

# 応用レベル(量子人工生命):テキスト

## 1 人工生命とは

- 人工生命 (Artificial Life、ALife: エーライフ) は、生命をシミュレーションすることで、生命に関するシステム (生命プロセスと進化) を研究する分野である。シミュレーションはコンピューター上で行われる。
- 生物学では、生物の詳細な「観察」というアプローチがとられる。観察によって得られた情報から、生物の基礎となる事象を明らかにしていくことになる。近年では、「システム生物学」という新たな分野も登場している。この分野は、生命現象をシステムとして理解することを目的としているが、観察をベースとしている点は、生物学と同じアプローチである。
- 人工生命は生物の観察ではなく、「生命の本質」に関する「仮説」を立てることから始まる。仮説をもとに人工的なシステムを「作る」ことで生命を理解しようとするアプローチであり、生物学との違いでもある。

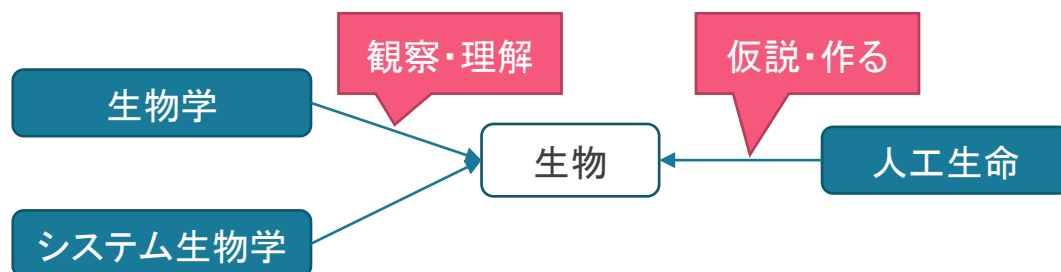


図. 生物に対するアプローチの違い

## 2 人工生命の歴史

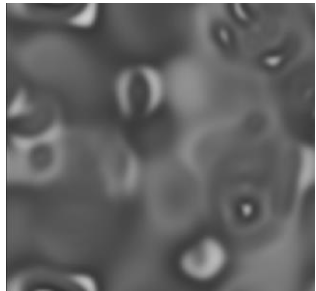
- 1952年、数学者の[アラン・チューリング](#)は、生物の持つ模様を数式化した論文「The Chemical Basis of Morphogenesis」(形態形成の化学的基礎)を発表。数学によって生命を記述する試みがなされた。また、より複雑な問題に対する、将来的なデジタルコンピューターの利用について触れられた。
- 1966年、[ジョン・フォン・ノイマン](#)は、研究をまとめた「The Theory of Self-reproducing Automate」を出版した。「自己増殖」という生命にとって最も根源的な特徴が数学的に記述された。
- 1986年、「Artificial Life」(人工生命)は、アメリカの理論的生物学者、[クリストファー・ラングトン](#)によって提唱された。1987年、国際会議「International Conference on the Synthesis and Simulation of Living Systems」(通称、Artificial Life I) がロシアモスクワ国立研究所で開催された。科学的なコミュニティとしてのALifeはここから始まったとされている。

## 3 参考資料

- 作って動かすAlife (岡 瑞起、オライリー・ジャパン)
- Wikipedia: [人工生命](#)、[生物学](#)、[アラン・チューリング](#)、[ジョン・フォン・ノイマン](#)、[クリストファー・ラングトン](#)
- 筑波大学オープンコースウェア: Youtube (人工生命概論 第1回)

## 1 人工生命のモデルと自然界の模様

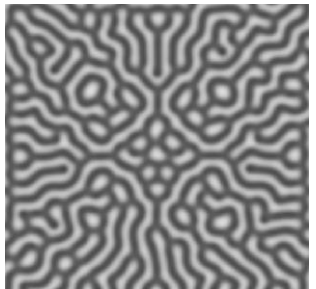
- 人工生命のモデルには様々あるが、そのうちいくつかのモデルで描き出されるパターンは、自然界に存在する模様とよく似ている。
- **Gray-Scottモデル**では、2つのパラメータ値を変えることで異なるパターンを描くことができる。そのうちのいくつかは、ある生物の表皮によく似ている。以下にその例を示す。



Gray-Scottモデル  
Wavesパターン



シマウミスズメ  
[https://commons.wikimedia.org/wiki/File:Thornback\\_cowfish\\_\(Lactoria\\_fornasini\)\\_\(40757394194\).jpg](https://commons.wikimedia.org/wiki/File:Thornback_cowfish_(Lactoria_fornasini)_(40757394194).jpg)

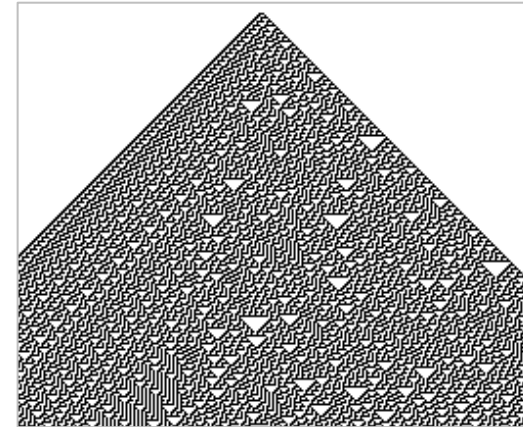


Gray-Scottモデル  
Amorphousパターン



淡水フグの一種  
[https://commons.wikimedia.org/wiki/File:Giant\\_Puffer\\_fish\\_skin\\_pattern.JPG](https://commons.wikimedia.org/wiki/File:Giant_Puffer_fish_skin_pattern.JPG)

- 白と黒の一次元配列を時間的に変化させる「**一次元セル・オートマトン**」には、変化のルールが**256種類**ある。そのうちのひとつである「**ルール30**」で描かれる模様は、イモガイの模様によく似ている。



一次元セル・オートマトン ルール30



イモガイ  
[https://commons.wikimedia.org/wiki/File:Textile\\_cone.JPG](https://commons.wikimedia.org/wiki/File:Textile_cone.JPG)

- 人工生命の仮説(ルール)の中には、偶然に描き出される模様や挙動が、**自然界に存在する生物に酷似**するものがある。生物の複雑な模様や振る舞いが、シンプルなルールによって再現される点が、人工生命の魅力のひとつでもある。

## 2 参考資料

- 作って動かすAlife(岡 瑞起、オライリー・ジャパン)
- 画像出典: Wikimedia(URLは個別に記載)

