Newsletter of the Japanese Society for Mathematical Biology No. 92 September 2020

日本数理生物学会

ニュースレター

第９２号

2020年9月

【2019 年研究奨励賞受賞者特別寄稿】

これまでの研究生活を振り返って

黒川瞬 *

1.　はじめに

研究奨励賞を受賞できたこと、とても嬉しく思います。また、選考委員の 6 名の先生方には、私が研究奨励賞を受賞するにふさわしいか否かの判断のために貴重な時間を割いて頂き、感謝しております。今後、この受賞を励みに私が研究を進めることで報いていければと考えています。ありがとうございました。

仕事を終えて自宅に帰ってから寝るまでの私の時間の過ごし方の一つとして、数理生物学会のニュースレターをリラックスした気持ちで読むというのがあります。賞を頂けただけでなく、文章を書いて寄稿する機会まで頂けるということで、今回書き手に初めて回ります。緊張していますが、楽しんで書きたいと思います。

私の数理生物学会での活動は 2 回のポスター発表に限られます。1 回目は、会社員として働いていた時に同志社大学で 2015 年 8 月に行われた大会で、2 回目は、アカデミアに戻ってきてから高知工科大学在籍時に東京工業大学で 2019 年 9 月に行われた大会です。口頭発表をしたことも、シンポジウムで発表したこともないので、私のことを知っている方は数理生物学会員の中であまりいないと思います。ですので、これまでの私がしてきたことの内、数理生物の研究に関連すると考えられる部分をメインに、振り返って書いてみたいと思います。

2.　大学入学以前

大学入学以前は、問題集にある問題を拡張するなどして、数学および物理の問題を作って解く、ということをしていました。解けなかったら簡単に、解けたら難しくし、「自分が解ける範囲で最難関の問題は何か？そして、その答えは何か？」という課題に取り組んでいました。数理生物という分野なので、数学から入っているか生物から入っているか、ということがありますが、このように、私のバックグラウンドは数学です。数学と物理が好きでしたし試験で得点をとることも得意でした。

*東京大学大学院総合文化研究科

3.　大学時代

2003 年 4 月に東京大学に理科一類で入学し、大学では部活でボートに専念することにしました。3 年進級時の進学振り分けでは、理学部地球惑星物理学科に進むことにしましたが、勉強は部活引退の 4 年生の 8 月まではあまりしませんでした。ボートは苦手でしたが、チームメイトの力も借り、全日本軽量級選手権という大会においてエイトという種目で 4 位になれたことがせめてもの救いです。部活を引退するまでは、就職活動を行っていなかったため就職先もなければ、勉強しなかったため取得単位も少なく、それに加えて、大学院入試も受けていませんでした。ボート部を引退後、大学を 5 年で卒業し、その後民間企業で就職することを目指して少し就職活動をしましたが、内定を得ることはなく、大学院修士課程に進むことに目標を変更しました。

しかし、地球惑星物理学科のまま修士課程に進むことは、選択肢から消すことになります。きっかけの一つは、地球惑星物理学科の同期の仲間と一緒にグアムに行った卒業旅行（私はこの年には卒業できませんでしたが）です。同期の仲間は風が吹いただけでもその理由について、1 時間ほど議論に熱中していました。それを見て、東京大学の地球惑星物理学科は、そういった話題において興味関心的にも能力的にも二十歳前後の世代では日本最高レベルの人達が集まっていると考えられるので納得はできる一方で、勉強はそれまでしていなかったので勉強して追いつくにしても、熱意の部分で私はついていけないと考えました。同期の仲間との差が大きく、埋めるのが大変に思えました。

修士課程に進むにあたり地球惑星物理学科のまま進むことを選択肢から除いた私は、別の学科を考えるわけですが、私の状況は、数学は自分の中では得意であるし好きでもあるが、大学に入ってからは数学をしっかりと学んでいないというものでした。修士課程に進むからには少なくとも修士論文は書けるような研究室をもちろん選びたいので、「数学は使うけれども、自分が理解できないような高度な数学は使わない研究をする研究室」という観点で大学院での研究室を選ぶことになります。

数学を使うということで理学系研究科か、工学系研

究科が思い浮かびましたが、工学系研究科に進むことは選択肢から除外し、理学系研究科の中から選択しようと思いました。1年生か2年生の頃に受けたオムニバス形式の「工学概論」という授業に関心を持てなかった経験をよりどころにしました。

理学系の授業は、大学院でどの専攻、どの研究室に進むかを決める参考にするため天文学や化学など幅広く受けていましたが、（1）系統樹の数がOTUsのどのような関数になっているかに関する内容などを含む、山岸明彦教授の授業（2）進化の定義、ハーディー・ワインベルクの法則などの内容を含む、青木健一教授の集団生物学の授業（3）採餌効率、進化ゲーム理論、ＥＳＳ (Evolutionarily Stable Strategy)、繰り返し囚人のジレンマゲームなどの内容を含む、井原泰雄講師の行動生態学の授業、など生物学の授業も受けました。その結果、数学的な要素を含み、かつ、生物以外の分野と比較して数学的には簡単な問題が生物学の中では扱われることがあることを知りました。大学での数学をしっかりと学んでいない自分でも理解できる数学的なトピックがあると知れたのは大きな喜びでしたし、修士論文を書き修士号がとれそうな分野があることを知れたことは安堵でもありました。上で書いた3つ目の行動生態学の授業担当の井原泰雄講師は年齢も近く話しやすかったことなどから、井原講師の研究室に入るのを目指すことにしました。「理論生物学入門」、「生き物の進化ゲーム」もこの頃に読むようになりました。大学院入試は合格することができたので、結果、大学には5年間在籍し、大学院修士課程に進学することになります。

修士課程に進むことは決めたものの、修士号取得後は博士課程には進まずに民間企業に就職することを目指していました。具体的な職種は、アクチュアリーという仕事です。この仕事は、就職活動失敗後に書店で立ち読みしていた際に知った仕事なのですが、数学を使って保険料の計算などをする仕事です。「日本アクチュアリー会」で検索すると詳しい説明が出てきますが、1年に1回試験が行われ、一次試験5科目、二次試験2科目からなります。一次試験の5科目にすべて合格するとアクチュアリー準会員になり、二次試験の受験資格を得ます。二次試験の2科目にも合格するとアクチュアリー正会員になります。（1）修士1年までにアクチュアリーの資格試験にある程度合格し、（2）修士号を取得し、（3）取得した後は保険会社に入ることが、大学院に入る前に思い描いた大学院修士課程での目標になりました。（1）は、学部時代の就職活動の失敗経験から、就職活動時に採用されやすくなるように備える必要性を感じていたためです。

表1　2人ゲームの利得行列

自分の戦略	相手の戦略 A	相手の戦略 B
A	a_1	a_2
B	b_1	b_2

4.　東京大学大学院修士時代

2008年4月から2010年3月までを修士課程の大学院生として過ごしました。一見損であると思われる行動を見つけてきてその適応的意義を探るのが、進化生物学の大きなトピックです。協力行動は「自分にとって損であり、相手にとって得である行動」と、定義されることがありますが、自分にとって損であるという性質が定義に入っているため、その存在は説明を要すると即座に言えます。修士1年に上がる直前の3月頃に協力行動の進化に関するreview paperであるNowak(2006)を読むように井原泰雄講師にすすめられました。そして、その論文の中で綺麗で、かつ、感覚にあう式を発見しました。表1において、$a_1+2a_2>b_1+2b_2$ならば戦略Aが有利というものでした。学部5年目の際に、井原泰雄講師の授業で、戦略BがＥＳＳであるための条件は$a_2<b_2$であるという話を習っていましたが、当時の私は、「戦略Ｂばかりの集団からスタートさせた場合、a_2,b_2の重みがa_1,b_1の重みよりも大きいのは理解できるが、a_1,b_1が全く進化に影響を与えないなんてことはなく、$(a_1-b_1)+m(a_2-b_2)$ $(m>1)$の正負で、進化は決まるはずである」という感想を持っていたので、とてもしっくりきました。その式はNowak et al.(2004)で導出されているということがNowak(2006)に書かれてあったので、Nowak et al.(2004)を読みました。そこには簡単に理解できる数理モデルと高校数学の内容だけによる式の証明が書かれていました。

上述の式である$a_1+2a_2>b_1+2b_2$は、二者間での相互作用についての式であったため、修士1年の5月頃、一般のn者間での相互作用についての式を導出することを試み、実際に導出することができました。その式は、表2のように、利得行列をおいた場合、$\sum_{k=1}^{n} ka_k > \sum_{k=1}^{n} kb_k$を満たせば、戦略Aが進化するという式でした。$n=2$から一般の$n$に拡張した結果、綺麗さが保持されていることに嬉しくなったことを今でも覚えています。私はこの結果を論文にまとめ、修士1年の10月に、Proceedings of the Royal Society Biological Sciences Bに投稿しました。結果はminor revisionであり修士1年の12月に受理されました（Kurokawa & Ihara, 2009）。受理された時に、「よかった」と言ってくださった時の井原先生の笑顔を今でもしっかりと記憶しています。修士論文が書けるかどうかさえ不安で、修士論文が書けそうな研究室を血眼になって探した私でしたが、「日本学術振興会特別

表 2 n 人ゲームの利得行列

自分の戦略	$n-1$ 個体の相手のうち戦略 A をとる個体の数					
	$n-1$	$n-2$	$n-3$	$\cdots$	1	0
A	a_1	a_2	a_3	$\cdots$	a_{n-1}	a_n
B	b_1	b_2	b_3	$\cdots$	b_{n-1}	b_n

研究員ＤＣ１が通る確率が高い」とこの受理でもって判断し、博士課程に進む計画に変えました。しかし、「保険会社に就職しアクチュアリーになる」という人生設計に変わりはなく、当時の人生設計は「博士号取得後、保険会社に就職しアクチュアリーになる」というものでした。修士時代の 2 年間で、第一著者の論文が 2 本受理されました。

5. 東京大学大学院博士時代

2010 年 4 月から 2013 年 3 月までを博士課程の大学院生として過ごしました。アクチュアリーの就活は、学部卒および博士卒と比較して修士卒が優遇されると言われていました。そこで、修士卒に対して不利な側面を補うために、アクチュアリーの試験は毎年 1 回 12 月に行われますが、博士 1 年の 12 月まではアクチュアリー資格試験の受験勉強を集中的に進めました。時間を投入したことも実り、その 12 月の受験でもってアクチュアリー準会員になりました。

博士 1 年の 12 月にアクチュアリー試験が終わってから博士 2 年の夏頃までは、研究をできるだけ進めるようにしました。博士 2 年の夏から博士 3 年の春までは、就職活動で毎日平均して 2 社、3 社、日によっては 4 社、会社説明会もしくは面接に行くような生活になり、就職活動一辺倒という感じでした。そのため、研究が再開できるのは、博士 3 年の春、生命保険会社に内定をもらってからになりました。博士 3 年で博士号が取得できてよかったのですが、就活を有利に運ぶための資格試験の勉強と就活そのものに多大な時間を投入することになり、結果としてあまり研究を進めることなく研究生活をいったん終えました。研究をすることは好きではありましたが、論文を書くことはこの時はあまり好きにはなれませんでした（今では論文を書くことは好きです。）。また、英語で論文を書く必要がありますが、英語を母国語とする研究者もいることに不平等を感じてもいました。博士時代の 3 年間で、第一著者の論文が 1 本、それ以外の論文が 1 本受理されました。

6. 会社員時代

2013 年 4 月から会社員として過ごしました。準備も実り、博士号取得後アクチュアリー候補として生命保険会社で働くことができました。多くの時間を投入し、周到な準備と計画性をもって、アクチュアリー候補として生命保険会社で働く権利を得たのですが、結局 2 年と 5 か月で辞めることになりました。会社の業務内容は論文執筆ではもちろんありませんが、無理やり論文執筆に換算すると、会社の先輩方の業務を進めるスピードは 1 年に 15 本くらい論文を書くスピードであったように思います。すさまじく速く、会社の先輩方が優秀すぎると感じていました。会社員時代での仕事は、その業務が発生した経緯を知らないため、する理由がわからない業務もたくさんありましたが、アカデミアに戻ってきてから同様のことをしてみると大変役立つことも数多くあります。この間はもちろんですが、論文を生産することはありませんでした。

