

国際 ICT 利用研究学会

国際 ICT 利用研究  
研究会講演論文集  
第 9 回

2021 年 3 月 21 日

オンライン開催@ZOOM

## 第9回 国際ICT利用研究学会 研究会 プログラム

下記のように第9回研究会をWeb開催しますので、ご案内申し上げます。

今回も集合開催は中止し、Web上での開催を実施します。このWebでのご講演・聴講によって本研究会が成立したものといたします。

本研究会の講演論文集は後日 OnLine edition: ISSN 2432-7956 として Web 上に置きます。

また、会員により発表された内容で、あらためて投稿された後、査読を経た論文（非会員共著者は1名まで）については、Transactions of the IIARS（IIARS 研究会論文誌）No.3、もしくは Journal of IIARS（IIARS 学術研究論文誌）Vol.5 以降に掲載します。

山下倫範（立正大学）

日時 2021年3月21日（日） 11:30-16:35

会場 Web（会員には直接お知らせいたします）

参加費 無料

### プログラム

11:30 開会の挨拶 福田真規夫（太成学院大学）

#### 第1セッション（11:35-12:35、座長 次郎丸 沢（OME）

11:35-11:50

○星野龍世、並木愛結、根本 光（大東文化大学経営学部）、木川 裕（日本大学法学部）  
遠隔医療を高齢者に浸透させるための提案

11:50-12:05

○岡村 和（日本大学法学部）

SNS普及から見る若者の間でマッチングアプリが流行した心理的要因

12:05-12:20

○櫻井広幸（立正大学心理学部）

VR研究を中核とする、教育、テレワーク、ロボット、AIの連携

12:20-12:35

○鎌田光宣（千葉商科大学人間社会学部）

ブロックチェーンの仕組みと課題

12:35-12:40 休憩

#### 第2セッション（12:40-13:40、座長 田中雅章（ユマニテク短期大学）

12:40-12:55

○佐原麻衣子（日本大学法学部）

テレビドラマ視聴率とSNSを活用したプロモーション戦略の関係性

12:55-13:10

○木川明彦（立正大学）

ポストコロナにおける情報教育の役割

13:10-13:25

○松本貴裕（大阪電気通信大学）、福井昌則（兵庫教育大学）、高見友幸（大阪電気通信大学）

画像を題材とした競技プログラミング用コンテンツの開発

13:25-13:40

- 佐久間貴士（千葉県立保健医療大学健康科学部）  
保健医療系大学における情報倫理教育の取り組み

13:40-13:45 休憩

**特別セッション**（13:45-14:10, 座長 鈴木治郎（信州大学））

- 渡瀬泰成（立正大学）  
数学定理の証明のコンピュータ（証明支援器）による形式化の研究

14:10-14:15 休憩

」

**第3セッション**（14:15-15:15, 座長 田中敏幸（慶應義塾大学））

14:15-14:30

- 田中雅章（ユマニテク短期大学）  
保育士養成における相互評価法を活用した工作実践

14:30-14:45

- 中村洋介, 曾根 新（福島大学）  
日本の活断層直上の高速道路分布の調査

14:45-15:00

- 木子 香（大阪電気通信大学）  
『三才図会』の雙陸盤面図について

15:00-15:15

- 高見友幸（大阪電気通信大学）  
日本の古代都城における設計数値の継承（序報）

15:15-15:20 休憩

**第4セッション**（15:20-16:35, 座長 宮田大輔（慶應義塾大学））

15:20-15:35

- 佐藤礼華, 戦 揚（大阪電気通信大学）  
3DCG 動画及びVR の活用による手術学習支援システムの構築

15:35-15:50

- 田中敏幸, 小林寛知（慶應義塾大学理工学研究科）  
完全畳み込みネットワークを用いた大腸生検診断支援システム

15:50-16:05

- 次郎丸 沢（OME）  
ハイブリッド形式の国際会議における参加費の設定に関する一考察

16:05-16:20-

- 永田 清（大東文化大学）  
Eclipse 統合環境における PARI Library を使った C プログラミングとその応用ー

16:20–16:35

○鈴木治郎（信州大学）

連分数公式を予測するラマヌジャン・マシンの紹介

16:35 閉会の挨拶 上山俊幸（千葉商科大学）

# 遠隔医療を高齢者に浸透させるための提案

星野 龍世<sup>†</sup> 並木 愛結<sup>†</sup> 根本 光<sup>†</sup> 木川 裕<sup>‡</sup>

<sup>†</sup>大東文化大学 経営学部 <sup>‡</sup>日本大学 法学部

キーワード：遠隔医療，高齢者，インターネット

## 1 はじめに

近年、新型コロナウイルス感染症の影響で感染を恐れ病院に行けないという高齢者が増えている。

日経メディカルが実施した新型コロナウイルス感染症の影響による外来患者数の変化について医師 3,427 人へのアンケート<sup>1</sup>によると、昨年 3 月と比べて患者数が減っているのは全体の 53.4%であった。外来患者が 50%以上減っているのは 3.5%、25%以上 50%未満は 13.7%、25%未満が 36.0%で、約半分以上で患者数が減少していると回答 (図 1) している。