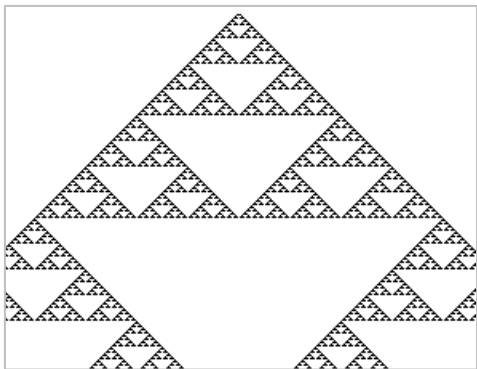
## 1 人工生命の手法

- 人工生命には、現在までの研究によって蓄積された様々なモデル、ルール、シミュレータが存在する。ここではその一部を紹介する。

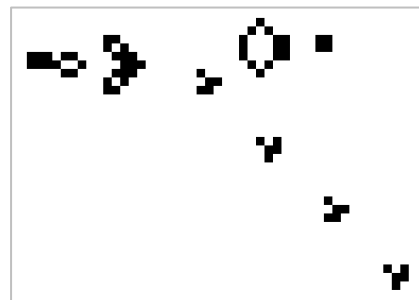
## 2 さまざまなモデル



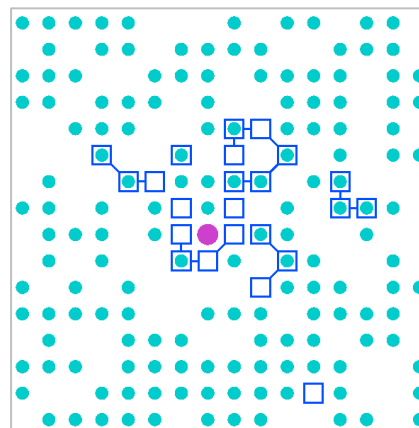
- **Gray-Scottモデル**は、反応拡散系に分類されるモデルのうちのひとつである。
- 左図はこのモデルによって描かれたパターンである。2つの物質が互いに**反応**(化学反応)して空間に広がる(**拡散**)することで、不規則なパターンを描く。色が濃い部分に、より多くの物質が含まれている。



- **セルラー・オートマトン**は、格子状に並んだセルで表現されるモデルである。
- 左図は「**一次元セルラー・オートマトン**」によって描かれたパターンである。本来は単一行で表現されるモデルだが、左図では時間経過(タイムステップ)ごとに上から下に描画結果を並べている。



- 左図は、「**二次元セルラー・オートマトン**」によるものである。**ライフゲーム**とも言われる。セルがタテ・ヨコの二次元に展開されている。
- 各セルに存在する個体が周囲の個体と**相互作用**し、**誕生・維持・死亡**を繰り返しながら複雑なパターンを描く。



- **SCLモデル**は、個体(図のドット)の周囲に膜分子(図の青枠)が生成されるモデルである。
- このモデルでは、生物が自ら行動を決定しているように見えるため、生命の持つ**自律性**に関する議論が展開された。

- ここで紹介したモデルは、ほんの一部に過ぎない。ほかにも多くのモデルやシミュレータが存在しており、現在でも開発が続いている。

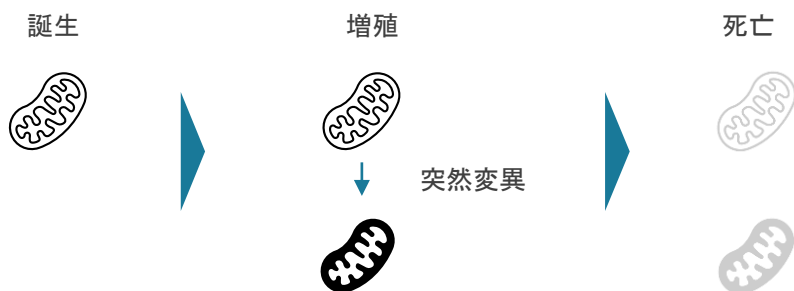
## 3 参考資料

- 作って動かすAlife(岡 瑞起、オライリー・ジャパン)



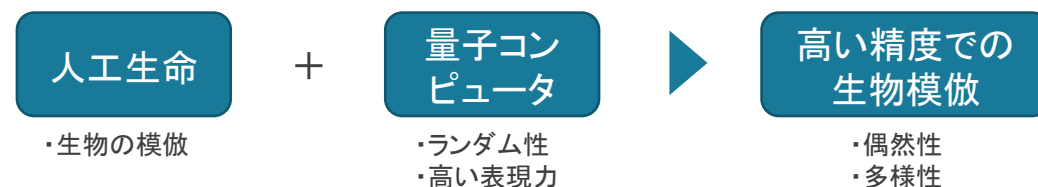
## 1 生命と偶然性

- 生物を模倣することは、生命を理解するための第一歩であり、人工生命を研究する目的でもある。
- 生物は誕生、増殖(生殖)、死亡を繰り返す中で、突然変異を起こすことがある。突然変異は、生物の進化において非常に重要な役割を果たす。
- 突然変異(ここでは、遺伝子突然変異のこと)は、遺伝子のミスコピーや、外部環境(放射線、化学物質)によって受けた損傷の修復時に起こると考えられている。
- ダーウィン進化論では、「生物の進化には目的があるのではなく、単なる『結果』にすぎない」としている。つまり、「たまたま環境に合った」という偶然でしかない、ということである。
- 突然変異は偶然によるものであり、進化もまた偶然である。生命は多くの偶然が折り重なって存在していると考えることができる。



## 2 量子コンピュータの適用

- 量子コンピュータは様々な分野での応用が期待されているが、中でも量子コンピュータの持つランダム性と高い表現能力は注目すべき点である。
- 人工生命(生物の模倣)に量子コンピュータを適用する意義として、自然界で繰り広げられている生命の偶然性と多様性を、量子コンピュータが持つランダム性と表現能力によって再現することで、複雑な系をより高い精度で模倣することができると考えられる。
- 当教材では実際に量子人工生命を実装しながら、量子コンピュータ適用の可能性を探っていく。



## 3 参考資料

- [「Quantum Artificial Life in an IBM Quantum Computer」\(U. Alvarez-Rodriguez, 2017\)](#)
- 高校の生物が根本からわかる本(藤井 亘、中経出版)
- 作って動かすAlife(岡 瑞起、オライリー・ジャパン)

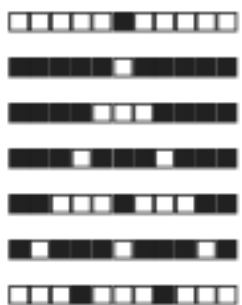
## 1 セルラー・オートマトンとは

- **セルラー・オートマトン** (cellular automaton) は、格子状のセルと単純な規則による、シンプルなモデルである。1940年代にジョン・フォン・ノイマンとスタニスワフ・ウラムによって考案された。シンプルでありながら豊かな表現力を有しており、複雑な自然現象を模した結果を得ることができる。
- セルラー・オートマトンにはいくつかの種類があり、すべてに共通する**4つのルール**がある。

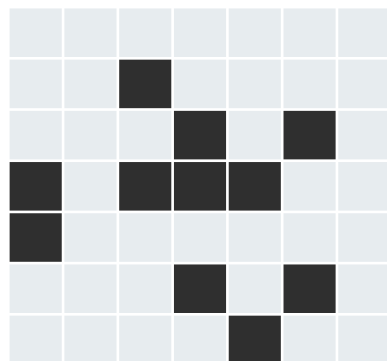
## 2 4つの基本ルール

### ① 空間があること

セルを格子状に敷き詰める空間があることが一つ目のルールである。空間には一次元(一行の格子)、二次元(縦横の格子)、三次元(立体)がある。



一次元セルラー・オートマトン



二次元セルラー・オートマトン

### ② 時間があること

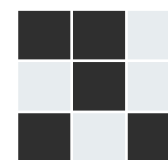
セルの状態は時間によって変化する。つまり、**時間単位(ステップ)**があることが二つ目のルールとなる。

