin{bmatrix} \mathbf{x}_1^{(4)} \\ \mathbf{y}_1^{(4)} \\ \mathbf{x}_2^{(4)} \\ \mathbf{y}_2^{(4)} \end{bmatrix} = \begin{bmatrix} -4.85294803825705009448219411407783514907e-48 \\ 8.88888888859272571565517338593902275969e-2 \\ 9.99999999900044929050272711437470141813 \\ 2.11720493992969174634832967759936526331e-35 \\ 8.88888888859272571565517338593902275969e-2 \\ -4.85294803825705009448219411407783514907e-48 \\ 2.11720493992969174634832967759936526331e-35 \\ 9.99999999900044929050272711437470141813 \end{bmatrix}$$

図 8:  $N = 2$  の Jacobi 行列  $J(\mathbf{Y}_0)$ ,  $J(\mathbf{Y}_4)$  と平衡点  $\mathbf{Y}_4$