

いますぐ始める 数理生命科学

MATLAB プログラミングから
シミュレーションまで

佐藤 純 著

コロナ社

は じ め に

分子生物学，発生生物学，細胞生物学，神経科学などの生命科学の諸分野において，データの統計解析のみならず数理モデリング，数値シミュレーション，画像解析などの数理・情報系の知識や技術が必要となる場面が増えています。物理や工学の分野では，対象とする現象を微分方程式に置き換えた数理モデルを用いることで，新たな現象を予測したり，現象の背景にあるメカニズムを明らかにする手法が一般的ですが，生命科学の分野でも，数理モデルを取り入れた数理生物学が注目されています。

このような研究を理解し，推進するためには，コンピューターのプログラミングを習得する必要がありますが，多くの生命科学研究者や学生の方々はそのような教育を受ける機会がなかったり，苦手意識のせいでなかなか始められないという場合が多いと思われます。筆者自身は生命科学者で，プログラミングは趣味としてやっていた程度ですが，最近のプログラミング言語は非常に扱いやすく，またコンピューターも高性能になっているので，素人でも実用的なプログラムを作成できるようになっています。シミュレーションを行うということは，微分方程式を解くプログラムを作成するということとほぼ同じ意味になりますが，そのためには，本来，きちんと数学を学ぶ必要があります。ですが，プログラミングと数学を両方学ぶのは非常に大変ですので，本書では数学の詳細はできるだけ割愛し，プログラミングができるようになることを第一の目的としています。高校レベルの数学と物理の知識はあったほうがよいですが，最近では高校で習わなくなった行列や微分方程式については，必要に応じて解説しています。

世の中にはさまざまなプログラミング言語が存在しますが，中でも本書では MathWorks 社が販売している MATLAB を取り扱います。ただ，プログラム言

語というのはどれも基本的な考え方は似通っていて、一つの言語を習得すれば他の言語を扱うのはそれほど難しいことではありません。もちろん言語によってそれぞれ特徴があり、MATLAB の場合は行列の扱いが得意であることが大きな利点です。データのグラフ化が簡単なので、たった 1 行の命令で見栄えのよいグラフを作成することができます。また、ツールボックスと呼ばれるオプションを入手することで、ディープラーニングのような人工知能の技術も簡単に取り入れられます。MATLAB は有料であり、せっかく作ったプログラムでも、MATLAB がインストールされたコンピューターでないと使うことができないという問題点はありますが、アカデミック価格や学生価格であればかなり入手しやすくなっていますし、MATLAB Home と呼ばれる安価なホームライセンスも用意されています。また、1 か月間であれば無料の試用版を使うこともできますし、GNU Octave (以後 Octave と呼びます) という MATLAB と互換性のあるプログラミング言語であれば、無料でダウンロードすることができます。

少しでも興味を持たれた方は、ご自分のコンピューターに MATLAB もしくは Octave をインストールし、いまずぐ始めてみてはいかがでしょうか？ 少し難しく感じられることもあるかもしれませんが、意外とこんなに単純なものなんだ、と思ってもらえればそれだけでも大きな一歩です。数理系の発表を聞いても物怖じせずに内容を理解できるようになるかもしれません。本書のカバーする範囲は限られていますが、これをきっかけに、より専門的なスキルを身に付けていただければ、それが近い将来、生命科学の進歩につながると期待しています。チャレンジしていただければ幸いです。

なお、本書の執筆にあたっては以下多数の方々からさまざまなコメントをいただき、何とか完成にこぎつけました。この場を借りて感謝申し上げます。

北海道大学 長山雅晴先生、はこだて未来大学 田中吉太郎先生、秋田大学 山崎正和先生、鮎川友紀先生、新潟大学 杉江淳先生、産業技術総合研究所 古藤日子先生、東京大学 多羽田哲也先生、三浦正幸先生、樫尾宗志朗先生、早稲田大学 花嶋かりな先生、京都大学 井垣達吏先生、和田弥生さん、吉田大祐さん、大阪大学 近藤滋先生、稲木美紀子先生、基礎生物学研究所 小山