7. 京都大学博士研究員時代

2015 年 9 月から 2016 年 7 月までの 11 か月在籍しました。一度会社に入ってから、アカデミアへ復帰するための就職活動は大変だったと記憶していますが、京都大学の栗山浩一教授の研究室で博士研究員として雇用されました。当時 31 歳で、論文がたった 4 本。恐怖の中で、研究を再開することになります。栗山教授の研究室は、農学部の環境経済学の研究室です。「自分がしたい研究を好きなように進めていい」と雇用主である栗山教授にはおっしゃって頂けたので、東京大学大学院生時代と同様、京都大学在籍時には進化ゲーム理論を用いた協力行動の進化の理論研究を進めました。

東京大学大学院（修士・博士）時代は、研究室内のメンバーは進化ゲーム理論を知っていましたし、基本的には指導教員である井原泰雄講師と十分に議論して共著で仕上げていました。ですので、当時は自分の研究の話を聞いてもらうことができましたが、京都大学では、進化ゲーム理論を知っている人が研究室にいませんでした。そこで、自分の研究の話を聞いてくれる人が自分の研究を進めるために欲しかったので、まずは進化ゲーム理論の勉強に対して好意的で許容的な学部生であった福冨雅夫氏に進化ゲーム理論の基礎の知識を吹き込みました。そしてその上で、話を聞いてもらっていました。

東京大学の大学院生時代は、研究室のメンバーにも十分にコメントをもらっていましたし、指導教員である井原講師と十分に議論した上で雑誌に論文を投稿していたので、査読者からのコメントがたくさんある時はそれに対してどう改変するかを考えるのが正直億劫であると感じてしまった時もありました。とてもありがたいことのはずなんですが、コメント過剰だと感じてしまっている時も正直なところありました。しかし、京都大学に来てからは、投稿までに自分の論文の内容を聞いてくれるのは福冨雅夫氏だけで、しかも京都大学に来てしばらくたつまではそれもない、というコメント欠乏の状況にうって変わりました。この大きな変化により、2 つの良いことがあったと思います。1 つ目は他の人に自分の研究に対してコメントをしてもら

えることのありがたさに気付けたことです。出版されても誰も読まない可能性は十分にあるわけですが、少なくとも編集者と査読者2人の合計3人は読んでくれて、査読者とのやり取りではとりわけ自分の研究の内容に関して議論してもらえるわけなので感謝しています。査読者への返事を考えている時間が、研究のプロセスの中で一番好きな時間になりました。これは大きな変化でした。2つ目は自分でしっかりと考えて研究を進められるようになったことです。基本的には自分だけで研究を進めることになったため、コメントがくれる人が周囲にたくさんいた東京大学大学院生時代よりも自分の研究をより一層自分自身でしっかりと見るようになったと思います。単著で初めて論文を書いたのも、京都大学時代であり、受理された時は本当に自信になりました。コメント過剰の東京大学の大学院生時代とコメント欠乏の京都大学の博士研究員時代の両方を経験できたことが自分にとっての幸運だと感じています。

京都大学では、自分からわざわざ遠くに行かなくても「生物数学の理論とその応用」が開催されますし、Hanna Kokko など論文生産力がとてもある人も京都大学には来られるので、「時間を失わない」という意味ではとてもありがたい環境でした。研究面で考えたら、今までで最も濃密な時間を過ごせたように思います。また、文系カルチャーの研究室であり、発表の時はスライドのコピーを事前配布したり、文献を引用する際に査読を経ていないものであっても引用したりする人もいるなど、違いを知ることもできました。2016年は第一著者の論文が6本、それ以外の論文が1本出版されました。

8. 中国科学院博士研究員時代

2016年8月から2017年9月までの1年2か月在籍しました。京都大学着任前に、面識はなかった中国科学院動物研究所の Yi Tao 教授にメールを送り応募していましたが、京都大学在籍時にオファーを頂きました。研究室が取り扱っているテーマが自分にとって魅力的であること、海外であることの2点から、Yi Tao 教授のグループに博士研究員として入ることにしました。まず、1点目の「テーマ」ですが、自分が東京大学大学院時代、京都大学博士研究員時代と取り組んでいた協力行動の進化の研究を行っていることに加え、モソ族の婚姻形態など、自分が興味をそそられる研究テーマに取り組んでいることが私には魅力的にうつりました。2点目の「海外であること」ですが、会社員時代に海外の博士研究員のポジションに応募した際、「海外に応募すると海外に住んだ経験はあるか、海外で働いた経験はあるか」を問われるという経験がありました（結果は不採用でした。）。ですので、Yi Tao 教授のグループに入ること自体も魅力的でしたが、中国を離れた後の就職活動で海外に応募した場合にも中国での経歴・経験は有利に働くと考えました。私自身はずっと日本にいてもいいのですが、日本に限定するとポジションを見つけるのが大変だと感じていました。

Yi Tao 教授のグループは、私以外は全員中国人でした。ミーティングは、グループ全体のミーティングとプロジェクトごとのミーティングがありましたが、グループ全体でのミーティングは、中国語で行われていました。スライドがある場合は漢字を見ればだいたい分かるのでまだいいのですが、スライドがない場合は話を見失っていました。自分が発表する際は、英語で発表することを許可してもらっていました。家庭教師を雇うことはしませんでしたが、家庭教師を雇えばよかったと後悔しています。自分がリードしているプロジェクトの個別のミーティングでは共同研究者にも英語で話してもらっていました。

グループのメンバーである Xiudeng Zheng 助教らが書いた協力行動の進化に関する論文（Zheng et al., 2017）を雑誌に投稿する前に読み、コメントするということもしました。協力行動は、自分にとって損である行動なので、協力行動が進化するためにはなんらかのメカニズムが必要ですが、Zheng et al.(2017) は、相手が協力した場合は関係を継続する一方で、相手が非協力をした際は関係をうちきるメカニズムを考えました。この場合、協力者は協力者同士で関係を継続できる一方で、非協力者は協力者と関係を継続できないため、協力者のほうが協力をしてもらいやすい状況が生まれ、協力が進化しえます。Zheng et al.(2017) は、これを数少ないパラメータで表現するとてもシンプルな数理モデルを構築しました。数学的にも綺麗な論文でした。Zheng et al.(2017) との出会いは、それ以降の私の研究にもつながっており (Kurokawa, 2019)、中国科学院に行って得られた最も大きな財産かもしれません。

朝昼晩、研究所の食堂が開いていたこともあり、食事は研究所の食堂で食べることが多かったです。特に、朝限定のメニューである肉マンはおいしく、食べたらおなかが痛くなることは半ば分かっていながら食べていました（そして、おなかが痛くなっていました。）。目玉焼きも朝限定メニューなのですが、これもおいしくてよく食べており、多分中毒になっていました。朝ご飯の時間帯は比較的すいているのですが、昼ご飯の時間帯は、研究所で食べる人が多いためか、混んでいました（図1）。

中国人研究者とちびまる子ちゃんを一緒に見たりするなど、研究以外の交流ができたこともよかったです。住んでいると格段に時間があるので、学会などの限られた時間で過ごすよりも贅沢な時間の使い方ができました。また、研究者以外とも仲良くなりました。中国で知り合った中国人と、カフェでカードゲームをしたり、万里の長城など観光地に行ったり、内モンゴルに

図 1　中国科学院の食堂で昼食を食べる筆者

泊まりで旅行に行ったりして、楽しい週末を過ごすことができました。中国では、英語でコミュニケーションがはかりにくい際は筆談する、ということで意思疎通をはかっていました。共通言語が 2 個あるのは大きいです。これは日本人が持つ大きな利点であると思いました。

中国に住むと困ったこともありました。夏場、職場は大丈夫なのですが、家は 2 日に 1 日くらいのペースで停電になっていました。建物全体での電気の使用量がある閾値を超えると、建物全体が停電になるという仕組みなのですが、夏はクーラーを使う人が多く、電気の使用量が閾値を超えやすいためです。自分だけがクーラーを使わなくても停電になってしまうので大変でした。職場から帰宅した時に建物全体から明かりが消えていたら、それは停電しているということなので、スマホをライト代わりに使い階段をのぼって自分の部屋に行き、自分の部屋に入ってからは、電気を充電しておけるタイプの電気スタンドで、明かりを灯していました。中国科学院は 1 か所にあるわけではなく中国の至る所にある研究所ですが、私がいた動物研究所は北京にあり、私がいた時の冬は例年と比較すると暖冬にも関わらずマイナス 10 度まで冷え込むなど、日本と比べると寒く、そして長いものでした。湖も凍りますが、凍った湖の上でのスケートは初めての経験であり、娯楽の種類も日本とは違うことに気づきました。また、これは季節を問わずですが、お湯も蛇口をひねれば出るものではなく 1 回のシャワーを浴びるためには 50 分くらいは給湯器のようなものをあたためる必要がありました。なので、自宅に帰ったら、給湯器をあたためるという作業を最初にする必要があり、「自宅に帰ってすぐにシャワーを浴びて寝る」ということはできませんでした。浴槽が家にはなかったのですが、私はお風呂に入るのが好きであるため、外のお風呂に月に二回程度は行っていてそれが楽しみでした。

給与は支払われる日付が決まっていなかったので毎月催促していました。日本では、決まった日に何も言わずとも勝手に給与が振り込まれているので、そういったところで時間を失わないという日本の良さに気付くこともできました。中国人は、本当に言葉を使ってコミュニケーションをとる民族であり、せっかく意

思疎通に便利である言葉というものを有しているのにも関わらず、日本人は言葉を使ってコミュニケーションをとることをしないことにも気付かされました。また、中国で働いた経験から、中国でのポジションに関する情報が回ってきやすくなりました。日本に限定するとポジションを見つけるのが難しいので、これも中国で働いて良かったことだと思います。

中国での研究生活および、生活は総じてとても良いものでしたが、高知工科大学の小林豊准教授より高知で働くことのお誘いを受けて、高知に移ることにしました。2017 年は第一著者の論文が 6 本、それ以外の論文が 1 本出版されました。

9. 高知工科大学助教（Research Associate）時代

2017 年 10 月から 2019 年 9 月までのちょうど 2 年間在籍しました。私はそれまで、大阪→東京→埼玉→東京→千葉→大阪→京都→北京と比較的都会ばかりに住んできたので、都会でないところに住むのは初めてでした。新学術領域である「パレオアジア文化史学」のプロジェクトで雇用されていました。ヒトの性質は遺伝子だけではなく、模倣などの非遺伝的手段によって伝達される情報と定義される文化によっても伝達されます。文化の例としては、石器製作の技術などがあげられます。遺伝子は親から子へ、子から孫へ伝えられる一方で、文化は血のつながりがない赤の他人に対しても伝わるので、遺伝モデルよりもより広範な伝達を扱えるモデルが文化進化を扱う上では必要です。東京大学大学院、京都大学、中国科学院動物研究所では、遺伝子によるヒトの行動の進化をメインに取り扱ってきましたが、高知工科大学では、文化進化を取り扱うことになりました。ボスの小林豊准教授とは、雑談から研究の話に至るまで、いろいろな会話をしてもらいました。アカデミアに戻ってきてからこれまで京都大学、中国科学院、高知工科大学、東京大学の 4 か所にいましたが、ボスとの会話がその中で最も多かったのは、高知工科大学でした。2018 年、2019 年の 2 年間で、第一著者の論文が 6 本、それ以外の論文が 1 本出版されました。