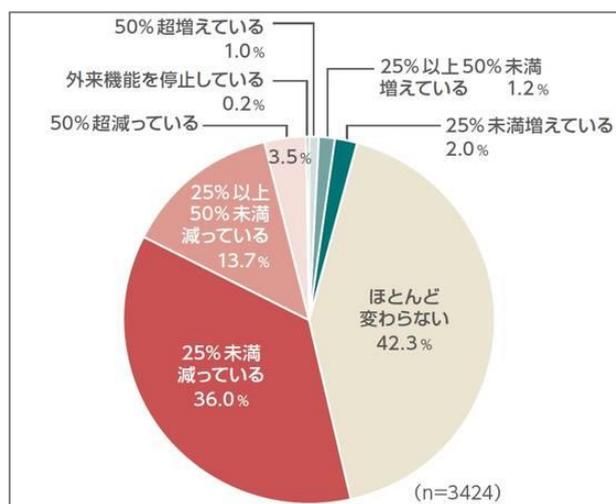


図 1. 2020 年 3 月と比べた患者数の変動

上記のデータから、少しの体調不良であっても病院に行かないで我慢する人がコロナ前と比較して増加していることがわかる。もともと高齢者は加齢とともに体調を維持、またはそれを治療する

ために数多くの病院に通っている割合が高い。通常であれば普通に行っていた手術も、緊急な手術であれば行すが、今すぐ行わなくてもよい手術であれば、感染リスクを回避するために延期する人も増えているのだろう。

しかし、通院を避けることによるリスクも同時に考えられる。たとえば、通院を我慢することで持病が悪化してしまう高齢者もいるだろう。このことにより決められた薬の処方が不十分となるなど様々な問題が生じてくる恐れがある。これを放置することは、高齢者医療にとって将来的にも大きなリスクとなることが考えられるだろう。

これを改善するための方法として遠隔医療が挙げられる。

厚生労働省によると、東京都における電話や情報通信機器を用いて診療を実施する医療機関は 2,040 施設<sup>2</sup>であり、このうち、電話診療を除くオンライン対応医療機関は 1,915 施設、そのうち初診から対応している医療機関は 879 施設であり、2020 年 4 月に東京都でオンライン診療を実際に行った医療機関は 379 施設とされ<sup>3</sup>、もちろん現在ではさらに増加していると考えられる。

ところで、遠隔医療を行うのにパソコンやタブレット、スマートフォンなどが必要になってくる。

<sup>2</sup> 厚生労働省、「電話や情報通信機器を用いて診療を実施する医療機関の一覧 (東京都)」, 新型コロナウイルス感染症の感染拡大を踏まえたオンライン診療について (対応医療機関リスト), <https://www.mhlw.go.jp/content/000732314.pdf>, (2021 年 3 月 19 日参照)

<sup>3</sup> Med it tech, 「東京都でのオンライン診療実績のある医療機関は 379 施設 2020 年 4 月実績」, 2020 年 5 月 30 日, (2021 年 3 月 19 日参照)

<sup>1</sup> 日経メディカル「患者数の変動について」(2020 年 7 月 9 日)

高齢者のスマートフォン等によるインターネットの利用状況は、ここ数年で急激に伸びている(図2)。

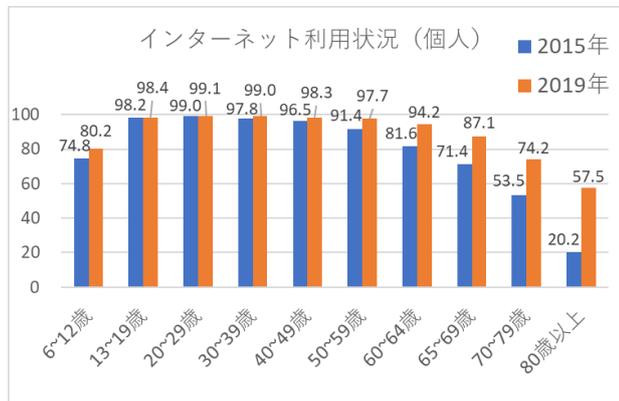


図2. 個人のインターネットの利用者の割合<sup>4</sup>

しかし、高齢者がスマートフォンに代える理由の多くは決して積極的なものばかりではなく、近いうちにフィーチャーフォン(ガラケー)が使えなくなるための買い替え促進という携帯キャリアの一種の販売戦略という消極的な理由であるという指摘<sup>5</sup>もある。実際、高齢者の多くはビデオ通話の利用方法が分からず、遠隔医療のハードルが非常に高くなっていることが考えられる。