宏史先生，沖縄科学技術大学院大学 清水一道さん，Goethe-Universität Frankfurt 青木一郎さん，金沢大学 向田直史先生，井上啓先生，三枝理博先生，津野祐輔先生，瓜生耕一郎先生，当研究室大学院生 友水豪志さん，内藤寿稀さん，MathWorks 社 吉田幸恵さん，林涼子さん，沖田芳雄さん，村松未輝雄さん，廣田康二さん，井上道雄さん，大開孝文さん，菱田寛之さん

すべてのコメントに十分に対応できたかはわかりませんが，できるだけ読みやすく，実用的で，かつこれからの数理生物学に少しでも貢献できる内容にするよう努力したつもりです。また，出版にあたってコロナ社にはたいへんお世話になりました。この場を借りて御礼申し上げます。

2020 年 11 月

佐藤 純

MATLAB は MathWorks, Inc. の登録商標です。本書では，MATLAB およびその他の製品名にTM，®マークは明記しておりません。

本書を発行するにあたって，記載内容に誤りがないように可能な限り注意を払いましたが，本書の内容を適用した結果生じたこと，また，適用できなかったことに関して，筆者，コロナ社とも一切の責任は負いませんのでご了承ください。

目 次

1 準備運動

| | | |
|-------|----------------|----|
| 1.1 | プログラミングとは | 1 |
| 1.2 | インストール | 2 |
| 1.3 | 関数電卓として使ってみる | 2 |
| 1.3.1 | コマンドウィンドウ内での計算 | 2 |
| 1.3.2 | 関数の計算 | 5 |
| 1.3.3 | 逆関数の計算 | 6 |
| 1.4 | 変数 | 7 |
| 1.4.1 | 変数とは | 7 |
| 1.4.2 | 変数の名前 | 9 |
| 1.5 | ベクトル | 10 |
| 1.5.1 | ベクトルとは | 10 |
| 1.5.2 | ベクトルの要素を操作する | 12 |
| 1.6 | ベクトルを使ったグラフの描画 | 14 |
| 1.6.1 | plot を使ってみる | 14 |
| 1.6.2 | plot の応用 | 16 |
| 1.7 | 行列 | 19 |
| 1.7.1 | 行列を定義する | 19 |
| 1.7.2 | 行列の大きさ | 23 |
| 1.8 | 行列の計算 | 24 |
| 1.8.1 | 行列の和と積 | 24 |
| 1.8.2 | アダマール積 | 26 |

2 MATLAB プログラミング入門

| | |
|-------------------------------|----|
| 2.1 スクリプト | 29 |
| 2.1.1 スクリプトを作る | 29 |
| 2.1.2 グラフを描くスクリプトを作る | 32 |
| 2.2 円 と ら せ ん | 36 |
| 2.2.1 極座標を使って円とらせんを描く | 36 |
| 2.2.2 円とらせんを描く 応用編 | 38 |
| 2.3 for 文 | 42 |
| 2.3.1 for 文でループを作る | 42 |
| 2.3.2 for 文のネスト | 44 |
| 2.4 数値の整数化 | 46 |
| 2.5 数値の文字化 | 47 |
| 2.6 数値と文字が混在した文章 | 48 |
| 2.7 while 文 | 50 |
| 2.8 じゃんけん と if/switch 文 | 54 |
| 2.8.1 if 文による条件判定 | 54 |
| 2.8.2 switch 文による条件判定 | 56 |
| 2.9 関 数 | 60 |
| 2.9.1 じゃんけんを判定する関数 | 60 |
| 2.9.2 スクリプト内に関数を配置する | 63 |
| 2.10 2次元画像の扱い | 65 |
| 2.10.1 imagesc による画像表示 | 65 |
| 2.10.2 2次元パターンを作り出す | 69 |
| 2.11 アニメーション | 73 |
| 2.11.1 plot のアニメーション | 73 |
| 2.11.2 imagesc のアニメーション | 76 |
| 2.11.3 ムービーファイルの保存 | 79 |
| コラム 1 デバッグについて | 81 |

| | | |
|--------|--------------------------|----|
| 2.12 | グラデーションリング | 81 |
| 2.12.1 | グラデーションリングを描く | 81 |
| 2.12.2 | グラデーションリングを回転させる | 87 |
| コラム 2 | よくあるバグ その 1: 打ち間違い | 91 |