10. 東京大学博士研究員時代

2019 年 10 月に、東京大学に博士研究員として着任しました。私は、東京大学に博士研究員として着任するまで、東京大学大学院、京都大学、中国科学院動物研究所、高知工科大学、と一貫して、ヒトの進化に関して調べてきましたが、今は、基盤研究 B「個体群動態に与える個体履歴の影響」で Richard Shefferson 准教授に雇用されて植物の進化について調べています。より具体的には、「光合成も行わず、成長も繁殖もしない休眠という時期を持つ植物が存在し、これは一見不適応に思えるが、適応的なのか、それとも不適応的なのか？そして、不適応ならば、そのような状態を持つに至れたのはなぜか？」という問いに対して理論的に取り組んでいます。一見非適応的な性質を見つけてその説明を試みるという意味では、協力行動の進化の研究と同じであり、問いが問いとして成立していることが簡単に納得できる点が私は嬉しいです。その一方で、休眠の進化に対して取り組む際に有用な道具は、協力行動の進化に対して取り組む際に有用な道具とは異なり、行列モデルを多用します。Hal Caswell の Matrix Population Models というタイトルの本が休眠の進化に取り組む際に有用な行列モデルの技術を身に着ける上で役に立つ本であると感じており、この本を片手に固有値、左固有ベクトル、右固有ベクトルの計算などをしています。数理的な意味でも面白いと感じています。

11. 最後に

先月 2020 年 4 月に私は 36 歳になりました。子供の頃の自分が今の自分を見たら、研究をしていることにはがっかりしないと思いますが、36 歳時点で平々凡々たる結果しか残せていないことにはがっかりすることと思います。そんな自分ですが、この賞を頂けてとても嬉しかったです。応募した際は、受賞するかどうかがいつ決まるのか、また、その知らされ方に関しても知らず、突然受賞のメールを頂いた時は驚き、「賞を頂くということはこんなに嬉しいものなのか」と感じたことを思い出します。会社を辞めた 31 歳の頃、失うものを失って、そして、恐怖の中研究を再開しましたが、アカデミアに戻ってきた後に形があるものを頂けて、戻ってきてよかったなと思わせてくれました。

また、この寄稿文を書く中で振り返り、私と共に研究に携わって下さった方々、名前も分からない査読者の方々のおかげで私の研究は改良されてきたということを改めて確認しました。名前を書き始めると本当にきりがないと思うので書きませんが、お礼申し上げます。単著含め、どの論文も自分だけの力で書き上げたものではないです。

私がしたいことは、「自分が解ける範囲で最も難しく、また、答えが綺麗である問題と、その答えのセットを記述し、他の人がアクセスできる状態にしておく（出版しておく）」ということですが、この賞を頂いたことを励みにして、私がしたいこのことをこれまで以上にやっていき、これから巻き返しをはかっていきたいと思います。

参 考 文 献

[1] 関村利朗, 梯正之, 山村則男, 竹内康博 (2007) 理論生物学入門 現代図書

[2] 酒井聡樹, 近雅博, 高田壮則 (1999) 生き物の進化ゲーム

—進化生態学最前線:生物の不思議を解く—共立出版

[3] Nowak, M. A., 2006. Five rules for the evolution of cooperation. Science, 314, 1560–1563. (doi: 10.1126/science.1133755)

[4] Nowak, M. A., Sasaki, A., Taylor, C., Fudenberg, D., 2004. Emergence of cooperation and evolutionary stability in finite populations. Nature, 428, 646–650. (doi: 10.1038/nature02414)

[5] Kurokawa, S., Ihara, Y., 2009. Emergence of cooperation in public goods games. Proceedings of the Royal Society B, 276, 1379–1384. (doi:10.1098/rspb.2008.1546)

[6] Zheng, X-D., Li, C., Yu, J-R., Wang, S-C., Fan, S-J., Zhang, B-Y., Tao, Y., 2017. A simple rule of direct reciprocity leads to the stable coexistence of cooperation and defection in the Prisoner's Dilemma game. Journal of Theoretical Biology, 420, 12–17. (doi: 10.1016/j.jtbi.2017.02.036)

[7] Kurokawa, S., 2019. Three-player repeated games with an opt-out option. Journal of Theoretical Biology, 480, 13–22. (doi: 10.1016/j.jtbi.2019.07.012)

[8] Caswell, H., 2001. Matrix Population Models, 2nd Edition. Construction, Analysis, and Interpretation. Sinauer Associates, Inc., Sunderland, MA.

The 1927 epidemic model of Kermack and McKendrick : a success story or a tragicomedy?*

Odo Diekmann, Utrecht University

1. How it started.

At the end of the 19th century, it was suspected that malaria was not caused by bad air, as the Italian name suggests, but by a parasite. It was, however, a mystery how the parasite was transmitted. In 1897 Ronald Ross resolved this mystery by establishing that mosquitos acted as a vector. Ross received the 1902 Nobel prize for physiology/medicine for this discovery and in 1911 he was knighted.

Ross had many talents (he wrote poems and novels, composed music, made drawings, ...). He also was a keen mathematician and after his discovery he used mathematical models to show that malaria could be eliminated by bringing the ratio of mosquito - to human density below a certain threshold value.

In 1901, Ross went on a mission to Sierra Leone to improve the local anti-malarial measures. Among the other people taking part in the mission, was the 25-year-old Anderson Gray McKendrick, who, after completing medical studies at the University of Glasgow, had entered the Indian Medical Service and was trained in military and tropical medicine. It seems safe to conclude that Ross instilled into the young McKendrick a sincere interest in both infectious diseases and mathematics (the former being natural and to be expected, the latter being catalysed by the coincidence that McKendrick possessed a probably even greater talent for mathematics than Ross himself).

After the Sierra Leone mission, McKendrick served for almost two decades in India, where he investigated rabies in particular (the treatment of patients and the preparation of vaccine being an important part of these investigations). In 1920 he returned to Scotland for health reasons and then he started to work at the Royal College of Physicians in Edinburgh. In 1921, the 23-year-old (bio)chemist William Ogilvy Kermack joined the college. In 1924, Kermack (who had safely completed service in the Royal Airforce during the First World War) lost the sight in both of his eyes in a lab explosion. It is not known to the writer of this story when and how the collaboration between McKendrick and Kermack started, but in 1927 the Proceedings of the Royal Society A contained the joint paper "A contribution to the mathematical theory of epidemics" [1]. A truly brilliant paper!

2. The model.

The incidence of an infectious disease is defined as the number of new cases per unit of time (e.g., per day) in a specified population or geographical area (or per unit of area). The force of infection on a susceptible individual is defined as the probability per unit of time of the susceptible becoming infected. A key assumption of the Kermack-McKendrick model states

$$\text{incidence} = \text{force of infection} \times \text{size of the susceptible subpopulation}$$

(this is a variant of the Principle of Mass Action of chemical kinetics, see [2]; often 'size' is measured in terms of spatial density, but alternatively one can consider the number of inhabitants of a certain geographical area, for instance a country; the main requirement is that 'incidence' and 'susceptible subpopulation' use the same measuring stick).

The next step is to elaborate the idea that the current force of infection is generated by individuals that were themselves infected some time ago (note that now we focus on direct host-to-host transmission rather than on vector transmission). So we want to relate the current force of infection to the incidence in the past. Here, following earlier work by Ross and Hudson (see [2,3], also for the precise references), [1] adopts a top down approach and introduces as the one-and-only model ingredient a description of

*This lecture was presented at the meeting of the Finnish Society of Science and Letters held in the House of Sciences, Kirkkokatu 6, Helsinki, October 21, 2019.

the expected contribution to the force of infection at time τ after infection. Here τ varies from zero to infinity, but common sense suggests that the contribution vanishes for both small and large values of τ. For any specific disease one may, in principle, use a within-host model of the struggle between parasite and immune system to provide a quantitative version of this description bottom up. The aim of the Kermack-McKendrick paper [1], however, is to derive the qualitative consequences of the assumption, to develop a general theoretical framework as a backbone for studies of specific diseases.

The next assumption is that the population is demographically closed, meaning that transmission of the infectious disease happens on a fast time scale relative to the time scale of demographic turnover, i.e., the time scale of birth, death and migration of host individuals (think of an influenza epidemic hitting a human population in, say, a period of half a year, while humans have a life expectancy of, say, 80 years).

The final assumption is that infection ends with complete immunity, i.e., a host individual cannot be infected a second time.

As a consequence of the last two assumptions the susceptible subpopulation only changes due to the infections that happen. More precisely: the rate at which the size of the susceptible subpopulation diminishes exactly equals the incidence.

As shown in [1,4,5] one can derive, from these assumptions, a scalar nonlinear renewal equation [6] for the force of infection as a function of time. An analysis of this equation then yields the following epidemiological insights:

i) if a newly infected individual is introduced in an otherwise susceptible population, the expected number of secondary cases, called the Basic Reproduction Number and denoted by the symbol R_0 (pronounce as "R nought"), equals the product of the total population size and the expected cumulative (i.e., integrated over all tau) contribution to the force of infection; if $R_0 < 1$ the introduction does NOT lead to an epidemic, but if $R_0 > 1$ it may very well lead to an epidemic outbreak and, if it does, a fraction $1-x$ of the host population will become infected, with x the positive solution of the equation

$$x = \exp(-R_0(1-x))$$

ii) In the initial phase of an outbreak the incidence grows exponentially with rate r (the so-called Malthusian parameter) implicitly defined as the positive root of a relatively simple (characteristic) equation. Note that data about a newly emerging infectious disease yield estimates of r while estimating the control effort needed to stop the outbreak requires information about R_0 (control reduces R_0 with a certain factor and the aim is to bring R_0 to a value below one). So it is of paramount importance to understand how R_0 and r relate to each other and here the dependence of the contribution to the force of infection on the time tau elapsed since infection plays a crucial role.

Once again, we refer to [1,4,5] for a detailed exposition.

3. A very special case of the model.

If one assumes that a newly infected individual 'enters' the compartment of infectious individuals and
– provides a constant contribution to the force of infection while being in this compartment
– does stop contributing when leaving the compartment
– leaves the compartment with a constant probability per unit of time (or, in other words, has an exponential sojourn time)
one can substitute the renewal equation by an equivalent system of ordinary differential equations (ODE), called the S-I-R compartmental system (with S standing for 'susceptible', I for 'infectious' and R for 'removed'). This S-I-R system is nowadays the prototype epidemic model, figuring in countless textbooks.

4. Success (somewhat overshadowed by misconception).

On September 15 2020, [1] had 7940 citations on Google Scholar and the website of the Royal Society mentioned that the paper was downloaded 38194 times since first published online. These figures suggest that the paper is immensely successful. Yet there are reasons to doubt that it is ever read! Indeed, an incessant community-enforced misconception is that the paper is just about the very special case, the S-I-R model. A comical consequence is that time and time again manuscripts are submitted, and often even published, about a "generalization" that is in fact only a slightly less special case of the true Kermack-McKendrick model.

One may wonder why this misconception is so persistent. No doubt the tendency to parrot plays an important role. But another factor is, very likely, the fact that ODE are the standard modeling formalism for most applied mathematicians and that a wealth of theory and a multitude of tools exist for ODE, while, in contrast, renewal equations occupy an obscure infinite-dimensional corner of the world of dynamical systems (but see [7] for recent developments). It is tempting to deny the unfamiliar ...

Anyhow, the 1927 Kermack-McKendrick paper is, in my opinion, definitely a success story, but with a tragicomical lining because the true gem remains mostly unnoticed, despite the spotlights.

5. About motivation and attitude.

When authors of 'generalization' manuscripts are told about the real contents of [1], they tend to be irritated. Quite recently, the information even triggered a somewhat aggressive reaction: you are whining about a paper from 1927, that is prehistory! The aim of our paper is to show that we are smarter than all those other young scientists that compete for a tenure track position and only recent papers matter for us. (Admittedly this is a rather free and subjective translation of the literal text, but I claim that it captures the essence.)

Now contrast this with the picture of Anderson Gray McKendrick that W.M. Hirsch sketches in [8]. It is clear that McKendrick's motivation was to fight infectious diseases while building on general insights derived in the context of mathematical models. His goal was to develop knowledge and understanding and, ultimately, control. As far as one can guess, he was not thinking in terms of competition. The quote of Hirsch

"McKendrick was a truly Christian gentleman, a tall and handsome man, brilliant in mind, kind and modest in person, a skillful counsellor and administrator who gave of himself and knew how to enable others."

in fact suggests the opposite!