### ③ セルの状態があること

セル(格子)ごとに状態があることが三つ目のルールとなる。**状態はいくつでも指定可能**だが、生と死のふたつの状態を持たせるケースが多い。

### ④ セルの状態を変化させる条件があること

セルの状態は、さまざまな条件によって変化する。この条件が明確にあることが最後のルールとなる。基本的に、対象セルの次の状態は、周囲のセルの状態によって決まる。一次元の場合は左右にあるふたつのセル、二次元の場合は周囲にある8つのセルによって次の状態が決まることになる。



二次元の場合、生きている状態にある中央のセルが、次のステップで生きたままとなるか、あるいは死んだ状態になるかは、「周囲8つのセルのうち、いくつが生きた状態であるか」という条件によって決まる。

## 3 参考資料

- Wikipedia : [セル・オートマトン](#)、[ライフゲーム](#)
- 作って動かすAlife(岡 瑞起、オライリー・ジャパン)

## 1 二次元セルラー・オートマトンとは

- 二次元セルラー・オートマトンは、セルラー・オートマトンの一種で「ライフゲーム」とも呼ばれる。ジョン・ホートン・コンウェイが考案した。
- 生と死のふたつの状態を持つセル(細胞)が二次元の格子状に並び、時間単位(ステップ)の経過によって生死を繰り返すことで、独特の模様を描き出す。

## 2 二次元セルラー・オートマトンのルール

- ルールは非常にシンプルである。
- あるセルの生死は周囲8つの近傍セル(ムーア近傍)によって決まる。具体的には、以下の4つのルールからなる。

## ① 誕生 (死⇒生)

死んでいるセルの近傍に生きたセルが 3つ ある場合、誕生する。



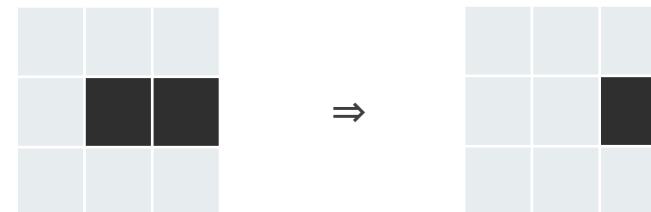
## ② 生存 (生⇒生)

生きているセルの近傍に生きたセルが 2つ または 3つ ある場合、生存したままになる。



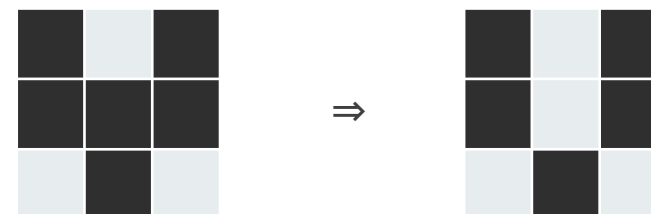
## ③ 過疎 (生⇒死)

生きているセルの近傍の生きたセルが 1つ以下 の場合、死滅する。



## ④ 過密 (生⇒死)

生きているセルの近傍に生きたセルが 4つ以上 ある場合、死滅する。

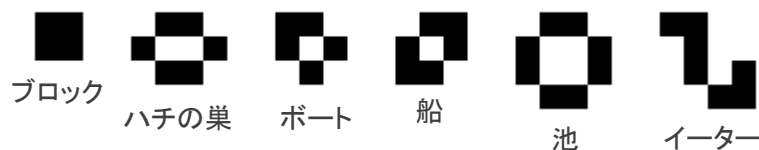


## 3 さまざまなパターン

- ライフゲームで描き出される模様は初期条件によって決まる。
- 時間経過によって死滅するパターンと生き延びるパターンがある。
- 生き延びるパターンは4つに分類される。

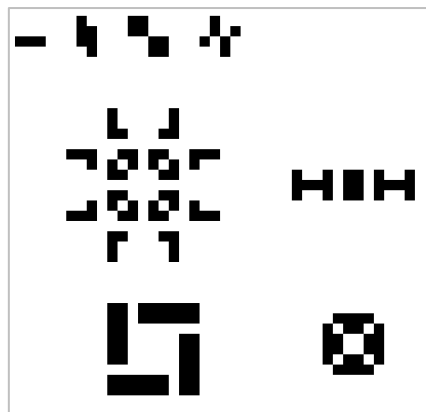
## ① 固定物体(Still lifes)

時間が経過しても変化しない物体。



## ② 振動子(Oscillators)

その場で周期的なパターンを繰り返す物体。

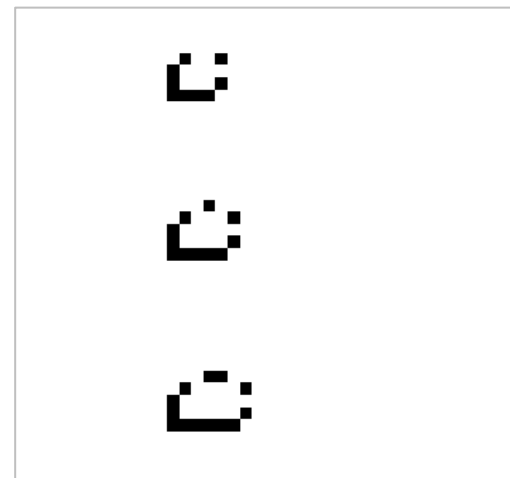


上段: ブリンカー、ヒキガエル、ビーコン、時計  
中段: パルサー、ペンタデカスロン  
下段: 銀河、八角形

※動作している様子は以下のファイルを参照。  
「2-2-3\_二次元セルラー・オートマトン\_demo1.gif」

## ③ 移動物体(Spaceships)

一定のパターンを繰り返しながら移動する物体。



上から、軽量級宇宙船、中量級宇宙船、  
重量級宇宙船

※動作している様子は以下のファイルを参照。  
「2-2-3\_二次元セルラー・オートマトン\_demo2.gif」

## ④ 繁殖型(Infinite growth)

無限に増え続けるパターン。(実際は描画領域に制限される)



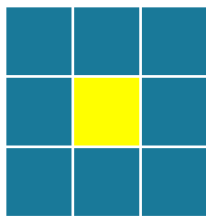
グライダーガン

※動作している様子は以下のファイルを参照。  
「2-2-3\_二次元セルラー・オートマトン\_demo3.gif」



## 4 ルールのバリエーション

- 通常、二次元セルラー・オートマトンでは、コンウェイが考案した4つの基本ルールを適用するが、それ以外のルールを適用することもできる。
- 基本ルールでは周囲に3つの生命があれば誕生し、2つか3つの場合は生き残り、それ以外は死滅した。このような標準のライフゲームを「23/3」と表す。ルール変更のバリエーションとしては、「23/36」(HighLife)が有名である。
- 基本ルールでは、周囲8つのセル(ムーア近傍)を条件としたが、上下左右の4つのセル(ノイマン近傍)や、その他の条件を適用することもできる。
- 生命の数や近傍のバリエーションを変更することで、より複雑な条件でライフゲームを展開することができる。