3 生命現象の時間変化をシミュレーションする

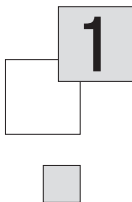
| | | |
|-------|---------------------------------|-----|
| 3.1 | 微分方程式を解く | 93 |
| 3.2 | 蛋白質の合成と分解 | 95 |
| 3.2.1 | オイラー法による微分方程式の解法 | 95 |
| 3.2.2 | 厳密解との比較 | 99 |
| コラム 3 | よくあるバグ その 2: 文法エラー | 103 |
| 3.3 | 造血幹細胞の数理モデル | 104 |
| 3.3.1 | 造血幹細胞の数理モデル その 1 | 104 |
| 3.3.2 | 造血幹細胞の数理モデル その 2 | 108 |
| 3.3.3 | ロジスティック方程式 | 111 |
| 3.4 | 糖代謝の数理モデル | 113 |
| コラム 4 | よくあるバグ その 3: 数値や文字の扱いが不適切 | 118 |
| 3.5 | 伝染病の数理モデル | 120 |
| 3.5.1 | SIR モデル | 120 |
| 3.5.2 | 複数のパラメーターの結果を比較する | 123 |
| 3.5.3 | 基本再生産数 R_0 を導入する | 129 |
| 3.6 | ロトカ・ヴォルテラモデル | 131 |
| 3.6.1 | オイラー法を用いた数値計算 | 131 |
| 3.6.2 | ode45 を用いた数値計算 | 134 |
| コラム 5 | ブレイクポイントの設定 | 138 |

4 生命現象の時間・空間変化をシミュレーションする

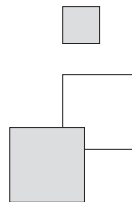
| | | |
|-----|-----------------------------|-----|
| 4.1 | 拡散性物質による空間パターン形成の時間変化 | 140 |
| 4.2 | 拡散方程式の直観的な説明 | 141 |

| | |
|--|-----|
| 4.3 拡散方程式の計算 | 143 |
| 4.3.1 拡散の数値計算アルゴリズム | 143 |
| 4.3.2 for 文を用いた拡散の計算 | 147 |
| 4.3.3 del2 を用いた拡散の計算 | 153 |
| 4.3.4 行列を用いた拡散の高速計算 | 155 |
| コラム 6 よくあるバグ その 4: アルゴリズムがおかしい | 157 |
| 4.4 モルフォゲンによるパターン形成 | 158 |
| 4.5 ショウジョウバエ翅のパターン形成 | 161 |
| 4.5.1 Dpp の拡散の計算 | 161 |
| 4.5.2 Dpp の標的遺伝子の発現 | 164 |
| 4.6 ショウジョウバエ脚のパターン形成 | 167 |
| 4.6.1 Dpp と Wg による <i>Dll</i> 遺伝子の発現制御 | 167 |
| 4.6.2 活性化型 Dpp 受容体の効果 | 172 |
| コラム 7 よくあるバグ その 5: 数学的な問題 | 176 |
| 4.7 チューリングモデル | 176 |
| 4.7.1 活性化因子と抑制因子からなる反応拡散方程式 | 176 |
| 4.7.2 濃度の上限を設定する | 180 |
| 4.7.3 if 文を使わずに濃度の上限を設定する | 182 |
| 4.7.4 チューリングパターン | 185 |
| コラム 8 パラメーターの設定 | 192 |
| 4.8 心臓拍動のシミュレーション | 193 |
| 4.8.1 フィッツヒュー・南雲方程式 | 193 |
| 4.8.2 フィッツヒュー・南雲方程式に拡散を加える | 195 |
| 4.8.3 不整脈の数値シミュレーション | 198 |
| 4.8.4 心臓拍動のメカニズム | 199 |
| コラム 9 dt と dx の値 | 202 |
| 4.9 「分化の波」の数理モデル | 203 |
| 4.9.1 分化の波, proneural wave | 203 |
| 4.9.2 EGF と AS-C からなる 2 変数モデル | 207 |
| 4.9.3 Notch と Delta を加えた 4 変数モデル | 211 |