So the most tragical aspect of the Kermack-McKendrick story is probably not that, despite the many citations, the paper is hardly ever read, but that the story illustrates how we developed a system of evaluation of scientific research that drives young talents to publish quickly and profusely without much attention for meaning or for 'old' papers in which tremendous wisdom lies hidden.

Acknowledgement.

Suggestions by Hans Heesterbeek and KaYin Leung are gratefully acknowledged.

References.

[1] Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London.* Series A, Containing papers of a mathematical and physical character, 115 (772): 700–721.

[2] Heesterbeek, J. A. P. (2005). The law of mass-action in epidemiology: a historical perspective. In K. Cuddington & B. Beisner (Eds.), *Ecological Paradigms Lost: Routes of Theory Change*, pp. 81–105. San Diego, USA, Academic Press.

[3] Heesterbeek, J. A. P. (2002). A brief history of R_0 and a recipe for its calculation. *Acta Biotheoretica*, 50 (3):189–204.

[4] Breda, D., Diekmann, O., de Graaf, W. F., Pugliese, A. & Vermiglio, R. (2012): On the formulation of epidemic models (an appraisal of Kermack and McKendrick), *Journal of Biological Dynamics*, 6, sup2: 103–117.

[5] Diekmann, O., Heesterbeek, J. A. P., Britton, T. (2012) *Mathematical Tools for Understanding Infectious Disease Dynamics*, Princeton University Press.

[6] Gripenberg, G., Londen, S. O., Staffans, O. (1990) *Volterra Integral and Functional Equations.* No. 34 in Encyclopedia Math. Appl., Cambridge University Press, Cambridge.

[7] Breda, D., Diekmann, O., Liessi, D., Scarabel, F. (2016) Numerical bifurcation analysis of a class of nonlinear renewal equations. *Electron. J. Qual. Theory Differ. Equ.* 65: 1–24.
DOI 10.14232/ejqtde.2016.1.65

[8] http://mathshistory.st-andrews.ac.uk/DNB/McKendrick.html (accessed on September 15, 2020)


Odo Diekmann

編集後記

本稿は昨年（2019 年）10 月にフィンランド科学アカデミーで行われた Odo Diekmann（ユトレヒト大名誉教授）による講演原稿である。ただし一部データは今年 9 月の時点でアップデートされている。今回，Diekmann 教授の好意によって，ここに載録させていただくことになった。記して感謝したい。Odo

Diekmann[1)]は，Journal of Mathematical Biology の名誉編集委員でもあり，長年，欧州の応用数学，数理生物学界のリーダーとして活躍していることは紹介するまでもないであろう。主要な著作として，文中のリファレンス [5] 以外に，

– The Dynamics of Physiologically Structured Populations (with Hans Metz), 1986
– Delay Equations: Functional-, Complex- and Nonlinear Analysis (with Stephan van Gils, Sjoerd Verduyn Lunel and Hans-Otto Walther), 1995
等がある。

感染症数理モデルの歴史に関して，少しだけ背景を説明しておきたい。Kermack and McKendrick の 1927 年の論文は，近代的な感染症数理モデルの誕生を告げる画期的な論文であり，今日まで繰り返し参照されてきている。しかしながら，その数学的な含意が十分に明らかになったのは半世紀後，1970 年代末以降に Diekmann, Metz, Thieme, Gripenberg, Lauwerier などの主としてオランダ，ドイツの数学者，数理生物学者による現代的な定式化と数理解析がおこなわれるようになってからである。しかしその後も，Kermack and McKendrick の原論文は「引用されども読まれず」という状態が続いている。それがシリーズペーパーの第一報であることも忘れられている。Diekmann が指摘するような再生方程式による本質的な定式化や，感染齢構造化モデルは，数理科学界においても今日まで広まらず，単純な常微分方程式による SIR モデルのみが人口に膾炙し，オウム返しのように語られている。それゆえ，いまでも「Kermack and McKendrick モデルの一般化」と称して，その実，本来の Kermack and McKendrick モデルの特殊な形態にすぎないモデルを提唱する論文があとをたたない。

読まずに引用するのは笑い話かもしれないが，歴史と伝統へのリスペクトを欠いている。しかし，それにもまして Diekmann が警告するのは，そのような過去の論文，過去の知恵を軽視する態度が，ひたすら「新しい成果」を追い求める若い研究者の間で蔓延していて，学問の本質を見失わせているのではないか，ということであろう。それはいうまでもなく，若い研究者がアカデミイのなかで地位を得るための競争がますます激しくなっていて，同輩の中でぬきんでるために，いかに競争的に優位な「新奇な」結果を示すか，という相対的，競争的観点が支配して，本来の探求の動機と歴史が見失われていることによる。これは若者の責任でもなく，老人の繰り言でもなく，システムの堕落ともいうべきではないか。Diekmann が引用する Hirsch が描く McKendrick のありようはこれとは対極に位置する。

今回の COVID-19 においては，膨大な感染症数理モデル論文が日々出版されているが，その著者の多くは数理生物学や数理疫学の研究者ではなく，これまでの感染症数理モデルの伝統と成果を全く知らない人々である。そこにおいて日々演じられる「車輪の再発明」に嘆息した人も少なくないであろう。しかし一方で，現下の感染症流行をなんとかして防ぎたいという真摯な動機は疑うべくもなく，先人の知恵に学ぶ千載一遇の好機でもある。本講演は，いまだ SARS-CoV-2 の陰も見えない時期におこなわれたが，その頃から中国では新型コロナウィルスによる感染が密かに始まっていたと考えられる。図らずもこれ以上ないタイミングにおいて発せられた Diekmann の警告であったと言うべきであろう。(稲葉 寿，東京大学)

[1)] https://www.uu.nl/medewerkers/ODiekmann

【第 2 回 数理生物研究 × 計算機】

力学系の数理解析と数値計算

一ノ瀬元喜 *，宮川大樹 †

1. はじめに

第 2 回目は，力学系の数理解析と数値計算についてやってみましょう．数理解析の部分は数理生物学を研究されている方なら，自分の手で計算されることが多いのでしょうが，ここは Python をフル活用してやっていきたいと思います．

今回の内容を図 1 に示します．まず，Python で代数計算を行うためのライブラリである SymPy の紹介をします．SymPy で解く力学系の題材としては，第 1 回でも用いたロトカ–ヴォルテラの捕食者–被食者モデルを再び取り上げます．この方程式は解析的に解けないので，数理生物の方にはお馴染みの線形安定性解析をしていきます．そのために，まず平衡点を求めます．次にヤコビ行列を計算し，2 つの平衡点についての固有値を求めます．さらに平衡点まわりの解の挙動をベクトル図で可視化します．最後におまけで個体群動態のアニメーションもつけたいと思います．では，以上のことを紙とペンではなく，Python を使って早速計算機でやっていきましょう！

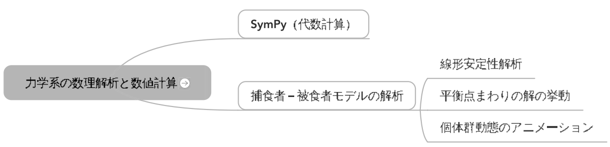

図 1 今回の内容

今回用いる全てのプログラムのソースコードは，前回紹介したこのニュースレター用の GitHub（https://github.com/igenki/biomathNL）[1] に置いてあります．ファイルのダウンロードは前回説明しましたが，もう一度簡単に書いておきます．このページの右上部分に「Clone or download」と書かれた緑色のボタンがあります．それをクリックして出てきたポップアップから「Download ZIP」をクリックしてください．そうするとリポジトリの内容がすべて zip 形式の biomathNL-master.zip という名前で一括でダウンロードできます．ダウンロードが完了したら，まずそれを解凍（展開）して，中身[1)]を前回 Desktop 上に作った biomathPython という名前のフォルダに移動させましょう．あとは以下の説明を読みながら，それぞれの該当プログラムを Jupyter で開いて実行していくとよいかと思います．

2. SymPy

SymPy[2] は，Python の代数計算ライブラリ[2)]です．通常のプログラムは，数の操作を行うことが多いですが，SymPy を使うと記号の操作（文字式の計算）を行うことができます．つまり x という記号を記号として扱った計算を行えるということです．記号計算，数式処理とも呼ばれます．このような代数計算（あるいは記号計算，数式処理）ができるソフトウェアとしては Mathematica，Maple，Maxima などが知られています．Mathematica は有料なので強力ですが，Python の SymPy でも十分な代数計算が可能ですので，今回はこれを用いて計算を行っていきます．

SymPy の使い方を説明します．Jupyter で SymPy.ipynb ファイルを開いてください．図 2 のコードが見れると思います．このセルを Shift+Enter で実行しましょう．最初の行で sp という名前で SymPy ライブラリを import して使えるようにします．次に x という記号を使いたいので，Symbol() 関数を使って x を使用可能にします．3 行目の print() 関数によりこの式の評価結果である 3*x + 1（ちゃんと記号で計算できていますね！）が出てくると思いますが，ご覧の通り見にくいです．よって，display() 関数を使って $3x+1$ と見やすい形

*静岡大学学術院工学領域
†静岡大学大学院

1) 解凍すると 01 と 02 というフォルダが出てくると思います．今回使うのは 02 のフォルダに入っているファイルです．01 は第 1 回で使用したファイルです．

2) Python で「モジュール」「パッケージ」「ライブラリ」という言葉が出てきます．○○モジュールを import する，と言った具合です．この 3 つの言葉はあまり区別なく，曖昧に使われることも多いですが,「モジュール」とは,（クラスや関数の定義が書かれている）Python のファイル（.py）のことで，一番小さい単位です．そのモジュールを複数集めたものが「パッケージ」，いくつかのパッケージのまとまりが「ライブラリ」と考えてもらうとよいでしょう．

（LaTeX 形式）にして出力します[3)4)]．

SymPy入門

```
In [1]: import sympy as sp
        x = sp.Symbol('x')
        print(x+x+x+1)
        # display()関数で見やすい形になる
        display(x+x+x+1)

3*x + 1
```

$3x+1$

図 2　SymPy 最初の一歩

次に x だけではなく，複数の記号の使い方について少しやってみましょう．次のセルは図3のようになっているはずですので，実行してみてください．2つのやり方を書いていますが，どちらでもよいです．上の方の出力結果は $x(x+y+z)$，下の方の出力結果は $a^2(b+c)$ が出ているのが分かります．

```
In [2]: # 複数記号の導入
        x, y, z = sp.symbols('x,y,z')
        expr = x*(x+y+z)
        display(expr)

        # 複数記号の導入（別バージョン）
        sp.var('a,b,c')
        expr = a*(b+c)*a
        display(expr)
```

$x(x+y+z)$

$a^2(b+c)$

図 3　複数記号の使用

次は因数分解をSymPyにやってもらいましょう（図4）．$x^3+3x^2y+3xy^2+y^3$ を factor() 関数に渡すとできます．$(x+y)^3$ という結果が出力されましたか？

次は微分です（図5）．diff() 関数の第1引数に微分させる式，第2引数に対象の変数を指定します．x^2 が x で微分されて $2x$，$\sin x$ が x で微分されて $\cos x$ と出力されたと思います．

最後に連立方程式を解いてもらいましょう（図6）．solve() 関数を使って，

```
In [3]: # 因数分解
        sp.var('x,y,z')
        expr = x**3 + 3*x**2*y + 3*x*y**2 + y**3
        factors = sp.factor(expr)
        display(factors)
```

$(x+y)^3$

図 4　因数分解

```
In [4]: # 微分
        x = sp.Symbol('x')
        display(sp.diff(x**2, x))
        display(sp.diff(sp.sin(x), x))
```