ムーア近傍



ノイマン近傍

## 5 ライフゲームの可能性

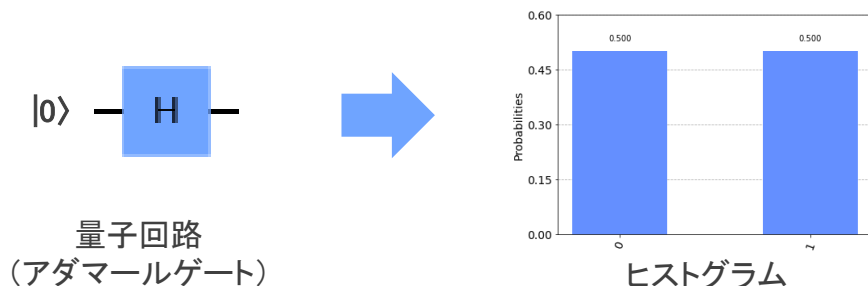
- コンウェイは、いくつかの基本的なルールが、考えられないくらい複雑な結果を生み出せることを示そうとした。そのため、ライフゲームのルールは、セルの誕生と生き残りが「稀」でも「頻繁」でもなく「適度」に起こるようにすべきと考えた。その結果として生み出されたライフゲームは、局所的な動きは完全に明白だが、全体としての動きを完璧に把握することはできないものとなった。
- ライフゲームの基本ルールは、前項で説明したように変更したり、あるいは新たなルールを適用したりすることで、さらに予測の難しい状況を作り出すことができる。基本ルールのシンプルさがライフゲームの魅力のひとつではあるが、ルールを変更することで生命をより高い精度でシミュレーションできる可能性があるともいえる。
- 長い間、多くの人々を魅了してきたという事実は、ライフゲームが我々に新たな示唆を与えてくれていることの証左ともいえる。今回、量子コンピュータを適用したライフゲームに取り組む。新たな「ルール」を適用することで、ライフゲームの可能性を見出していく。

## 6 参考資料

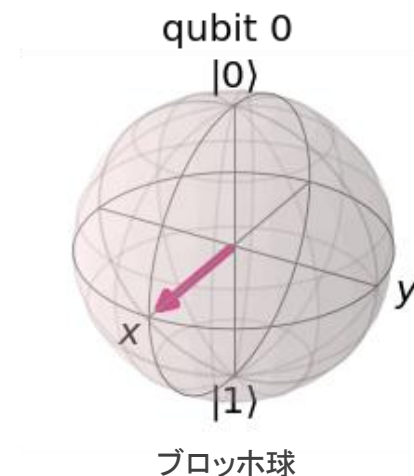
- Wikipedia: [セル・オートマトン](#)、[ライフゲーム](#)、[Conway's Game of Life](#)
- 作って動かすAlife(岡 瑞起、オライリー・ジャパン)
- 数学でみた生命と進化(カール・シグムンド、ブルーバックス)

## 1 量子コンピュータを適用する意義

- ここでは、二次元セルラー・オートマトンに量子コンピュータを適用した、当教材オリジナルの「量子二次元セルラー・オートマトン」 (量子ライフゲーム)を取り上げ、基本ルールの検討から実装、評価まで取り組む。まずは、量子ライフゲームに量子コンピュータを適用する意義を確認する。
- 従来のライフゲームは、コンウェイが考案した基本ルールに従い、厳密に進行する。一方、自然界はダーウィン進化論で説明されるように、偶然(確率過程)で成り立っている。そして、ライフゲームに量子コンピュータを適用するうえで注目すべき特徴として、**確率的な振る舞いと、連続値による表現力の高さ**があげられる。
- **確率的な振る舞い**については(テキスト1-4で「偶然性」としても触れている)、量子の重ね合わせ状態として確認することができる。たとえば、 $|0\rangle$ で初期化された量子ビットにHゲート(アダマールゲート)を作用させると、観測時には0と1の状態が50%ずつ観測されることになる。つまり、より複雑な混合状態を取ることで、さまざまな確率過程を表現することができるようになる。



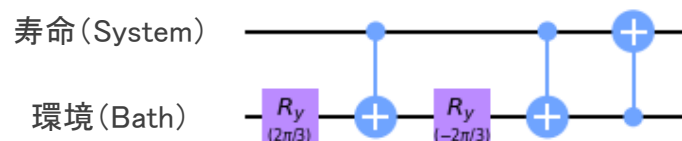
- **連続値**については、ブロッホ球で示されるような、**量子ビットの状態(量子状態)**として確認することができる。量子状態は連続値をとるため、球面上のどの点でも取ることができる。



- つまり、量子コンピュータが持つこれらの特徴を利用することで、**生命をより高い精度で模倣できる可能性がある**と考えることができる。これが量子コンピュータを適用する意義である。当テキストでは二次元セルラー・オートマトンへの量子コンピュータ適用を行う。

## 2 ルールの基本アイデア

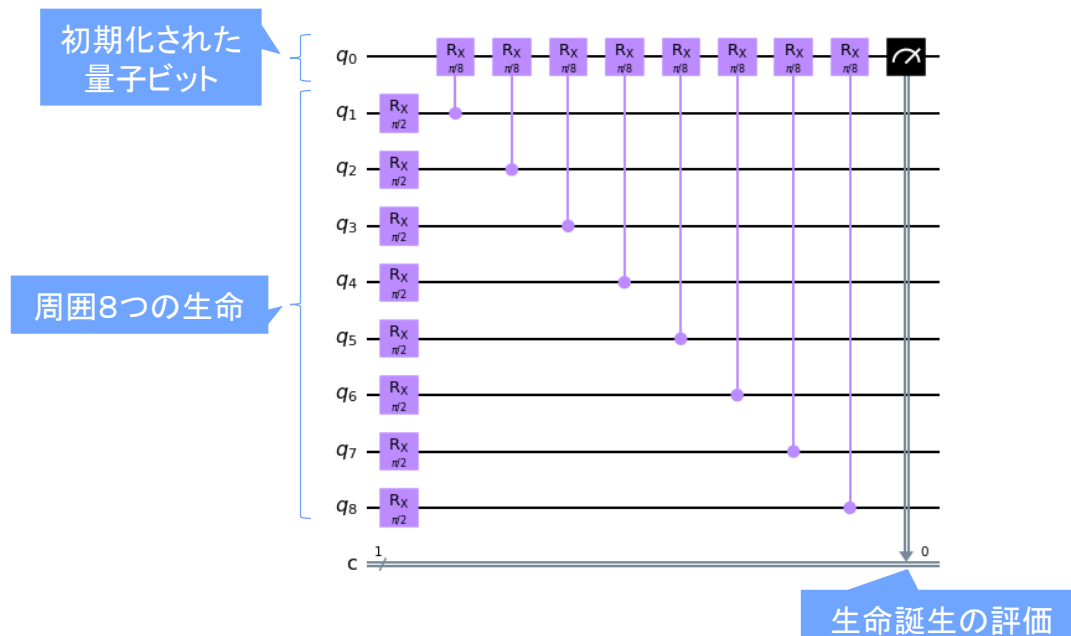
- 二次元セルラー・オートマトンに量子コンピュータを適用するにあたっては、新たなルールを考案し、適用する必要がある。まずは、《死》と《誕生》それぞれの基本アイデアを検討する。
- 《死》  
従来の基本ルールでは「生」と「死」のふたつの状態をとるが、生命を表現するにはいささか乏しい。そこで、連続値で表現された「寿命」を追加し、より豊かな生命の表現に変えていく。  
参考論文(参考資料の項を参照)では「Lindblad方程式」を適用することで、量子状態の崩壊(=寿命の変化)を表現している。これを参考に、今回は下図のような量子回路で寿命の減衰を模倣することとした。  
具体的には、寿命を表現する量子ビット(寿命量子ビット)に対し、環境を模した量子ビット(環境量子ビット)を作用させることでデコヒーレンスを生じさせ、寿命量子ビットの値(期待値)を徐々に減衰させる方法である。