| | | |
|---------|-------------------------------|-----|
| 4.9.4 | EGF 変異体における波の消失 | 215 |
| 4.9.5 | Notch 変異体における波の加速のメカニズム | 216 |
| 4.9.6 | なぜゴマシオにならないのか? | 219 |
| コラム 10 | 数式のデザイン | 222 |
| 付 | 録 | 225 |
| 引用・参考文献 | | 233 |
| お わ り に | | 236 |
| 索 | 引 | 238 |



準備運動



1.1 プログラミングとは

最近のコンピューターやスマートフォンは非常に賢いので、その頭脳ともいえる CPU（中央処理装置の略）はさぞかし賢いのだろうと思われるかもしれませんが、じつはそうでもありません。CPU は足し算や引き算といったレベルの非常に単純な計算しかできないのですが、単純な計算をうまく組み合わせ、それを超高速で実行することによって、表面上は賢く見えるのです。そのためには、人がコンピューターに実行させるためのプログラムを**プログラミング言語**を使って作る必要があります。

このプログラミング言語にはさまざまなものがありますが、CPU が直接理解できる言語は**機械語**と呼ばれる、最も基本的な言語です。ですが、本来 CPU が理解できることは非常に限られているので、機械語でプログラミングすることは人にとって非常に困難です。そこで、一般的には人にとってより理解しやすい**高級言語**を使ってプログラミングすることになります。この高級言語にもさまざまなものがあるのですが、一般的に使われる言語は大きく分けて**コンパイラー**と呼ばれるタイプと**インタープリター**と呼ばれるタイプがあります。コンパイラー用に使われたプログラムは一度まとめて機械語に翻訳（**コンパイル**）してから実行する必要があるのですが、少し手間が掛かりますが、プログラムの実行速度は速くなります。一方、インタープリター用のプログラムは一つひとつ、命令をその都度翻訳しながら実行されます。そのため実行速度は遅くな

りますが、翻訳という過程を意識せずに、プログラムを自由に編集したり実行したりできるので、非常に扱いやすくなっています。MATLAB はインタープリター言語なので、プログラミングの初心者でも扱いやすいというわけです。

1.2 インストール

まず MATLAB をインストールしてみましょう。MATLAB の無料評価版は 30 日間であれば無料で利用できますので、まずは MathWorks 社のウェブサイトからインストールするのがよいでしょう。「MATLAB 評価版」と検索すればすぐに見つかると思います。学生の方は大学生協などで所定の手続きをすれば学生価格で購入できます。MATLAB 本体だけでしたら、5 千円程度と学生でも十分に購入できる価格です。また、MATLAB Home と呼ばれるホームライセンスは 1 万 5 千円程度です。この機会に購入してみてはいかがでしょう？

もしくは MATLAB と互換性のあるフリーの言語である Octave をインストールして使われても構いません。GNU Octave のウェブサイトからダウンロードすることができます (<https://www.gnu.org/software/octave/>)。macOS の場合はこちら (https://wiki.octave.org/Octave_for_macOS) のほうがよいかもしれません (dmg ファイルを右クリックして開いてください)。インストールすると、コマンドラインインターフェース (CLI) 版とグラフィカルユーザーインターフェース (GUI) 版が使えるようになると思いますが、GUI 版のほうが MATLAB と近い感覚で使うことができてわかりやすいと思います。MATLAB とはレイアウトなどが少し異なりますが、大きな違いはないので、特に断わらない限り、本書では MATLAB を使っていることを想定して説明を進めます。