$2x$

$\cos(x)$

図 5　微分

$$\begin{cases} 2x+3y=6 \\ 3x+2y=12 \end{cases}$$

という連立方程式を解いてもらっています．6と12を左辺に移行して $=0$ としてから solve() 関数に渡していることに注意してください．解いた結果が，{x: 24/5, y: -6/5} のように出力されると思います．この形は Python では**辞書**と呼ばれるデータ構造です．後でも出てくるので辞書について簡単に説明しておきます．辞書とは**キー**と**値**がセット（ペア）になったデータ構造です．上の例でいくと x というキーに対応するのが 24/5 という値，y というキーに対応するのが-6/5 という値と見ます．

```
In [5]: # 連立方程式を解く
        sp.var('x,y')
        expr1 = 2*x + 3*y -6
        expr2 = 3*x + 2*y -12
        display(sp.solve((expr1, expr2)))

        {x: 24/5, y: -6/5}
```

図 6　連立方程式

以上で，今回必要な SymPy の機能の説明は終わりました．次の節からいよいよ本番です．

3.　ロトカ–ヴォルテラの捕食者–被食者モデルの解析

3.1　線形安定性解析

第1回に引き続き，以下のロトカ–ヴォルテラの捕食者–被食者モデルを扱います．

$$\begin{aligned} \frac{dx}{dt} &= ax - bxy, \\ \frac{dy}{dt} &= cxy - dy. \end{aligned} \tag{1}$$

ここで，$a,b,c,d>0$ です．この方程式は解析的に解

[3)] display() 関数は IPython.core.display モジュールに含まれますが，Jupyter Notebook を起動すると自動的にインポートされるので，display() という記述だけで使えます．

[4)] 最初に init_printing() 関数を一度記述しておけば，その都度 display() 関数を使わずに，セルの最後の数式を LaTeX 形式で表示してくれます．今回の記事では途中の式も明示的に出力させている場合があるので display() 関数をその都度使っています．また init_printing() 関数は Anaconda のバージョンによっては動かない場合もありますので，注意してください．

けないので，線形安定性解析をして，その挙動について理解を深めます．今度は Jupyter で LV_analysis.ipynb を開いてください．プログラムでは，まず必要ライブラリを import してから，用いる変数と方程式を定義します（図 7）．以下を実行すると f と g に入っている $ax-bxy$ と $cxy-dy$ が出力されると思います．

変数宣言 & 方程式の定義

```
In [1]: import sympy as sp
        import numpy as np
        import matplotlib.pyplot as plt

        # 変数, 定数の宣言
        sp.var('x,y,a,b,c,d')

        # 関数f,gの定義
        f = a*x - b*x*y
        g = c*x*y - d*y

        display(f)
        display(g)
```

$ax - bxy$

$cxy - dy$

図 7　変数の宣言と定義

次に上の $f=0$ と $g=0$ の連立方程式を solve() 関数に渡して，平衡点を求めます．図 8 のコードにより，2 つの平衡点 $(0,0)$ と $(d/c, a/b)$ を求め，equivs という名前の変数に格納し，最後に equivs を出力しています．

$f = 0, g = 0$ の連立方程式を解いて平衡点を求める

```
In [2]: eq1 = sp.Eq(f, 0)
        eq2 = sp.Eq(g, 0)
        equivs = sp.solve([eq1, eq2], [x,y])
        display(equivs)
```

```
[(0, 0), (d/c, a/b)]
```

図 8　平衡点を求める

これらの平衡点から少しずれた点の時間発展を調べます．そのずれは線形近似できるほど小さいとします．まず系を線形化したヤコビ行列を求めます．図 9 のように各要素を x または y で偏微分して求めます[1)]．次に要素をまとめて行列形式で表します．

ヤコビ行列が求まったので，1 つ目の平衡点である $x=0, y=0$ での固有値を求めていきましょう（図 10）．まずヤコビ行列の各要素に $x=0, y=0$ を代入したいです．そのために最初の行で (x, y) という行列（行ベクトル）を作ります．次の行が少しややこしいですが，zip() 関数を用いて複数のリストから要素を取り出し，まとめます．いま u の中身は (x, y) です．一方，equivs[0] というのは図 8 で求めた 1 つ目の平衡点 $(0,0)$ を指します．zip() 関数はこれらの 1 つ目同士，2 つ目

ヤコビ行列を求める

```
In [3]: # ヤコビ行列の各要素を求める
        j11 = sp.diff(f, x)
        j12 = sp.diff(f, y)
        j21 = sp.diff(g, x)
        j22 = sp.diff(g, y)

        # 要素を並べて行列にする
        J = sp.Matrix(([j11,j12],[j21,j22]))
        display(J)
```

$$\begin{bmatrix} a - by & -bx \\ cy & cx - d \end{bmatrix}$$

図 9　ヤコビ行列を求める

同士をまとめます．さらにそれを list() 関数でリスト化して xy_equiv1 という名前の変数に入れています．よって，xy_equiv1 を出力すると図 10 のように [(x, 0), (y, 0)] が出てきます．これで $x=0, y=0$ をヤコビ行列 J に代入する準備ができました．subs() 関数を使うと変数に値を代入することができるので，J の subs() 関数に xy_equiv1 を引数として渡せば，[(x, 0), (y, 0)] が代入された結果が返ってきて，それを display() 関数で出力しています．eigenvals() 関数で固有値を求めることができます．平衡点 $(0,0)$ を代入したヤコビ行列の固有値は eigenvals() 関数で求めると {a: 1, -d: 1} という答えが出てきます．この出力形式は，図 5 で出てきた辞書形式なのですが，このキーが固有値，値が重複度に対応しますので，固有値は a と $-d$ ということが分かります．よって，正の固有値があるので，平衡点 $(0,0)$ は不安定となります．より正確には，1 つの正の固有値と 1 つの負の固有値をもつので，平衡点 $(0,0)$ は鞍点です．

```
In [5]: u = sp.Matrix((x,y))

        # 平衡点 (0,0) についての固有値
        xy_equiv1 = list(zip(u,equivs[0]))
        print(xy_equiv1)
        display(J.subs(xy_equiv1))
        ev1 = J.subs(xy_equiv1).eigenvals()
        display(ev1)
```

```
[(x, 0), (y, 0)]
```

$$\begin{bmatrix} a & 0 \\ 0 & -d \end{bmatrix}$$

```
{a: 1, -d: 1}
```

図 10　平衡点 $(0,0)$ についての固有値

同様に，2 つ目の平衡点である $x=d/c, y=a/b$ での固有値を求めたのが図 11 です．出力結果が {-sqrt(-a*d): 1, sqrt(-a*d): 1} と出ているので，固有値は $\pm i\sqrt{ad}$ であることが分かります．固有値が純虚数であり，実部は 0 なので，（線形近似の範疇においては）平衡点 $(d/c, a/b)$ は中立安定となります．

---

1) よりスマートなヤコビ行列の計算方法も GitHub においたプログラムにはコメントアウトして載せています．

```
In [6]: # 平衡点 (d/c, a/b) についての固有値
        xy_equiv2 = list(zip(u,equivs[1]))
        print(xy_equiv2)
        display(J.subs(xy_equiv2))
        ev2 = J.subs(xy_equiv2).eigenvals()
        display(ev2)

        [(x, d/c), (y, a/b)]
```

$$\begin{bmatrix} 0 & -\frac{bd}{c} \\ \frac{ac}{b} & 0 \end{bmatrix}$$

```
{-sqrt(-a*d): 1, sqrt(-a*d): 1}
```

図 11　平衡点 $(d/c, a/b)$ についての固有値

3.2　平衡点まわりの解の挙動

平衡点まわりの解の挙動をベクトル場を用いて可視化してみましょう．まずは平衡点 $(0, 0)$ からです．

ここからは図だと長くなりますので，ソースコードの形式で見ていきます．ソースコード 1 をご覧ください．まず，パラメータ $a, b, c, d > 0$ の値を設定します．7 行目で equivs[0] に入っている (0,0) をそれぞれ xo, yo という変数に入れます．次に LX，LY に 0.05 を代入し，平衡点から ± 0.05 の範囲を表示範囲に設定し，ベクトル描画の刻み幅 gridwidth を 0.01 に設定します．

11-13 行目は格子点の生成です．ちょっとややこしいですが，11 行目の arange() 関数から見ていきましょう．NumPy の arange（）関数は指定された区間内で等間隔の値を持った配列を返します．関数中の最初の 3 つの引数は,「開始値，終了値，間隔」を示します．よって，ここでは，np.arange(-LX+xo, LX+xo+0.001, gridwidth, dtype=float) によって，-0.05 から $+0.05$ までの 0.01 刻みの値の配列を生成しています[1]．y の値についても同様に np.arange($-$LY+yo, LY+yo+0.001, gridwidth, dtype=float) によって，-0.05 から $+0.05$ までの 0.01 刻みの値の配列を生成しています．それらが，13 行目で np.meshgrid(x_array, y_array) の様に meshgrid() 関数に渡されていることが分かります．この meshgrid() 関数によって格子点を生成してくれます．具体的には，いま x_array と y_array の配列は同じように $[-0.05, -0.04, ..., 0, ..., +0.04, +0.05]$ となっているのですが，これらを meshgrid() 関数に渡すことで座標を表す格子点を生成してくれます．簡単な例で説明しましょう．np.meshgrid([0, 1], [0, 1]) であれば，[0, 0], [0, 1], [1, 0], [1, 1] の 2×2 の 4 つの格子点を生成してくれます．

16, 17 行目でベクトルの成分を計算しています．U がベクトルの x 成分，V がベクトルの y 成分です．これらを方程式に従って計算します．U も V も配列となることに注意してください．

最後に quiver() 関数を使って，ベクトルを描画させています．前から 4 つの引数は順番にベクトルの始点の x 座標，y 座標，x 成分，y 成分となります．

```
1  # 平衡点 (0, 0) まわりの解の挙動
2  a_ = 1
3  b_ = 0.01
4  c_ = 0.02
5  d_ = 1
6
7  xo,yo = equivs[0] # 平衡点
8  LX, LY = 0.05, 0.05 # 平衡点まわりの表示範囲（今
       回は± 0.05の範囲を表示）
9  gridwidth = 0.01 # ベクトルを描画する間隔
10
11 x_array = np.arange(−LX+xo, LX+xo+0.001,
       gridwidth, dtype=float)
12 y_array = np.arange(−LY+yo, LY+yo+0.001,
       gridwidth, dtype=float)
13 X, Y= np.meshgrid(x_array, y_array)
14
15 # 解の挙動を算出
16 U = a_*X − b_*X*Y
17 V = c_*X*Y − d_*Y
18
19 # プロット
20 plt.figure(figsize=(4,4))
21 plt.quiver(X,Y,U,V,angles='xy',scale_units='xy',
       scale=5.0)
22 plt.savefig('vector1.eps')
```

ソースコード 1　平衡点 $(0, 0)$ まわりの解の挙動

このソースコードを実行すると，図 12 が Jupyter 上に表示されたと思います．このベクトル図からも (0,0) が鞍点になっていることが分かります．

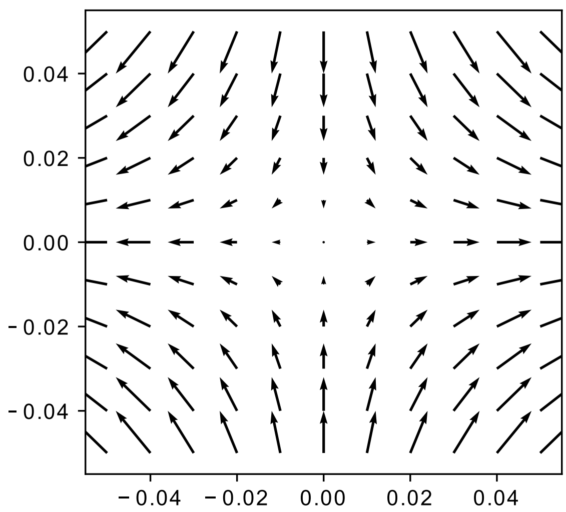

図 12　平衡点 $(0, 0)$ まわりの解の挙動

---

[1] 終了値を「LX+xo+0.001」と設定しています．LX+xo，つまり 0.05 で良さそうなのですが，arange() は正確には終了値未満の値で終了してしまうので，+0.001 として終了値を 0.051 に設定しています．