そこで、高齢者にオンライン診療のハードルを下げ、利用を促進させる方法について検討していきたい。

## 2 遠隔医療の現状

### 2.1 遠隔医療の定義

最近、新型コロナウイルス感染症の影響から感染対策として遠隔医療の必要性が高まっている。スマートフォンの高性能化も進み、気軽に遠隔医療サービスを受けることが可能となってきた。

令和元年7月に改定された「オンライン診療の適切な実施に関する指針」の中で、遠隔医療とは、「情報通信機器を活用した健康増進、医療に関する

行為」とされ、また、オンライン診療とは、「遠隔医療のうち、医師-患者間において、情報通信機器を通して、患者の診察及び診断を行い診断結果の伝達や処方等の診療行為を、リアルタイムにより行う行為」と定義されている<sup>6</sup>。

これらは単に高齢者医療への対応としてだけではなく、働く世代や子育て家庭などの社会的ニーズにも応える形でオンラインでの診察や健康相談のサービスの提供を行っている。こういったサービスは、現代の社会的ニーズに対応するばかりではなく、医師不足や医療費増加問題の解決の糸口にもなることも見込まれ、今後は更なる遠隔医療の普及により医療サービスがより充実したものになっていくことだろう。

また、遠隔医療が示す行為には大きく分けて、「医師と医師間の行為(D to D)」「医師と患者間の行為(D to P)」の2つに分けられる。医師と医師間の代表的な例としては、患者の画像診断データをやりとりし、遠隔で読影を行うなどの遠隔放射線画像診断や遠隔術中迅速病理診断などがあり、医師と患者間の遠隔医療の代表的な例としてはオンライン診療・オンライン受診勧奨・遠隔健康医療相談などがある(図3)。

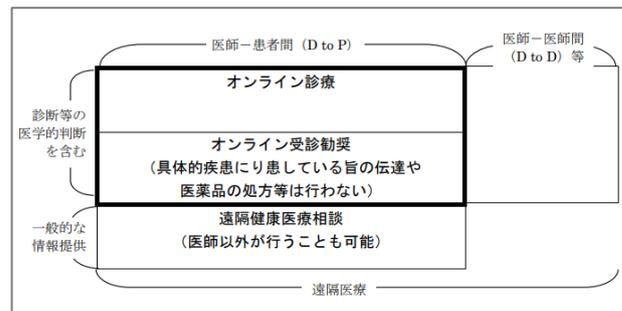


図3. 遠隔医療、オンライン診療、オンライン受診勧奨、遠隔健康医療相談の関連について<sup>7</sup>

### 2.2 わが国の遠隔医療ガイドライン

現在、わが国では医療のオンライン化がようや

<sup>4</sup> 総務省「通信利用動向調査」平成27年版・令和2年版より筆者作成

<sup>5</sup> 根本光, 木川裕, 「超高齢社会におけるICT技術を浸透させるための提案」, 第5回 国際ICT利用研究学会全国大会論文集, 国際ICT利用研究学会

<sup>6</sup> 厚生労働省, 「オンライン診療の適切な実施に関する指針」(2019年7月一部改訂), p5, <https://www.mhlw.go.jp/content/000534254.pdf>, (2021年3月19日参照)

<sup>7</sup> 同指針, p6

く動き出している段階であり、遠隔医療のガイドラインが作成され、遠隔医療実施の指針が示されている。

指針では、「直接の対面診察と同等でないにしても、これに代替し得る程度の患者の心身の状態に関する有用な情報を、オンライン診療により得ること」や、「初診は、原則として直接の対面による診療を行うこと」等が最低限順守する事項<sup>8</sup>として定められている。