1.3 関数電卓として試してみる

1.3.1 コマンドウィンドウ内での計算

無事、MATLAB をインストールして起動すると、図 1.1 のような画面が表

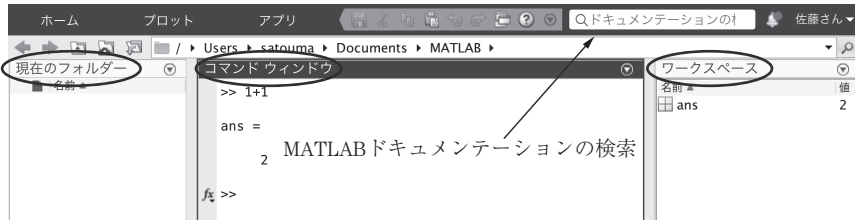


図 1.1 MATLAB の画面

示されると思います。ウィンドウは三つの枠から構成されており、左から「現在のフォルダー」、「コマンドウィンドウ」、「ワークスペース」と書かれています。MATLAB をインストールした場合は「MATLAB」という名前のフォルダーが作られ、これが「現在のフォルダー」となります。「ワークスペース」には MATLAB 内で使用している変数（詳しくは 1.4 節参照）の情報が表示されます。まずはこの「コマンドウィンドウ」をクリックしてください。すると、`>>` の右に点滅する縦棒が現れると思います。これはカーソルと呼ばれ、コンピュータのキーボードから文字を入力できる状態になっていることを意味します。

また、画面右上には虫眼鏡が描かれたボックスがありますが、ここにわからないコマンド名や単語を入力すると、関連した文書を **MATLAB ドキュメンテーション** と呼ばれる公式な説明書の中から検索することができます。詳細な

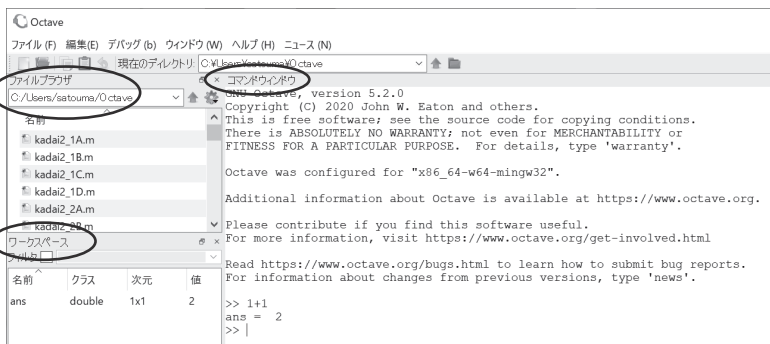


図 1.2 Octave の画面

説明が必要な場合はこの機能を活用しましょう。Octave は少しレイアウトが異なりますが、基本的には同じです (図 1.2)。

MATLAB や Octave はインタープリター言語なので、ユーザーが意識して命令を翻訳する必要がありません。そのため、プログラムを書かなくてもいきなり命令を実行させることができます。例えば、あたかも関数電卓のように計算させたりグラフを描かせたりすることができるのです。まずは関数電卓のように使ってみましょう。

コマンドウィンドウ中に

```
1+1
```

と入力し、Enter キーを押して改行してみてください。すると

```
ans =  
2
```

と表示されたと思います。つまり、 $1+1=2$ という足し算をしてくれたということです。もちろん引き算もできますし、整数だけでなく小数点も扱うことができます。

```
4.1-5.2
```

と入力し、Enter キーを押して改行するとつぎのように表示されます。

```
ans =  
-1.1000
```

掛け算と割り算は一般的な数学の記号とは異なり、**掛け算はアスタリスク (*)**、**割り算はスラッシュ (/)** を使います。 $3.2*1.5$ と入力すると答えは 4.8000 と表示されますし、 $3.14/2$ と入力すると答えは 1.5700 と表示されます。

では、 2^3 と入力したらどうなるでしょうか？ この場合、答えは 8 と表示されます。そう、**^ はべき乗**するという意味です。

数学では円周率は π と書きますが、コンピュータのキーボードに π はないので、MATLABではアルファベットでpiと書きます。試しにpiと入力し、Enterキーを押して改行してみてください。3.1416と表示されるでしょう。以後、本章では、つねにコマンドウィンドウ中に入力し、Enterキーを押して改行してください。