次に2つ目の平衡点 $(d/c, a/b)$ のベクトル場を描画してみましょう．やることは (0,0) の場合と同じです．こちらの平衡点は a, b, c, d の値に依存するので，その計算をする箇所である 8, 9 行目だけ異なります（ソースコード2）．

```
1  # 平衡点 (d/c, a/b) まわりの解の挙動
2  a_ = 1
3  b_ = 0.01
4  c_ = 0.02
5  d_ = 1
6
7  xo,yo = equivs[1] # 平衡点
8  xo = xo.subs(list(zip([c,d], [c_,d_]))) # パラメータの値を代入
9  yo = yo.subs(list(zip([a,b], [a_,b_]))) # 同上
10
11 LX, LY = 0.05, 0.05 # 平衡点まわりの表示範囲(今回は± 0.05の範囲を表示)
12 gridwidth = 0.01 # ベクトルを描画する間隔
13
14 x_array = np.arange(−LX+xo, LX+xo+0.001,
       gridwidth, dtype=float)
15 y_array = np.arange(−LY+yo, LY+yo+0.001,
       gridwidth, dtype=float)
16 X, Y= np.meshgrid(x_array,y_array)
17
18 # 解の挙動を算出
19 U = a_∗X − b_∗X∗Y
20 V = c_∗X∗Y − d_∗Y
21
22 # プロット
23 plt.figure(figsize=(4,4))
24 plt.quiver(X,Y,U,V,angles='xy',scale_units='xy',
       scale=5.0)
25 plt.savefig('vector2.eps')
```

ソースコード 2　平衡点 $(d/c, a/b)$ まわりの解の挙動

実行すると図13が表示されたと思います．平衡点のまわりの軌道は平衡点に近づきも離れもしない中立安定であることが図からも確認できました．

3.3　個体群動態のアニメーション

前回の第1回では，捕食者－被食者の個体群動態をグラフで可視化しました．今回はそれをアニメーションで可視化してみたいと思います．

アニメーションの原理は簡単で，**リストに入れた画像データをパラパラ漫画のように順に表示**するだけです．図解したものを図14に用意しました．またソースコード3はアニメーションのプログラムのテンプレートを示しています（これは単なるテンプレートなので動きません）．図とテンプレートを照らし合わせながら以下の説明を読んでください．

ソースコード3の2行目から説明します．%はJupyterのマジックコマンドの利用を指します．マジックコマンドは，様々な処理を Jupyter 上で実行可能にしてくれます．matplotlib nbagg は，Jupyter 上でアニメーションを表示させる機能を提供します．4，5行目で画像とアニメーションに必要な Matplotlib ライブラリを import します．7-9 行目では，「必要な処理を書く」と書いていますが，ここには例えば「x のデータを生成する」のような処理が来ます．このプログラムはテンプレートなので何も書いていません．12 行目は図14を参照しながらご覧ください．fig という名前で描画する領域を確保しています．15 行目は画像データをストックするリストを ims という名前で作成します．最初は空です．17 行目から画像を生成していく処理です．18-20 行目の「必要な処理」には（x のデータを基に）y のデータを生成するといった処理が入ります．折れ線グラフを描く場合は 23 行目で画像を生成して im に入れています．括弧の中の hoge[1]には，実際には，x と y のデータを入れます．折れ線グラフではなく，散布図にしたい場合は 26 行目，2 次元画像データにした

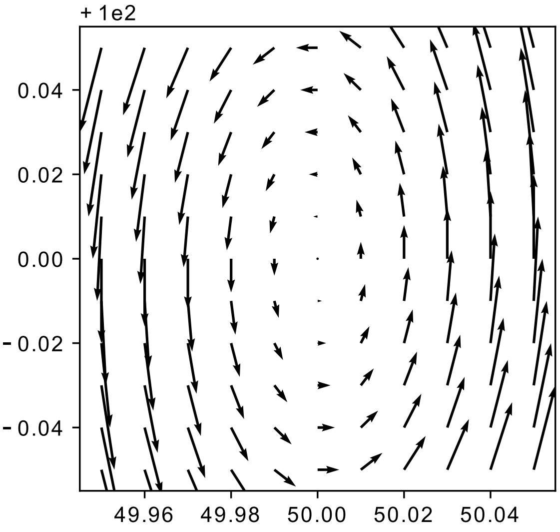

図 13　平衡点 $(d/c, a/b)$ まわりの解の挙動

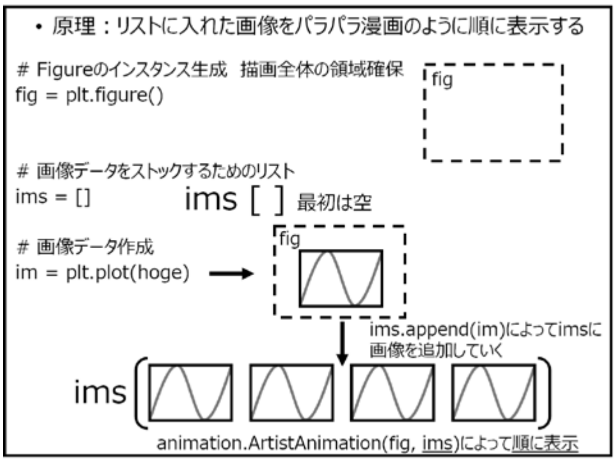

図 14　アニメーションの仕組み

[1] hoge とはサンプルのコードの変数名として使われたりする文字列で，特に意味はありません．

い場合は29行目のコメントアウトを外して，23行目をコメントアウトします．32行目で画像データを先ほど作ったimsというリストに追加していきます[2)]．35行目でArtistAnimation()関数を使って[3)]，imsに入っている画像データを順に表示していきます．

```
1  # Jupyter 上でアニメーション実行のため必要
2  %matplotlib nbagg
3
4  import matplotlib.pyplot as plt
5  import matplotlib.animation as animation
6
7  '''
8  必要な処理を書く
9  '''
10
11 # Figure のインスタンス生成
12 fig=plt.figure()
13
14 # 画像データをストックするためのリスト
15 ims=[]
16
17 for i in range(MAXT+1):
18     '''
19     必要な処理を書く
20     '''
21     # 画像データを生成
22     # 折れ線グラフ
23     im=plt.plot(hoge)
24
25     # 散布図
26     # im=plt.scatter(hoge)
27
28     # 2次元画像データ
29     # im=plt.imshow(hoge)
30
31     # 画像データをリストに追加
32     ims.append(im)
33
34
35 ani=animation.ArtistAnimation(fig, ims)
36 plt.show()
```

ソースコード3　アニメーションプログラムのテンプレート

Pythonによるアニメーションの仕組みを理解したところで式(1)のロトカ–ヴォルテラ方程式の個体群動態をアニメーションしてみましょう．JupyterでLV_animation.ipynbを開いてください．同じものをソースコード4に載せています．これを見ると，第1回で個体群動態のグラフを作った時のコードと上記のテンプレートを組み合わせたようなコードになっていることが分かると思います．上から簡単に解説していきます．1-9行目まではテンプレートと同様です．11-15行目でa, b, c, dのパラメータ値を設定します．18，19行目は被食者と捕食者の個体数（xとy）の初期値を設定，21-26行目はオイラー法に関する設定です．34，35行目にあるようにx_listとy_listで時間ごとのそれぞれの個体数を管理します．時刻$t=0$の画像をアニメーションでの最初の画像としたいので，xとyについての時刻$t=0$での画像を38，39行目でそれぞれ生成します．40行目で画像を結合（im_x+im_y）し，1枚の画像データにしてからimsのリストに追加します．

42行目からがメインの処理でfor文によって時間を進めていきます．44，45行目で次時刻のxとyの値を計算します．画像データは10回に1回生成するため，51行目のif文があります．58-60行目は38，39行目の処理と同じで$t=0$から現在までの画像を生成して，xとyの画像をまず生成してからそれを1つの画像にしてリストに追加（61行目）とします．64行目はテンプレートと同じアニメーションの作成処理です．intervalオプションに10を設定しているので，10ミリ秒毎に図を切り替えます．実行すると図15がアニメーションで表示されると思います．実際のアニメーションの動画はYouTube（https://youtu.be/8Ac7IOqN5tE）に置いてあるので，そちらをご覧ください．

```
1  # Jupyter 上でアニメーション実行のために必要
2  %matplotlib nbagg
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import matplotlib.animation as animation
6
7  # アニメーション関連の宣言
8  fig = plt.figure()
9  ims = []
10
11 # Lotka−Volterra 方程式に必要な正定数の宣言
12 a = 1
13 b = 0.01
14 c = 0.02
15 d = 1
16
17 # x と y の初期値
18 x0 = 20
19 y0 = 20
20
21 # オイラー法の時間幅は 1/200とする
22 nt = 200
23 dt = 1/nt
24
25 MAXT = 10 # 時間の最大値
26 MAX_STEP = MAXT ∗ nt # ステップの最大値
27
28 # 変数x, y はステップ s におけるx,
       y の値を保持し，毎ステップ更新する
29 s = 0
30 x = x0
31 y = y0
```

[2)] ArtistAnimationにはグラフのデータのリストを渡さないといけません．plot（折れ線グラフ）の場合は，リストを返すので，ims.append(im)でよいですが，scatter（散布図）やimshow（2次元画像データ）の場合は，リストを返さないので，ims.append([im])というように[ ]をつけてリスト化して渡す必要があります．

[3)] ArtistAnimationの代わりにFuncAnimation関数を使うアニメーションの作成方法もあります．

```

# プロットに使うxとyの値を保持するリスト
x_list = [x0]
y_list = [y0]

# 時刻t=0のプロット
im_x = plt.plot([0], x_list, c="green", label='x')
im_y = plt.plot([0], y_list, c="red", label='y')
ims.append(im_x + im_y)

for i in range(MAX_STEP+1):
    # オイラー法の実施
    x_ = x + (a*x - b*x*y)*dt
    y_ = y + (c*x*y - d*y)*dt
    x = x_
    y = y_
    s += 1

    # 10回に1度プロット
    if s%10 == 0:
        # 現在のx,yをプロット用リストに追加
        x_list.append(x)
        y_list.append(y)

        t_array = np.arange(0, (s+1)/nt, dt*10) # 横軸用の配列

        # t=0から現在までをプロットし，アニメーションに登録
        im_x = plt.plot(t_array, x_list, c="green")
        im_y = plt.plot(t_array, y_list, c="red")
        ims.append(im_x + im_y)

plt.legend() # 凡例の表示
ani = animation.ArtistAnimation(fig, ims, interval=10) # アニメの作成
plt.show()
```

ソースコード 4　個体群動態のアニメーションのプログラム

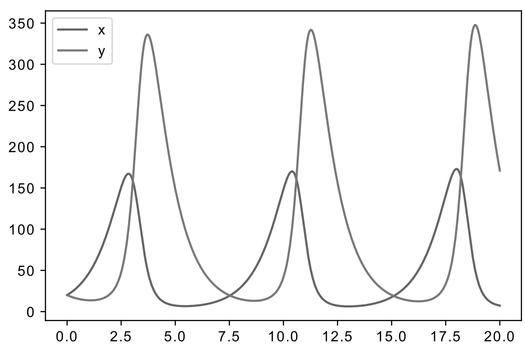

図 15　ロトカ–ヴォルテラ方程式の個体群動態のアニメーション．

4. おわりに

今回は力学系の数理解析と数値計算についてロトカ–ヴォルテラの捕食者–被食者モデルを題材にPythonでいろいろやってみました．

数理解析は代数計算ができるSymPyライブラリを使いました．全て手で計算できる内容ですので，実際にやるときはSymPyを使ってやる必要は特にないかもしれません．しかし，手で解いた後の検算もSymPyを使えば簡単にできますので，手で計算をしつつ，SymPyで計算結果を確認するという使い方は有効かもしれません．同様のことは，Mathematicaなどを使ってもできると思います．

アニメーションはMatplotlibの機能でやってみました．今回はただの個体数を示す線が動くだけでしたが，個体ベースモデルをアニメーションで可視化すれば，見ているだけでも楽しいと思います．また別の機会で作ってみたいと思います．