### 3 アンケート調査

#### 3.1 調査方法(被験者・実施時期)

前述したように、高齢者のインターネット利用におけるハードルの高さが遠隔医療の普及を妨げているとの仮説をもとに調査を実施した。

コロナ禍という状況のもと知人等の協力を得て身近な高齢者を対象にアンケートを実施した。

被験者数は26名であり、「はい」・「いいえ」という形式で4つの設問に回答してもらった。調査時期は2021年2月である。

#### 3.2 調査結果

「スマートフォンを持っているか」という設問に対し、26人中13人が「はい」、13人が「いいえ」と回答した。また、「スマホの使い方を知っているか」という設問に対し、10人が「はい」と答え16人が「いいえ」と回答した。

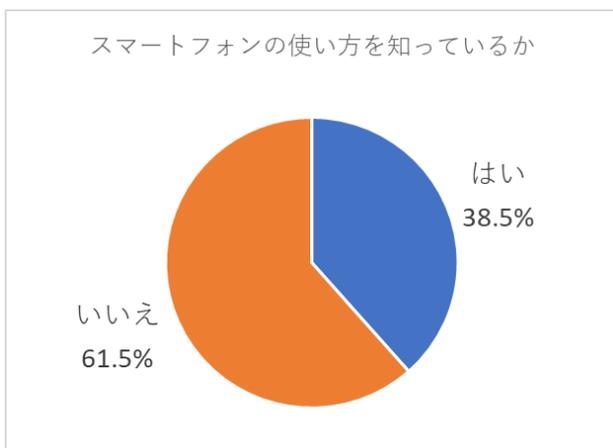


図4. スマホ操作方法の知識の有無

次に、「ZOOMの使い方を知っているか」という設問に対して、1人が「はい」、25人が「いいえ」と回答した(図5)。

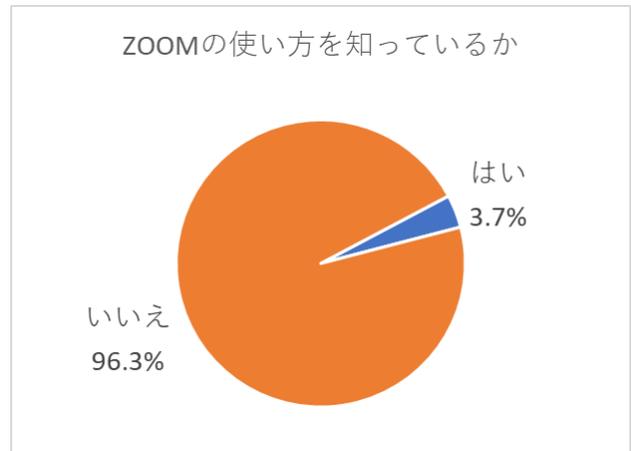


図5. ZOOM使用方法の知識の有無

最後に「アプリのインストール方法を知っているか」という設問では、前の設問と同じで1人が「はい」、25人が「いいえ」と回答した。

スマートフォンの所持率は半数であったが、実際に所持していてもスマートフォンの使い方については簡単な操作のみ知っているという割合がほとんどであった。Zoomの使用方法については、知っていたのは1人のみで、ビデオ通話を使える割合が少ないことが窺える。アプリのインストール方法についても同様の結果であった。

### 4 高齢者のスマートフォンの利用

#### 4.1 高齢者の利用可能範囲の拡大

このような状況を改善するためには、高齢者に遠隔医療の流れを教え、実際に利用できるようにトレーニングを行う場を設ける必要がある。併せて、ZOOM等のビデオ通話の使い方、必要なアプリのインストール方法などを教えることで遠隔医療が利用できない高齢者を減少させられるのではないだろうか。

#### 4.2 高齢者を対象とした講習会の必要性

高齢者のスマートフォン利用における「知らない」、「分からない」を少しでも減少させるためには、高齢者が気軽に参加できる講習会の実施が近道である。講習会に参加することでスマートフォ

<sup>8</sup> 前掲指針, pp11-13

ン等の利用方法を理解でき、便利さを実感することができれば、高齢者のタブレットの保有率や普及率も増加するだろう。

## 5 高齢者講習会の実施

講習会を実施する上での注意点としてコロナ禍における感染対策を最優先する必要がある。特に高齢者は重症化や死亡などのリスクも高く、最大限リスクを軽減することが大切である。このように感染予防を前提とする講習計画と十分な準備が必要である。

たとえば、使用する部屋について人が触れる場所のアルコール除菌や、常時換気を実施する。また、アルコール消毒液を設置し、参加者全員にアルコール消毒とマスク着用などの対策が必要である。また、席と席の間に十分な間隔を空け、パネルで仕切るなどソーシャルディスタンスの確保も重要となってくる。