1.3.2 関数の計算

平方根も計算できます。ルートの記号もキーボードにはありませんが、sqrtという**関数**を使います。ここでいう関数とは、数学でいう関数と似たようなもので、括弧内の数値を使って何かの計算をしてくれます。つまり、sqrt(5)と入力し、改行すると、2.2361と答えてくるのです。もちろん、括弧内には好きな数字を入れて構いません。

では試しにsqrt(-2)と入力したらどうなるでしょうか？ 何と0.0000+1.4142iと答えてきます。このiとはもちろん**虚数**のことで、iの二乗は-1となります。MATLABは特別な処理をしなくてもそのまま虚数に対応してくれるのです。試しにi*iと入力すれば-1と答えてくれますし、(1+2i)*(1-2i)では5と答えてくれます。

先ほど関数の例としてsqrtが出てきましたが、もちろんこれ以外にもさまざまな関数を使うことができます。absは**絶対値**を求める関数ですので、例えばabs(3-5)とすると、-2の絶対値である2と答えてくれます。

ほかにも、**三角関数**が一通り使えます。例えば $\sin \frac{\pi}{2}$ はsin(pi/2)と表記し、1が答えとなります。cos(pi)は-1、tan(pi/4)は1となります。ではちょっと意地悪してtan(pi/2)としたらどうなるでしょうか？ 数学では $\tan \frac{\pi}{2}$ は $+\infty$ ですが、じつはコンピューターにとって無限大という概念は扱いにくいものなのです。この場合、答えは1.6331e+16と表示されます。これは一体何でしょうか？

その前に、ここで100000*100000と入力してみてください。これはかなり大きな数で、0が五つ掛ける0が五つなので、0が10個も付くことになりま

索

引

| | | | | | |
|------------|-----|------------|------------|------------|----------|
| 【あ】 | | 【き】 | | 【さ】 | |
| アダマール積 | 26 | 機械語 | 1 | 三角関数 | 5 |
| アルゴリズム | 157 | 基本再生産数 | 129 | 【し】 | |
| 【い】 | | 逆関数 | 6 | シグナル伝達 | 158 |
| 閾 値 | 193 | ギャップ結合 | 195 | 自己複製 | 104 |
| イコール | 55 | 行 | 19 | 四捨五入 | 46 |
| 一次関数 | 103 | 行 列 | 10, 19 | 周期境界条件 | 149 |
| 遺伝子 | 158 | 極 限 | 6, 95, 144 | 収 束 | 99 |
| インスリン | 113 | 極座標系 | 15 | 受容体 | 140, 158 |
| インスリン抵抗性 | 117 | 極座標 | 36 | 常微分 | 141 |
| インタープリター | 1 | 虚 数 | 5 | 常微分方程式 | 103 |
| インデックス | 21 | 切り上げ | 46 | 情報伝達 | 158 |
| インデント | 46 | 切り捨て | 46 | 自律神経 | 195 |
| 【え】 | | 【く】 | | 心 筋 | 195 |
| エビジェネティック | 207 | グローバル変数 | 136 | 神経幹細胞 | 207 |
| 【お】 | | 【け】 | | 神経上皮細胞 | 207 |
| オイラー法 | 98 | 血糖値 | 113 | 心 室 | 195 |
| オブジェクト | 80 | 現象論的 | 177 | 心 房 | 195 |
| 【か】 | | 厳密解 | 94, 99 | 【す】 | |
| 拡 散 | 140 | 【こ】 | | 数値解 | 94 |
| 掛け算 | 4 | 高級言語 | 1 | 数 列 | 11 |
| 活性化因子 | 177 | 勾 配 | 141 | スカラー | 10 |
| 活動電位 | 193 | 誤 差 | 134 | スクリプト | 29 |
| カリウムイオン | 193 | コマンド | 14 | 【せ】 | |
| カリウムチャネル | 193 | コメント | 31 | 正規分布 | 220 |
| 幹細胞 | 104 | コロン | 13 | 静止膜電位 | 193 |
| 関 数 | 5 | コンパイラー | 1 | 絶対値 | 5 |
| | | コンパイル | 1 | セミコロン | 8 |
| | | | | 漸 近 | 99 |
| | | | | 線型常微分方程式 | 103 |