## 【参考】Lindblad方程式

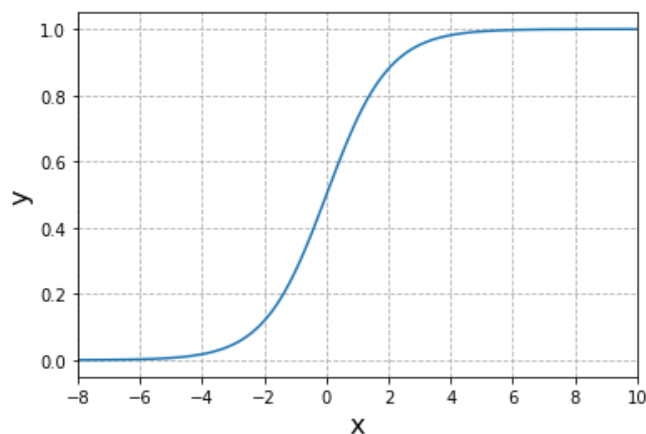
デコヒーレンスのある系で密度演算子の時間発展を記述する方程式。考えている系をS(System)、環境をB(Bath)と書く。環境Bの影響は、時間発展(時間の経過)によりSの測定値に現れる。

- 《誕生》  
従来の基本ルールでは周囲8つのセルのうち、3つのセルに生命が存在する場合に「誕生」することになっている。  
今回は周囲のセルに存在する生命の寿命をもとに、新たな生命が誕生するかを判定する方式とした。具体的には、初期化された量子ビットに対し、周囲8つの生命の寿命量子ビットを制御ビットとしたRxゲートを順次適用し結果を測定することで、観測値から生命誕生の可能性を評価する。



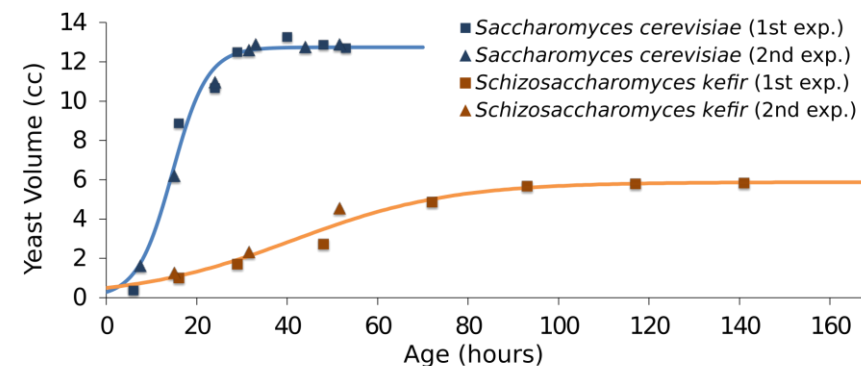
## 3 評価方法の検討

- 基本ルールは自由に設定できるが、どのようなものでも良いわけではない。ルールの妥当は何かしらの基準をもって評価する必要がある。例えばコンウェイは二次元セルラー・オートマトンのルールを考案する際、「セル(生命)の誕生と生き残りは、稀にでも頻繁にでもなく『適度』に起こる必要がある」と考え、これを重要な評価軸とした。
- 今回は「自然界での生命活動を模倣すること」を目的とした評価軸を設定する。具体的には、ロジスティック曲線(下図)への近似を評価指標とする。  
自然界における生命の増え方はロジスティック曲線に近似できることが知られている。生命の数は徐々に増加するが、ある値を上限に収束していく、というものである。



ロジスティック曲線

- 前頁の基本アイデアを適用したシミュレーションの結果情報として、生命(セル)の数の推移を得ることができる。この推移がロジスティック曲線に近似できれば、自然界に近いシミュレーションができています(自然界を模倣できている)と考えることができる。
- 以上の評価指標をもって、量子コンピュータを適用した新たなルールの検証を行っていく。