高齢者の方への説明は大画面で説明した方が見やすいので、資料はなるべく大きな画面で表示するなどの工夫が必要である。可能であれば、講習会の会場は大学の教室を使用させてもらえればと考えている。大学にとっても地域への貢献としての意義もあり、進める価値は十分にあると考えられる。

遠隔医療の診察方法については、ガイドラインや指針に従って適切な方法で実施するシミュレーションを用意し、それに対応可能な知識を身に付けてもらうことを目的とする。

## 6 おわりに

本稿では、スマートフォンの利用方法が分からず、遠隔医療を利用できない可能性の高い高齢者に対し、高齢者に対応した講習会を実施することで遠隔医療に対するハードルを下げ、コロナ禍の医療に貢献する方法を検討した。

今回の調査においては十分な被験者数の確保は難しく、高齢者の現状を反映したものとなっているとは言えないかもしれない。被験者数を増やすには、今後ある程度コロナが収束するのを待たなければならないだろう。

しかし、いくつかの先行研究や総務省、厚生労

働省の調査結果からも高齢者にとって情報機器の操作は簡単なものではないことはわかっている。今回の新型コロナウイルス感染症において、私達若者は感染を広める悪役としてマスコミなどで捉えられている面も少なからずあるだろう。そうした若者の一人として、少しでも高齢者を守る手伝いができればという気持ちが本研究のきっかけでもあった。この国を築いてくれた先輩達高齢者に私達若者がどんな形であれ恩返しをすることができれば、本研究にも価値があったのではないだろうか。そうなることを願っている。

## 参考文献

- [1] 日経メディカル, 「患者数の変動について」, 2020年7月9日
- [2] 厚生労働省, 「オンライン診療の適切な実施に関する指針」, 2019年7月
- [3] 神奈川県保険医協会遠隔医療に関する実態調査結果, 2019年4月12日
- [4] 根本光, 木川裕, 「超高齢社会における ICT 技術を浸透させるための提案」, 第5回 国際 ICT 利用研究会全国大会論文集, 国際 ICT 利用研究会

# SNS 普及から見る若者の間でマッチングアプリが流行した心理的要因

岡村 和  
日本大学 法学部

キーワード：マッチングアプリ，Instagram，流行

## 1 はじめに

現代ではSNSによって簡単に異性の芸能人やアイドルグループと繋がれるようになり、現実では得られない様な世界観や非日常感を日常的に浸ることができるようになった。一方で、そのような充足感をSNSで埋めようとするほど脳内で勝手に「理想的な恋愛相手」というイメージが脳に形成され、大学やアルバイトなどオフラインで出会う異性に対して理想と現実のギャップが生まれ、オフラインの出会いに対して恋愛離れが起きているのでは無いだろうか。

事実、近年20代前半の若者を中心として恋愛行動における出会いのきっかけ作りに変化が生じている。

現代では最適なパートナーをオフラインに留まらず、オンライン上で見つけるアプリケーションサービス「マッチングアプリ」が流行している。

マッチングアプリとは「主に恋人や結婚相手同士がWeb上で繋がる仕組みのアプリ全般のこと」<sup>1</sup>を指す。気軽に理想の条件に合った異性との出会いのきっかけを作れる可能性がある反面、トラブルに巻き込まれるなどのリスクも存在するため、まだ世間的にはそのようなオンラインでの出会いに対して懐疑的な声が多いと言える。しかし、今後もマッチングアプリはインターネット環境の整備に伴い利用者が増加していくと予想されるため、教育者側にも利用者側の目線に立ち、社会環境の

変化によって生まれた要因を考慮、理解した上で教育方法を検討する必要があると言える。

そこで本稿ではマッチングアプリが流行した心理的要因について、特に世界観や非日常感を味わいやすいSNS「Instagram」の利用度と、恋愛や結婚の理想度の間の相関関係をアンケート調査によって統計的に明らかにし、考察を行うことを目的とする。

## 2 マッチングアプリの現状

ブライダル総研が発表した「婚活実態調査<sup>2</sup>」によると、2000年から2019年までに婚約したカップルの内、実際にネット系婚活サービスの利用を通じて結婚したカップルの割合も明らかになっている。(図1<sup>3</sup>)