最後に本記事との関連情報を紹介します．ビンガムトン大学の佐山先生の本では，力学系（離散時間と連続時間のどちらも）のモデルとその解析について、Pythonの豊富なコード[1]とともに紹介されています．英語ですが無料（！）[2]でダウンロードできますので，興味のある方はご覧ください [3]．同じく佐山先生が管理されているPyCX（Python-based sample code repository for complex systems research and education，つまり，複雑系の研究と教育のPythonのサンプルコードのリポジトリ）のGitHubサイト [4] には，110余りのサンプルコード（.pyなので，Pythonの素のコードの形式）が置いてあり，自由にダウンロードできます．Pythonの2/3系どちらにも対応しています．サンプルコードの中には力学系のモデルもたくさん含まれています．一からコードを作らなくても，ここにあるコードを見ながらいろいろ変更していく形でも十分勉強になると思います．

また，本記事の内容は第1回でも紹介した九州大学の野下先生の数理生物学演習 2019年度のページ [5] の中の第4回関連リンクから「数理生物学演習　第4回　ロトカ-ボルテラ モデル」の資料を参考にさせていただきました．そちらを参照しながら，本記事を読んでいただければ，プログラムに対する理解がさらに深まると思います．

以上で今回は終わりです．また次回の記事でお会いしましょう！

謝辞

以下の方々（敬称略）は，本記事に目を通してくださったり，大変有益なコメントを下さいました．心より感謝いたします．
伊東啓（長崎大学），佐山弘樹（ビンガムトン大学），鈴木麗璽（名古屋大学），野下浩司（九州大学），間宮

[1] ただし，Python2系で書かれています．本記事で使用している3系とは文法などにいろいろ違いがありますので，ご注意ください．

[2] 印刷版を買うこともできます．

安曇（静岡大学）

以下の編集委員の方々にも本記事をチェックしていただきました．ありがとうございました．

佐藤一憲（静岡大学），宮崎倫子（静岡大学），守田智（静岡大学）

参 考 文 献

[1] ニュースレター用の GitHub. https://github.com/igenki/biomathNL.

[2] SymPy. https://www.sympy.org/en/index.html, Accessed 27 May 2020.

[3] Hiroki Sayama. *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY Textbooks, 2015. https://textbooks.opensuny.org/introduction-to-the-modeling-and-analysis-of-complex-systems/.

[4] Python-based sample code repository for complex systems research and education. https://github.com/hsayama/PyCX.

[5] 野下浩司. 数理生物学演習 2019 年度. https://koji.noshita.net/page/compbio/compbio2019/, Accessed 30 March 2020.

学会事務局からのお知らせ

1. 会員管理システム (SMOOSY) の導入

2020 年 12 月から、会員管理システム (SMOOSY、`https://www.atlas.jp/services/membership`) を導入します。SMOOSY 導入に伴い、会員の皆様は振り込んだ会費・未納会費の確認や登録されている住所確認・変更、領収書の発行等を各自で実行して頂くことが可能となります。また、これまでは会費の納入方法は郵便振替のみでしたが、クレジットカード等によるオンライン決済も可能となります。システム導入するにあたり、現在登録されているメールアドレスが有効か必ず確認をしてください。現在利用されている有効なメールアドレスでなければパスワードの発行等はできませんので十分注意をしてください。もし、登録したメールアドレスがわからない場合、未登録の場合は事務局 (secretary@jsmb.jp) まで問い合わせてください。

はじめてログインをする際に実施していただくことは下記の通りです。

(1) SMOOSY システムからログインボックスの [パスワード再発行はこちら] をクリックして、ログイン ID（会員番号またはメールアドレス）を送信する。
(2) パスワード設定 URL をクリックする。
(3) パスワードを登録する。

ただし、パスワードは事務局が代理で登録することはできませんので十分に注意してください。smoosy でできること（はじめてログインする場合、会員情報を確認・変更、メールアドレスの確認・変更、支払方法の変更、請求書・領収書の発行）に関する詳細は https://fs.smoosy.atlas.jp /html /help /mypage /index.html をご参照ください。ただし、支払い方法はクレジットカード払い、コンビニ払い、郵便振替払いの 3 種のみです。新規に入会希望される方がみえる場合、下記アドレスを参考に入会手続きを進めていただくようにお伝えください。

https://fs.smoosy.atlas.jp /html /help /mypage / index.html

2. 会費納入のお願い

日本数理生物学会の年会費は 2020 年 1 月～12 月の 1 年分で

正会員	3000 円/年
学生会員	2000 円/年

です。会員は数理生物学会年会の登録費割引の特典を受けることもできます。また、会員は年会で発表することができ、学会役員選挙において投票することもできます。今年度または過去の会費未納の方は、下記口座への納入をお願いいたします。

【ゆうちょ銀行の振替口座】
口座番号:00820-5-187984
口座名称 (漢字):日本数理生物学会
口座名称 (カナ):ニホンスウリセイブツガッカイ
【ゆうちょ銀行以外から振込】
店名 (店番):〇八九 (ゼロハチキュウ) 店（089）
預金種目:当座口座番号:0187984

3.2021 年 大久保賞受賞者のお知らせ

2021 年の大久保賞候補者が決定しました。例年ならば，総会で報告を経て授賞式を実施する運びになりますが 2020 年はオンラインで学会を開催いたしますので受賞者をニュースレターにて報告させていただきます。

Michael Savageau (The University of California Davis)

上記の受賞者は JSMB、SMB 両学会の構成員からなる大久保賞選考委員の審議によって決定いたしました。JSMB 大久保賞選考委員：時田恵一郎．瀬野裕美，佐竹暁子，Alexander Anderson，Denise Kirschner，Ping Ye.

4. 第 15 回（2020 年度）日本数理生物学会研究奨励賞受賞者のお知らせ

選考委員会から 2020 年度研究奨励賞選考の結果が報告され運営委員会で承認を受けました。例年ならば，総会で報告を経て授賞式を実施する運びになりますが 2020 年はオンラインで学会を開催いたしますので受賞者をニュースレターにて報告させていただきます。

入谷 亮介 氏 （理化学研究所・数理創造プログラム)
岡田 崇 氏 （理化学研究所・数理創造プログラム)

今年度は４名の候補者（内男性４名，女性０名）がありました。選考は，まず，選考委員６名のうち，候補者と利害関係のある選考委員（共同研究者や研究資金をやりとりする関係者）を除く選考委員が第１段審査を行いました。第１段審査では，各候補者の主要論文３編を含む応募書類を各選考委員が個々に吟味し，５段階の絶対評価と所見を委員長に報告しました。そして，その評価結果を選考委員会で共有し，授賞者の推薦を行う第２段審査に進みました。第２段審査審査では，少なくとも１名を推薦することに合意し，投票によって，１名を決定しました。その後，さらに議論を行い，２人目の受賞者を決定いたしました。

入谷氏は，２０１６年に九州大学において博士の学位を取得し，その後，複数のポスドクを経て，現在，理化学研究所・数理創造プログラムの研究員（任期付き）の職に就いています。入谷氏の専門は進化生態学であり，数学のバックグラウンドを生かし，特に宿主・寄生者系に関する研究を行っています。大学院時代には，古典的な Hamilton and May (1977) のモデルを発展させ，寄生者の感染の有無に依存した宿主の移動分散率の進化を研究しています。学位取得後は，宿主の幼体から成体への成熟期間に応じて，寄生者が幼体と成体の宿主それぞれから搾取するレベルの進化ダイナミクスを数学的に厳密に解析しています。そして，その結果から多様な種の感染率データと比較可能な理論的予測を導いており，実証実験との融合にも力を入れている点は特筆に値します。学位取得後５年目にして論文８報発表，うち２報が単著，４報が筆頭著者であり，業績の主体性が評価できます。また，国内外の研究集会に積極的に参加し，海外との共同研究を積極的に推進している点や，評価の高い学術誌において研究を発表している点も評価されました。今後も，進化生態学的側面から数理生物学への寄与が期待されます。

岡田氏は，２０１３年に京都大学において博士の学位を取得し，その後，複数のポスドクを経て，現在，理化学研究所・数理創生プログラムの上級研究員（任期付き）の職に就いています。岡田氏は，素粒子の分野で学位を取得し，その後，１年間，素粒子分野でのポスドクを経た後，数理生物学の分野に転身したという，異色の経歴をもちます。 数理生物学に関する主な業績は，理化学研究所のポスドクに着任後行われた，化学反応ネットワークに関するものです。化学反応ネットワークのトポロジカルな構造と，そのシステムへの摂動に対する反応との関係についての非常に興味深いルールを発見し，数学の定理として発表しています。また，この研究を分岐の問題に対しても発展させており，化学反応ネットワーク理論のさらなる発展が期待されます。また，岡田氏の研究は，数学的に厳密であるだけでなく，数学的な研究結果を実証研究へと結びつけようとしている点も高く評価されました。研究成果が物理学におけるトップジャーナルに掲載されている点や，海外経験が豊富である点も評価されました。岡田氏が受賞されることで，多くの優秀な若手研究者が異分野から数理生物学に参入し，数理生物学の発展に寄与するようになることが期待されます。

以上により，本選考委員会は，「数理生物学に貢献をしている本学会の若手会員の優れた研究を表彰することにより，研究の発展を奨励し，わが国の数理生物学の一層の活性化をはかる」という授賞基準にふさわしい研究者として， 入谷亮介氏と岡田崇氏の両氏を日本数理生物学会研究奨励賞の授賞者として推薦いたします。

なお，今回，惜しくも選考に漏れた候補者も，それぞれ高く評価されたことを申し添えます。次回以降，再チャレンジを含めて，多くの方が応募されることを期待しています。

日本数理生物学会研究奨励賞選考委員会：岩見真吾，近藤倫生，佐竹暁子，瀬野裕美，時田恵一郎，今隆助（委員長）

5. 2020 年日本数理生物学会年次総会報告

Biomath メーリングリストおよび個別メールでお知らせした通り，2020 年の年次総会は運営委員会の議論と承認をもって仮の議決とし来年の総会で正式に承認されるものとなります。下記に提案の内容と議論の内容を記載いたします。

■報告事項

(1) 2021，2022 年大会の開催場所について

2021 年は宮崎大学，2022 年は広島大学で開催を予定していると報告があった。

(2) 2020 年研究奨励賞受賞者，大久保賞受賞者の報告

2020 年研究奨励賞受賞者は入谷亮介さん（理化学研究所)、岡田崇さん (理化学研究所)、大久保賞は Michael Savageau (The University of California Davis) さんが受賞したことについて報告があった。

(3) 学会員の管理，HP に関して

今年の 12 月に会員管理システム (SMOOSY) の導入を予定していることおよび導入に合 わせて HP も移行準備中であることが報告された。HP に関しては移行準備が整い、運営委員会での承認が済み次第、新しいホームページに移行する予定であることを報告した。

(4) その他

特になし

■議題

(1) 2019 年決算，2020 年予算執行状況，2021 年予算案

議論に際して，下記を事務局側から資料を提示した（資料を併せて参照されたい)。

（2019 年決算 説明）収入に関して，東京大会の還元費が組み込まれていることを提示した。支出に関して，次の内容を提示した。通信費の予算は 5000 円であったが支出は 40,972 円となった。土倉事務所から請求された通信費 (31,977 円) と幹事長が支出した郵送費 (8,995 円) を合算したものである。これらは選挙のための通信費を主目的としたものである。奨励賞経費に関して，シドニー大会による奨励賞受賞講演の延期により 2019 年に研究奨励賞受賞者（2017 年受賞者, 2018 年受賞者）および大久保賞受賞者を併せて表彰したため必要経費が予算より多くなった。外部委託経費に関して，2019 年に土倉事務所へ支払った経費は 2018 年の経費であり 2019 年の経費は 2020 年に請求される。