線型微分方程式 103
染色体 207

【そ】

増殖 104
草食動物 132
相図 192
相平面 201
相平面解析 201
側方抑制 204

【た】

大括弧 19

【ち】

チューリングモデル 177, 185

【て】

デバッグ 81
転写 158
転写因子 158
転置 25

【と】

統計力学 141
洞結節 195
等号付き不等号 69
糖尿病 116
ドミナントネガティブ 222

【な】

内積 25
ナトリウムイオン 193
ナトリウムチャネル 193

【に】

肉食動物 132

【ぬ】

ヌルクライン 200

【ね】

ネスト 44

【の】

ノックアウト 222
ノックダウン 222

【は】

配列インデックス 118
バグ 81
バグ取り 81
発散 164
反応拡散方程式 161

【ひ】

非線型常微分方程式 111
非対称分裂 207
被捕食者 132
ピリオド 15

【ふ】

フィッツヒュー・南雲方程式 194
不応期 194
不整脈 198
不等号 69
ブドウ糖 113
プログラミング言語 1
プログラム 29
分化 104
分化度 207
分化の波 203
分泌性リガンド 140, 159

【へ】

平衡点 200

べき乗 5
ベクトル 10
ベクトル場 201
変数 7
偏微分 141
偏微分記号 141
偏微分方程式 138

【ほ】

房室結節 195
ホジキン・ハックスレーモデル 193
捕食者 132
ホルモン 113

【ま】

膜結合型リガンド 158
膜電位 193

【み】

未分化 207

【も】

モザイク解析 172, 215
文字列 47
モルフォゲン 159, 161

【よ】

要素 11
抑制因子 177

【ら】

ラプラシアン 145, 153

【り】

リガンド 158
履歴 18

【る】

ループ 42

ルンゲ・クッタ法

135

【れ】

列

19

【ろ】

ロジスティック増殖 112

ロジスティック方程式 112

ロトカ・ヴォルテラモデル

131

【わ】

割り算

4

【B】

BMP4

161

【C】*cis*-inhibition

204, 211

【D】

Decapentaplegic

159

DNA 配列

158

Dpp

159

【K】 K^+

193

【L】

lateral inhibition

204

【N】 Na^+

193

【P】

proneural wave

203

【R】 R_0

129

RNAi

222

【S】

SIR モデル

120

【T】*trans*-activation

204, 211

【W】

Wg

168

Wingless

168

Wnt

169

【その他】

1 型糖尿病

116

2 階微分

142

2 型糖尿病

116

 ∂

141

 Δ

145

— 著 者 略 歴 —

1995 年 東京大学理学部生物化学科卒業
1997 年 東京大学大学院理学系研究科修士課程修了（生物化学専攻）
2000 年 東京大学大学院理学系研究科博士課程修了（生物化学専攻），博士（理学）
2000 年 カリフォルニア大学サンフランシスコ校博士研究員
2002 年 東京大学助手
2007 年 東京大学助教
2008 年 金沢大学特任准教授
2012 年 金沢大学教授
現在に至る

いますぐ始める数理生命科学

— MATLAB プログラミングからシミュレーションまで —

A Practical Introduction to Mathematical Life Sciences

— From MATLAB Programming to Simulations —

© Makoto Sato 2021

2021 年 1 月 8 日 初版第 1 刷発行



検印省略

| | |
|-------|------------|
| 著 者 | 佐 藤 純 |
| 発 行 者 | 株式会社 コロナ社 |
| 代 表 者 | 牛 来 真 也 |
| 印 刷 所 | 新日本印刷株式会社 |
| 製 本 所 | 有限会社 愛千製本所 |

112-0011 東京都文京区千石 4-46-10

発 行 所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844 ・ 電話 (03) 3941-3131 (代)

ホームページ <https://www.coronasha.co.jp>

ISBN 978-4-339-06762-0 C3045 Printed in Japan

(齋藤)



JCOPY <出版者著作権管理機構 委託出版物>

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-5244-5088, FAX 03-5244-5089, e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。