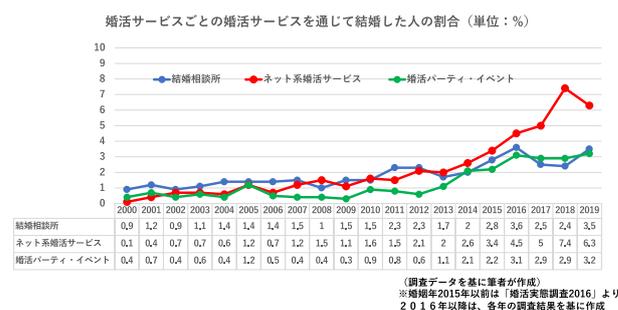


図1. 婚活サービスごとの婚活サービスを通じて結婚した人の割合

<sup>1</sup> マッチライブ編集部 三石. 「マッチングアプリとは? 利用前に知っておきたい出会い系との違いや料金体系・安全性を解説」より抜粋, 2020年12月22日, <https://find-model.jp/insta-lab/influencer-marketing-manual/> (閲覧日: 2021年1月13日)

<sup>2</sup> ブライダル総研. 「婚活実態調査2020(リクルートブライダ

ル総研調べ)」, 2020年9月16日, p1-12, [https://souken.zexy.net/data/konkatsu/konkatsu2020\\_release.pdf](https://souken.zexy.net/data/konkatsu/konkatsu2020_release.pdf). (閲覧日: 2021年1月13日)

<sup>3</sup> 前掲: 「婚活実態調査 2020(リクルートブライダル総研調べ)」 p4 「婚活サービスごとの婚活サービスを通じて結婚した人の割合」を基に筆者が作成.

ネット系婚活サービスによる結婚率が2000年から2019年にかけて0.9%から3.5%と約4倍の数値を示しており、また他のサービスの伸び率との比較をしても、近年のインターネットの普及に伴って著しく結婚する割合が増加していることが読み取れる。

さらに、マッチングアプリ「tapple」を運営する株式会社マッチングエージェント株式会社と、株式会社デジタルインファクトの共同での「マッチングアプリサービスに関して国内の市場規模調査」によると、2020年のオンライン恋活・婚活マッチングサービス市場は、前年比約2割増の620億円に拡大し、2025年には、2020年比約7割増の1,060億円に拡大すると予測が立てられている。(図2)<sup>4</sup>

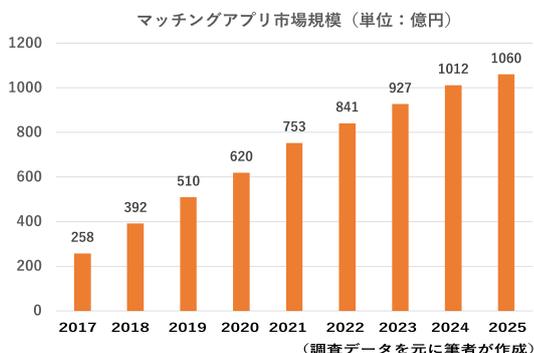


図2. マッチングサービスに関して国内の市場規模調査

このようにマッチングアプリ市場は成長著しい産業であることを示し、異性との出会い方の一つのきっかけとして主流になっていくことが予想される。

### 3 アンケート調査

#### 3.1 被調査者

本調査ではマッチングアプリ利用者の18歳～22歳の女子大学生30人に対し、アンケート調査を行った。また、調査期間は2020年9月下旬から2021年1月上旬まで行った。

なお、アンケート対象を女性に限定して検証を行った背景としては生物学的観点から見たときに、女性は男性を獲得するのではなく選ぶ立場にあることと、子孫を残せる数に限界があるため、理想の基準が上がりやすいという仮説から女性に限定して検証を行った。

そして、若年層の女性の中でも特にネットの利用率が高く、かつ時間的に余裕が多いと予想される女子大学生(18歳～22歳)に絞って行うこととした。また、調査に利用したマッチングアプリは「tapple」である。

#### 3.2 調査項目

以下はアンケート調査の質問項目である。(表1)

表1. アンケート調査の質問項目

【 Instagramの利用度についての調査】
当てはまるものを一つ選択してください
①1日にどれくらいInstagramを利用しますか?
<input type="checkbox"/> 30分以下 <input type="checkbox"/> 30分～1時間
<input type="checkbox"/> 1時間～2時間 <input type="checkbox"/> 2時間以上
②週に何回Instagramを開きますか?
<input type="checkbox"/> 開かない <input type="checkbox"/> ほぼ開かない(1～2回)
<input type="checkbox"/> たまに開く(3～4回) <input type="checkbox"/> ほぼ毎日開く(5～7回)
③週に何回自分についての発信、投稿をしますか?
<input type="checkbox"/> しない <input type="checkbox"/> ほぼしない(1～3回)
<input type="checkbox"/> たまにする(3～5回) <input type="checkbox"/> かなりする(5～7回)
【恋愛や結婚の理想についての調査】
どちらかを選択してください
①年収や職業、勤め先を気にする
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
②身長は175cm以上欲しい
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
③結婚に妥協はできない
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
④見た目が理想じゃないと付き合えない
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
⑤自慢できる彼氏じゃないとダメ
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
⑥優雅で裕福な生活がしたい
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
⑦専業主婦になりたい
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
⑧自分はいい女という自覚がある
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
⑨気に食わないところがあると冷める
<input type="checkbox"/> はい <input type="checkbox"/> いいえ
⑩過去に付き合っていた人より優れた人と付き合いたい
<input type="checkbox"/> はい <input type="checkbox"/> いいえ