（2020 年決算 説明）収入に関して，8 月上旬までの収入を示している。支出に関して，ニュースレターに関しては現在請求が来ている No. 90 の印刷代のみが支出としてある。外部委託経費，2019 年の土倉事務所から請求された経費を支出している。会員管理クラウドサービスを 2020 年 12 月から導入することに伴い 2020 年 12 月に，2020 年 12 月から 2021 年 3 月分の費用を支払う予定である (96,800 円，委託先の企業の決算年度に合わせるため請求が 2021 年 3 月までとなっている。)。web 会議システムを今年度から導入したのでその経費が支払われている。事務局諸経費について，2020 年予算より大幅に増加している。これは会則の英訳代金として 68,782 円を支出したこと，振込手数料および土倉事務所からの諸経費の請求を合算したものである。特別会計から，名古屋大会へ 300,000 円支出をしている。

（2021 年決算案 説明）一般会計の収入に関して，2020 年の予算執行後の残金から 2020 年のこれからの支出見積もりを差し引きし算出した。会費収入は 1,000,000 円とした。一般会計の支出に関して，ニュースレターは例年通りの予算とした。会費請求は例年 40,000 円程度かかるため，2021 年もそれに倣った。2021 年は選挙を実施するため，2019 年の支出から 150,000 円を選挙関連として計上した。事務局経費については，予算を大幅に上回ったため 2019 年決算を参考に多めに計上した。会員管理クラウドサービスは委託先の決算年度に合わせて 2021 年 4 月から 2022 年 3 月までの見積額を計上した。また，2020 年の土倉事務所への外部委託費は翌年の 2021 年に請求されるため，2019 年決算を基に計上した。事務所経費については 2019 年における土倉事務所からの請求等を勘案して多めに計上をした。特別会計の収入に関して，例年通り一般会計から 100,000 円の繰り入れを計上した。例年、年次大会は学会からの借り入れ分と参加費で賄い、黒字がでた場合は学会の一般会計の繰り入れとして計上し，翌年もしくは翌々年に特別会計へ繰り入れをしている年もある。ただし、2020 年の年次大会ではオンラインで実施し参加費を徴収しないため、大会還元費から一般会計への収入は見込まれない。

事務局からの資料報告・説明に対して，特別会計から年次大会運営費へ支出した後に年次大会運営費で黒字がでた場合の取り扱いについては複雑であるため明確に記載をするように意見がでた。

(2) 大久保賞選考委員 1 名の交代

事務局から，今年で任期満了となる時田恵一郎委員の次の委員として，増田直紀会員が運営委員会から推薦されたことを報告し，承認を得た。

(3) その他

特になし

6. Biomath メーリングリスト登録のお願い

日本数理生物学会では， 会員と会員でない数理生物学に関心をお持ちの方々との交流や情報交換を目的とする， Biomath メーリングリストを運営しています. Biomath メーリングリストには, 学会や会員からの重要な情報 (大会情報, 国内外の公募情報, 研究会や定例セミナーの情報, 学会賞の情報など) が投稿されますので, 日本数理生物学会に新規に入会されるときには, 合わせて Biomath メーリングリストへの登録をお願いしています. また, 現在会員の方で Biomath メーリングリストに未登録の方にもぜひ登録いただきますようにお願いいたします. つきましては, 未登録の方には, お手数ですが, 以下のいずれかの方法で Biomath メーリングリストへご登録ください.

(1) Biomath メーリングリストに自分で登録する：
登録は本文も件名も空白の電子メールを biomathml-subscribe@brno.ics.nara-wu.ac.jp にお送りいただくと, 確認メールが返送されます. それに返信していただくと入会することになります.

(2) Biomath メーリングリストに登録するが, 登録作業は事務局にしてもらいたい：
登録を希望する電子メールアドレスを事務局までお知らせください.

登録された皆様の電子メールアドレスは厳重に管理します. 登録者以外からは投稿できないシステムになっておりますので迷惑メールの心配もありません. 配送頻度も週に 1 通程度となっております. その他, Biomath メーリングリストに関しましては

http://jsmb.jp/biomath/biomath.html

に記載しております. 合わせてご覧ください.

7. 事務的事項のお問い合わせについて

入会, 退会の申し込み, 会員情報 (所属, 住所, ニューレター送付先など) の変更は, 業務委託先の土倉事務所 (bwa36248@nifty.com) にご連絡ください. 会費の納入状況の確認などの事務的問い合わせにつきましても, 土倉事務所までお問合わせください. それ以外の事項につきましては, 幹事長の岩田繁英（secretary@jsmb.jp）へお問い合わせください.

8. 事務局連絡先

幹事長　岩田繁英 (SHIGEHIDE IWATA)
会計　中丸麻由子 (MAYUKO NAKAMARU)
幹事　大泉 嶺 (RYO OIZUMI)
幹事　立木佑弥 (YUUYA TACHIKI)
〒108-8477 東京都港区港南 4-5-7
東京海洋大学学術研究院 岩田繁英
E-mail: secretary@jsmb.jp

また, 業務の一部委託先は次の通りです.

土倉事務所内 日本数理生物学会
〒 603-8148 京都市北区小山西花池町 1-8
Tel: 075-451-4844 E-mail: bwa36248@nifty.com

≪編集後記≫

会長の稲葉寿先生を通して，昨年 10 月のフィンランドで行われた Odo Diekmann 先生のレクチャーのノートをいただきました．今号から数回に亘って掲載いたします．1 回目は，Kermack と McKendrick の感染症モデルにまつわる話です．稲葉先生の編集後記も合わせてお読みください．誰でも「知っている」有名なモデルですが，あなたは本当に正しく理解できていますか，と問われたら，果たして自信を持ってイエスと答えられるでしょうか？　改めて，1927 年の論文をじっくりと読まなければと思っています．ここで紹介されている Kermack の事故のことや，McKendrick の人柄を知った上で読めば，また一味違ったものになりそうです．

2019 年研究奨励賞を受賞された黒川瞬さんに，ご自身のこれまでの研究生活を振り返っていただきました．会社員を経験されたり，積極的に色々な大学に所属して，色々な人たちと交流されたことがよくわかります．数理生物学を研究している人で，長期間にわたって中国に留学した経験のある人はまだまだ少ないと思いますので，もう少し詳しく話を聞いてみたいと思っている方も多いのではないでしょうか．

一ノ瀬元喜さんと宮川大樹さんには，連載の 2 回目として，Python による力学系の数理解析と数値計算について執筆していただきました．会員の皆さんにとってとても興味深い内容になっていると思います．ぜひご自分の計算機を使って試してみてください．(佐藤)

日本数理生物学会ニュースレター No.92
2020 年 9 月発行
編集委員会　佐藤一憲・宮崎倫子・守田智・一ノ瀬元喜
sato.kazunori@shizuoka.ac.jp
国立大学法人 静岡大学
〒 432-8561 静岡県浜松市中区城北 3-5-1
発行者　日本数理生物学会
The Japanese Society for Mathematical Biology
http://www.jsmb.jp/
印刷・製本　（株）ニシキプリント PDF 版

日本数理生物学会
2019年度決算

会計 中丸 麻由子

一般会計			2019年度予算		2019決算
収入					
	繰越		7,025,617		7,214,790
	会費		1,000,000		997,059
	大会還元金		0		454,673
	利子等		0		11
		計	8,025,617		8,666,533
支出					
	ニュースレター	過年度	0	(*1)	31,760
	ニュースレターN87	冬印刷	90,000		92,340
	ニュースレターN87	冬郵便	30,000		27,690
	ニュースレターN88	春印刷	90,000		92,340
	ニュースレターN88	春郵便	30,000		37,052
	ニュースレターN89	秋印刷	70,000		94,050
	ニュースレターN89	秋郵便	40,000		38,583
	会費請求		40,000		11,234
	名簿		0		0
	選挙(含 発送費, 旅費)		120,000		110,666
	通信費等	通信費	5,000		40,972
		ドメイン名経費	3,000		2,916
		奨励賞経費	15,000		18,036
		サーバ維持費	40,000		29,700
	事務局経費	外部委託	400,000	(*2)	344,542
		事務諸経費	7,000		14,426
	特別会計へ		100,000		100,000
		小計	1,080,000		1,086,307
	予備費(次年度繰越)		6,945,617		7,580,226
		計	8,025,617		8,666,533

特別会計				
収入				
	繰越		3,208,563	2,846,003
	繰り入れ(一般)		100,000	100,000
	寄付		0	0
		計	3,308,563	2,946,003
支出				
	年会大会費(含 託児支援, 旅費)		400,000	0
	CIJK大会費(含 託児支援)		0	0
	大久保賞関連		0	215,439
	SMB合同大会費(含 若手旅費援助)		1,000,000	0
		小計	1,400,000	215,439
	予備費(次年度繰越)		1,908,563	2,730,564
		計	3,308,563	2,946,003

備考
(*1) ニュースレター86号(2018年度分)の発送費(秋郵便代)として
(*2) 2018年度事務委託費として

監査報告

日本数理生物学会の 2019年度の収入および支出に関する証書類を調べ、全て適正に執行され、決算報告にも誤りが無いことを確認しました。

2020 年 9 月 2 日

会計監事 中岡 慎治

日本数理生物学会
2020年度予算執行状況・2021年度予算

会計幹事　中丸 麻由子
（執行状況は2020年8月21日現在）

一般会計			2020年度予算		2020年度執行状況		2021年度予算
収入							
	繰越		7,100,000		7,570,960		7,200,000
	会費		1,000,000		869,600		1,000,000
	大会還元金		0		0		0
	利子等		0		5		0
		計	8,100,000		8,440,565		8,200,000
支出							
	ニュースレター	過年度	35,000		0		35,000
	ニュースレターN90	冬印刷	95,000		55,400		95,000
	ニュースレターN90	冬郵便	35,000		0		35,000
	ニュースレターN91	春印刷	95,000		0		95,000
	ニュースレターN91	春郵便	35,000		0		35,000
	ニュースレターN92	秋印刷	95,000		0		95,000
	ニュースレターN92	秋郵便	40,000		0		40,000
	会費請求		40,000		0		40,000
	名簿		0		0		0
	選挙（含 発送費, 旅費）		0		0		150,000
	通信費等	通信費	5,000		0		20,000
		ドメイン名経費	3,000	(*1)	2,970		3,000
		奨励賞経費	15,000		0		15,000
		サーバ維持費	40,000	(*2)	29,700		30,000
	事務局経費	外部委託	400,000	(*3)	380,446	(*7)	400,000
		会員管理クラウドサービス		(*4)	0	(*8)	310,000
		Web会議システム		(*5)	67,437		70,000
		事務諸経費	7,000	(*6)	80,402		15,000
	特別会計へ		100,000		100,000		100,000
		小計	1,040,000		716,355		1,583,000
	予備費（次年度繰越）		7,060,000		7,724,210		6,617,000
		計	8,100,000		8,440,565		8,200,000

特別会計			2020年度予算	2020年度執行状況	2021年度予算
収入					
	繰越		2,730,564	2,730,564	2,530,564
	繰り入れ（一般）		100,000	100,000	100,000
	寄付		0	0	0
		計	2,830,564	2,830,564	2,630,564
支出					
	年会大会費（含 託児支援, 旅費）		400,000	300,000	300,000
	CIJK大会費（含 託児支援）		0	0	0
	大久保賞関連		0	0	300,000
	SMB合同大会費（含 若手旅費援助）		0	0	0
		小計	400,000	300,000	600,000
	予備費（次年度繰越）		2,430,564	2,530,564	2,030,564
		計	2,830,564	2,830,564	2,630,564

備考
(*1)　アリクイネット　2021年7月31日まで契約
(*2)　アプライド　2020年9月1日-21年8月31日分サーバー運用保守＋ドメイン更新／設定／管理
(*3)　2019年度事務委託費として
(*4)　2020年12月よりSmoosyの導入。2020年12月-21年3月分見積金額は96,800円（未請求）
(*5)　2020年度よりWherebyの導入
(*6)　会則の翻訳代68,782円を含む
(*7)　2020年度事務委託費として。なお、2021年度業務委託費の見積金額120,329円で、2022年度予算として計上。
(*8)　2021年4月-22年3月分Smoosy利用代

TABLE OF CONTENTS

Newsletter of the Japanese Society for Mathematical Biology　No. 92　September 2020

MATHEMATICAL BIOLOGY