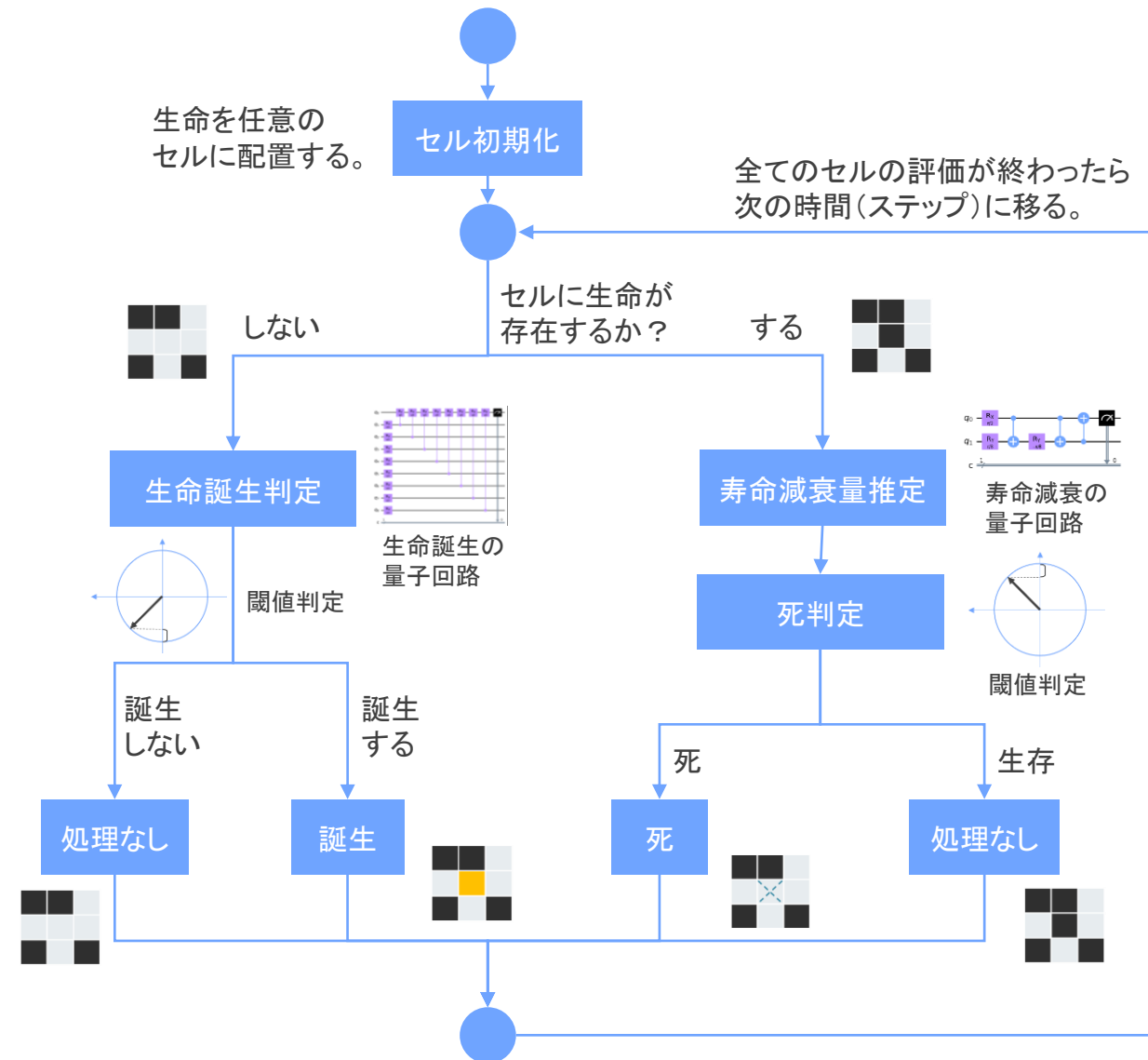


2種の酵母の個体群サイズ成長実験データ  
※推移がロジスティック曲線になっていることがわかる。

[https://ja.wikipedia.org/wiki/%E3%83%AD%E3%82%B8%E3%82%B9%E3%83%86%E3%82%A3%E3%83%83%E3%82%AF%E6%96%B9%E7%A8%B%E5%BC%8F#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%A5:B:Gause's\\_experiment\\_and\\_fitted\\_logistic\\_curves\\_\(single\\_species\).svg](https://ja.wikipedia.org/wiki/%E3%83%AD%E3%82%B8%E3%82%B9%E3%83%86%E3%82%A3%E3%83%83%E3%82%AF%E6%96%B9%E7%A8%B%E5%BC%8F#/media/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%A5:B:Gause's_experiment_and_fitted_logistic_curves_(single_species).svg)

## 4 処理フロー

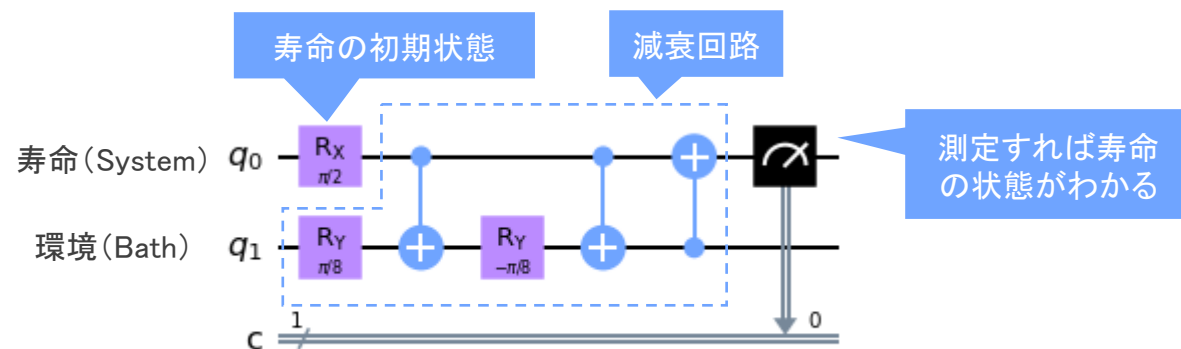
- 新ルールの基本アイデアを踏まえ、実装に向けた準備として処理フローを右図のように整理した。
- このフローは、9つのセル(中央+周囲8つ)のうち、中央のセルに注目したものとなっている。二次元平面を構成するすべてのセルに対してフローの処理を行ってから、次の時間(ステップ)に移る。なお、二次元平面は左端と右端、上端と下端がそれぞれ接続された連続的な平面として扱う。
- 処理の流れは、概ね以下のとおりである。
  1. 生命を任意のセルに配置する。(セル初期化)
  2. 中央セルに生命が存在するかをチェックする。
    - ・ 生命が存在する場合は、量子回路により寿命減衰量を推定し、死に至ったか(寿命が閾値を下回ったか)を判定する。
    - ・ 生命が存在しない場合は、量子回路により新たな生命の寿命を推定し、誕生するか(寿命が閾値を上回ったか)を判定する。
  3. すべてのセルの処理が完了したら、セルの状態を一斉に更新し、次の時間(ステップ)に移行する。
- 次項からは、「寿命減衰の量子回路」と「生命誕生の量子回路」の概要を順に説明する。



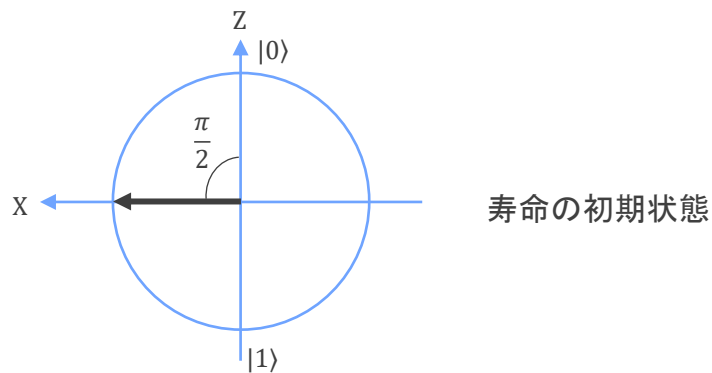


## 5 寿命減衰の量子回路

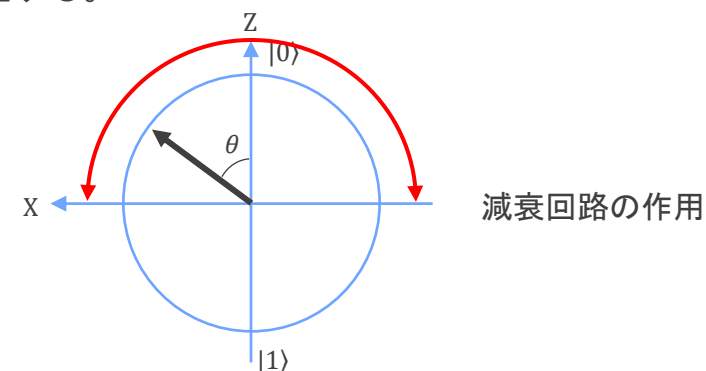
- 寿命の減衰は「2. ルールの基本アイデア」で説明した下図の回路を使用する。



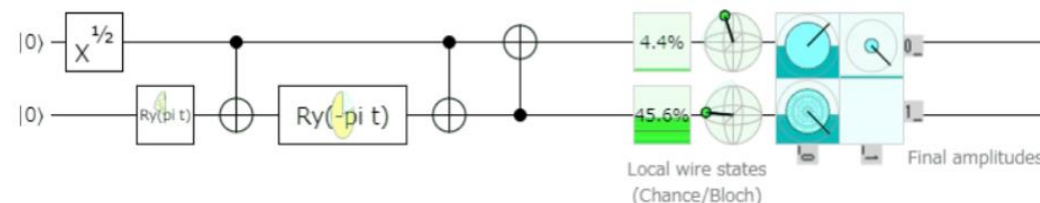
- 上記の例では、最初に寿命の量子ビットを、 $R_x\left(\frac{\pi}{2}\right)$  で初期化を行っている。  
寿命の値はZ軸上の値で表現されており、 $|1\rangle$ が最大、 $|0\rangle$ が最小となる。



- この量子状態に減衰回路を作用させると、初期状態 $\sim|0\rangle$ の範囲(赤矢印の範囲)で量子状態が変化する。つまり、変化は必ず減衰方向に生じるため、減衰回路を作用させる度、寿命が徐々に縮まることになる。なお、変化の量は  $R_y$  のパラメータ次第であり、値はランダムに決定する。