<sup>4</sup> 株式会社サイバーエージェント 国内のオンライン恋活・婚活マッチングサービス市場規模予測 2020-01-27, <https://www.cyberagent.co.jp/news/detail/id=24280>

### 3.3 分析手法

分析手法は単回帰分析を採用する。そのため、独立変数に「Instagramの利用度」、従属変数には「恋愛や結婚の理想について」を置くこととする。

単回帰分析による結果から導かれるR2値から独立変数である「Instagramの利用度」が、従属変数である「恋愛や結婚の理想について」の高さに対しての相関関係の有無を統計的に明らかにする。

### 3.4 評定の尺度

「Instagramの利用度」と「恋愛や結婚の理想について」の相関係数を求めるに当たり、単回帰分析においては一定の数値的尺度が求められるため、以下の尺度とする。

#### 3.4.1 恋愛や結婚の理想について

各設問で「はい」を選択した数に応じて従属変数を決定する。各設問の配点を1点とし、「はい」を選んだ数をそのまま評定とし、最高評定は設問が10個あるため10とする。

評定尺度は、以下の通りとする。(表2)

表2. 恋愛や結婚の理想について

<質問項目>	<回答項目/評定尺度>	
①年収や職業、勤め先を気にする	はい 各1点	いいえ 0点
②身長は175cm以上欲しい		
③結婚に妥協はできない		
④見た目が理想じゃないと付き合いな		
⑤自慢できる彼氏じゃないとダメ		
⑥裕福で優雅な生活がしたい		
⑦専業主婦になりたい		
⑧自分は良い女という自覚がある		
⑨気に食わないところがあると冷める		
⑩過去に付き合ってた人よりも優れた人と付き合いたい		
総合最高評定	10	

#### 3.4.2 Instagramの利用度について

三つの各設問で選択した回答項目の合計を独立変数とする。

各設問での回答によって配点が変わり、設問毎

の配点の合計をそのまま独立変数とする。最高評定は設問1が最大4、設問2が最大3、設問3が最大3のため三つの和である10が最高評定となる。

評定尺度は以下の通りとする。(表3)

表3. Instagramの利用度

利用時間	
30分以下	1点
30分～1時間	2点
1時間～2時間	3点
2時間以上	4点
最高評定	4
利用頻度	
ほぼ開かない(1～2回)	0点
たまに開く(3～4回)	1点
ほぼ開く(5～6回)	2点
毎日開く	3点
最高評定	3
投稿頻度	
しない	0点
ほぼしない(0～1回)	1点
たまにする(2～4回)	2点
かなりする(5回～7回)	3点
最高評定	3
総合最高評定	10

## 4 集計結果

これらの質問項目と評価尺度を基に単回帰分析を行った。

以下は上記の二つの尺度から集計した結果になる。なお、従属変数である「恋愛や結婚の理想について」は表4において「理想の高さ」として表記している。

表4. 集計結果

回答者	インスタ利用度	理想の高さ
回答者1	7	5
回答者2	8	6
回答者3	5	3
回答者4	8	9
回答者5	4	3
回答者6	6	4
回答者7	6	7
回答者8	5	2
回答者9	4	1
回答者10	7	2
回答者11	9	9
回答者12	10	9
回答者13	8	8
回答者14	5	5
回答者15	4	3
回答者16	10	7
回答者17	6	6
回答者18	6	8
回答者19	6	7
回答者20	2	3
回答者21	5	7
回答者22	7	7
回答者23	6	5
回答者24	9	10
回答者25	7	7
回答者26	5	3
回答者27	8	9
回答者28	6	8
回答者29	9	8
回答者30	2	3
平均値	6.333333333	5.8

## 5 結果及び考察

### 5.1 調査結果

下記の図は単回帰分析の結果を基に、縦軸に「恋愛や結婚の理想について」横軸に「Instagramの利用度」を置き、それらを散布図にプロットし回帰直線を示したものである。