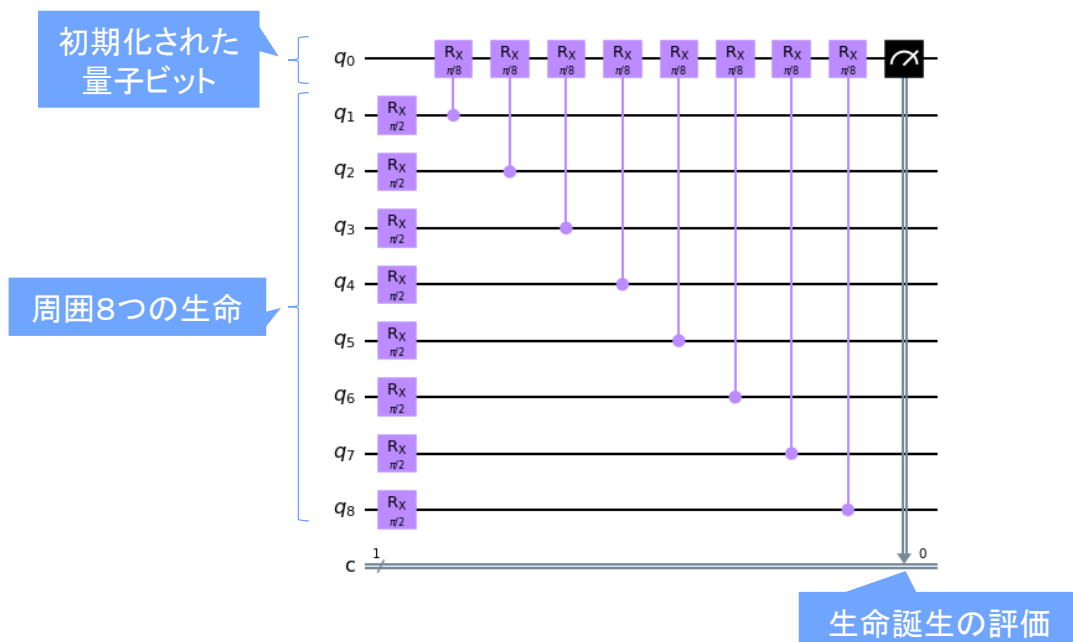


- 環境との相互作用を繰り返し(量子状態を変化させ)、減衰後の値(期待値)が特定の閾値(パラメータで設定)を下回る場合に「死」と判定する。
- この回路の動作中の様子は、「3-2\_量子二次元セルラー・オートマトン\_01\_death.gif」参照。(Quirkを利用し描画)



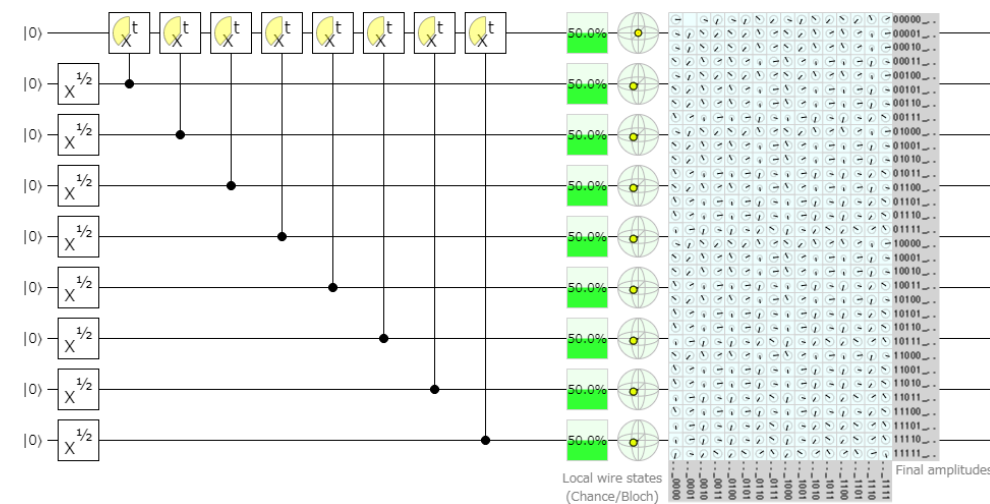
## 6 生命誕生の量子回路

- 生命の誕生は、初期化された量子ビット(0番目の量子ビット: $q_0$ )に対し、周囲8つの量子ビットを作用させたうえで、生命誕生の判定を行う。



- なお、 $q_0$ に対する作用の強さである $R_x$ の値は、パラメータとしてあらかじめ設定する。

- 生命誕生の判定は、 $q_0$ の最終的な期待値が、特定の閾値(パラメータであらかじめ設定)を上回る場合に「誕生」と判断する。
- この回路の動作中の様子は、「3-2\_量子二次元セルラー・オートマトン\_02\_birth.gif」参照。(Quirkを利用し描画)



## 7 量子ライフゲームの実装

- ここまでの基本アイデアをQiskitで実装した「量子ライフゲーム」の関数について説明する。  
ソースコードは「3-2\_量子二次元セルラー・オートマトン\_py.zip」に格納している。ファイル構成は以下のとおり。

```
(Root)
├─ quantum_game_of_life.py          ...①
├─ quantum_game_of_life_usage.py    ...②
├─ game_of_life_patterns.py         ...③
├─ game_of_life_patterns_add.py     ...④
└─ visualizers                     ...⑤
   └─ ...
```

- ①: 量子ライフゲームの実装本体。  
メイン関数のパラメータは右表のとおり。
- ②: 量子ライフゲームの「使い方」を実装した実行ファイル。  
①の実行に加え、実行結果のデータファイルと、データを  
プロットしたグラフ画像ファイルを保存する機能を追加している。
- ③: 二次元セルラー・オートマトンを可視化するためのファイル群。
- ④⑤: 初期状態を定義したファイル。(配列の定数)
- なお、③⑤は以下のソースコードを使用している。  
[https://github.com/alifelab/alife\\_book\\_src](https://github.com/alifelab/alife_book_src) (MITライセンス)

関数: quantum\_2d\_cellular\_automaton のパラメータ

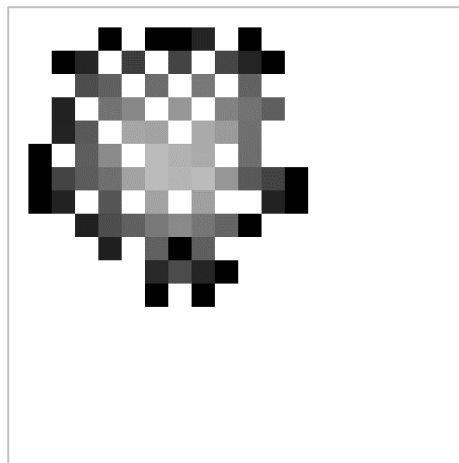
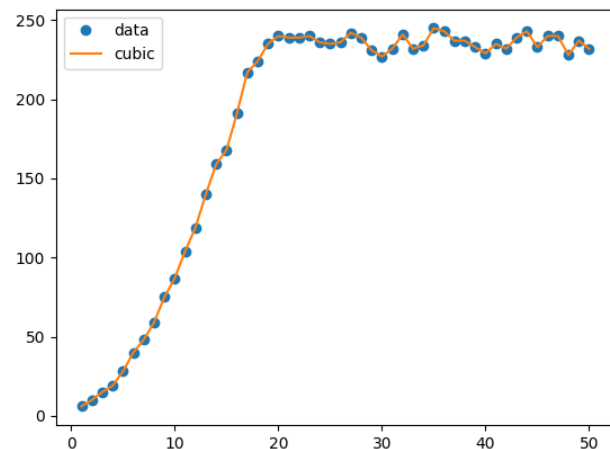
描画パラメータ	
width : int	描画領域の横グリッド数。 デフォルト : 20
height : int	描画領域の縦グリッド数。 デフォルト : 20
pattern : list of int	開始時の初期パターン。二次元配列で指定。 デフォルト : $[[1, 1], [1, 1]]$
寿命パラメータ	
env_interaction_theta : float	環境との相互作用の強さ。0~ $\pi/2$ の範囲で設定。(0で影響なし、 $\pi/2$ で最大) デフォルト : $\pi/8$
death_value : float	生き残り条件の閾値。寿命 (0~1) がこの値を下回ると死となる。例 : 0.1 デフォルト : 0.2
誕生パラメータ	
non_live : list of int	誕生しない条件。周囲で生存している生命の数をリストで指定する。 デフォルト : $[0, 1]$
live_theta : float	周囲で生存している生命による作用の強さ(rad)。大きいほど新たな生命が誕生しやすい。 デフォルト : $\pi/4$
live_low : float	誕生判定の下限閾値 (値が大きいと生まれにくい、疎状態の評価)。 デフォルト : 0.1
live_high : float	誕生判定の上限閾値 (値が小さいと生まれにくい、密状態の評価)。 デフォルト : 0.5
ファイルパラメータ	
data_file : bool	データファイル出力する場合はTrue。 デフォルト : False
data_file_name : str	データファイルのフルパス (ファイル名含む)。 デフォルト : ブランク (data_file=True の場合に指定必須)
data_num : int	記録するデータの数 (繰り返し数)。 デフォルト : 100
量子回路パラメータ	
shots : int	量子回路の実行回数 (ショット数)。 デフォルト : 1000

## 8 評価: ロジスティック曲線への近似

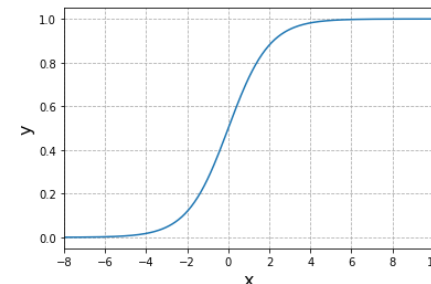
- ここでは、量子ライフゲームを実際に動作させ、生命(セル)の推移がどのようになるかを確認し、評価を行う。
- 今回の検証では、以下のパラメータを適用し生命の推移データを取得している。データはグラフにプロットしている。

```
width=20, height=20, pattern=CENTER_20,  
env_interaction_theta=math.pi / 8, death_value=0.2,  
non_live=[0, 1], live_theta=math.pi / 4, live_low=0.1, live_high=0.5,  
data_file=True, data_file_name=data_file, data_num=50, shots=1000)
```

パラメータ

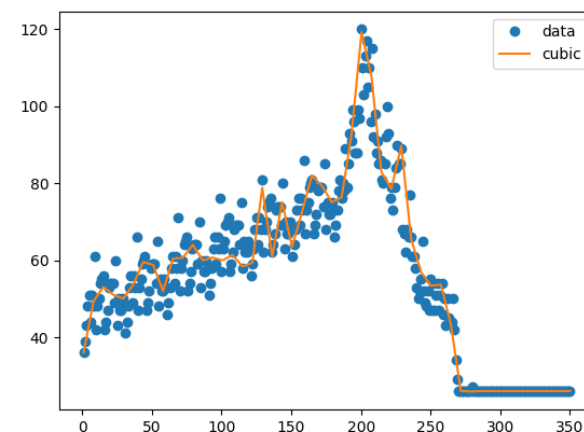
量子ライフゲーム: 動作中の様子  
(03\_Quantum.gif)量子ライフゲーム: 動作結果  
横軸: 経過時間(タイムステップ)、縦軸: 生命の数

- 結果として、生命の推移はロジスティック曲線に近似できるものとなった。



ロジスティック曲線

- 比較のため、従来の基本ルール(GliderGunパターン)で生命(セル)の推移データを取得したところ、下図のようになった。ロジスティック曲線とは異なる推移を示しているため、今回実装した量子ライフゲームの妥当性を確認することができる。



従来のライフゲーム: 動作結果 (GliderGun)

## 9 まとめ

- 当教材では、量子ライフゲームによる自然界(生命の振る舞い)の模倣に取り組み、妥当な結果(ロジスティック曲線への近似)を得ることができた。
- 一方、取組みを進める中で、実行時間に関する課題を認識した。具体的には、以下のとおりである。
  - 50タイムステップの実行処理にかかる時間  
コンウェイの基本ルール : 約0.85sec  
量子適用の新ルール : 約400sec →470倍の実行時間
- 今回実装した新ルールの検証では以下の理由によりシミュレータを用いている。
  - ・ 新ルールでは量子回路を多用していること。
  - ・ 実機へのアクセスには待ち時間を要すること。
  - ・ 無料アカウントでは5量子ビットまでしか使えないこと。

実機による検証を行うことができれば、実行時間を正しく評価することができるが、シミュレータ実行での処理効率化もひとつの課題であるといえる。

## 10 量子ライフゲームの可能性

- 今回考案した新ルールには、パラメータ調整や量子回路の変更などによる改善余地が多く残されている。
- パラメータ値は調整しやすいように関数として実装しているため、いろいろなパターンを試していただきたい。また、量子回路を変更することで、より複雑な表現を実装することもできるだろう。量子回路の学習が進んだら、量子回路を変更し、どのような結果が得られるかも試してみてほしい。
- 当教材を通じ、量子回路および量子プログラミングをより深く理解できるだろう。より精度の高い「量子ライフゲームによる生命の模倣」を目指し取り組んでみてほしい。

## 11 参考資料

- 参考論文:[「Quantum Artificial Life in an IBM Quantum Computer」](#)(U. Alvarez-Rodriguez, 2017)
- Wikipedia:[セル・オートマトン](#)、[ライフゲーム](#)、[Conway's Game of Life](#)
- 作って動かすAlife(岡 瑞起、オライリー・ジャパン)
- [物理とか\(ブログ\):Lindblad方程式の導出](#)
- 数学でみた生命と進化(カール・シグムンド、ブルーバックス)



本資料の著作権は、日本アイ・ビー・エム株式会社（IBM Corporationを含み、以下、IBMといいます。）に帰属します。

ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法律的またはその他の指導や助言を意図したものではなく、またそのような結果を生むものでもありません。本資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMまたはセッション発表者は責任を負わないものとします。本資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引きだすことを意図したものでも、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでもなく、またそのような結果を生むものでもありません。

本資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本資料に含まれている内容は、参加者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したものでも、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴは、米国やその他の国におけるInternational Business Machines Corporationの商標または登録商標です。他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。現時点でのIBMの商標リストについては、[ibm.com/trademark](http://ibm.com/trademark)をご覧ください。