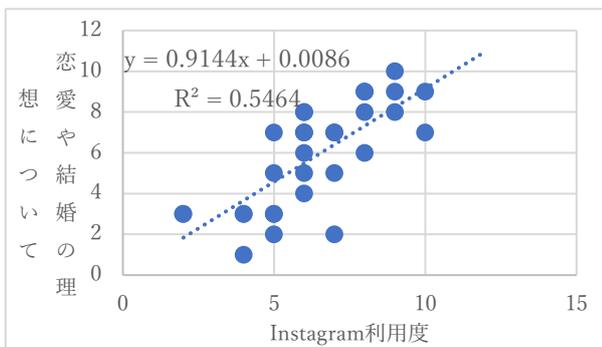


図3. 単回帰分析による調査結果

グラフより、単回帰式は $y=0.9144x$ を示し、相関関係の決定係数である $R^2$ 値は0.5464と有意な値を示した。

### 5.2 考察・結論

グラフより正の相関関係が見られたため、日常的にInstagramに触れる機会が多いほど求める理想像を高望みしやすくなることが示された。結果、自分の関わるコミュニティの異性に対し、恋愛対象の基準値に達する異性がおらず、マッチングアプリの利用へ移行したことが流行した一つの要因と考えられる。これが本稿における結論である。

## 6 今後の課題

これから積極的に恋愛活動に取り組むためには、自分の求める理想像が高すぎないか、そして自分の容姿や年収などを客観的に見て、過度に追いかけてすぎないように、自分の身の丈にあったパートナー選択を行うことが重要な考え方になるだろう。

また、今後の社会全体の流れとして、SNS依存しないような環境作りがより一層重要になると予想されるため、個人がSNS以外で充足感を満たせる趣味の発見や関わるコミュニティを増やすなどの取り組みが必要であり、それを気づかせるための啓蒙活動が教育者に求められると言える。

## 参考文献

- [1] ブライダル総研.「婚活実態調査 2020(リクルートブライダル総研調べ)」,2020年9月16日, [https://souken.zexy.net/data/konkatsu/konkatsu2020\\_relese.pdf](https://souken.zexy.net/data/konkatsu/konkatsu2020_relese.pdf). (閲覧日:2021年1月13日)

# VR 研究を中核とする、教育、テレワーク、ロボット、AI の連携

櫻井広幸<sup>†</sup>

<sup>†</sup>立正大学心理学部

キーワード：バーチャルリアリティ、テレワーク、フルダイブ型授業、魔法教室、ロボット、AI

## 1 はじめに

テレワークを強く推進する理由は、これまでのところ、大きく以下の三つである。

1. ワークライフバランス
2. 介護・育児等離職問題
3. 事業継続プラン (BCP)

このうち、三つめの BCP は、地震等の大規模災害の場合、一極集中の企業・事業はそこが被災すると事業が壊滅するため、テレワークによって拠点を分散させようという考え方である。大規模災害には世界規模の感染症によるパンデミックも含まれるが、これまでは他の二つの問題よりも訴求性が低い傾向にあり、こうしたテーマに触れている研究分野以外ではパンデミックという言葉も一般化していなかった。しかし今や、この言葉を知らない人の方が少ないほど、残念ながらごく短期間の間にこの事態があっけなく起きてしまった感がある。この被害は大学でも例外ではなく（むしろ大学には BCP の発想はなかった）、教育上、甚大な影響が発生している。VR の本質は体験（およびその提供）ということから考えると、VR は教育や学習と極めて親和性の高い研究テーマだといえる[1][2]。今回の感染症災害はやがて収まるはずであるが、自然災害や環境汚染、新しい感染症など、科学者としては、パンデミック自体は非常に残念ながらこの先また起きる可能性の方が高いという心構えを持たねばならない。したがって、VR による教育の研究は重要度を何倍にも増したと言えるであろう。

## 2 フルダイブ型 VR による教育と大学教員のテレワーク

本研究の中核は、フルダイブ型講義（ここでは、授業のほとんどを VR 空間で行う方式の講義を指す）を可能とする VR 空間の構築である[1]。方法としては、VR 空間構築ソフトにより、講師にと

って思い通りの教材を即興で作成・提示できる VR 空間を構築し、学生にとって学習を主体的に体験・理解できる空間を構築する。空間内の作業対象はワンド（杖）で指定し、作業内容は“呪文”のコマンドを発声して指定するルールを採る。先行研究では、サイバークロブの動きによってホワイトボードを出現させ、そこへの板書を可能とするシステムを試作した[2]。授業が VR 空間へのフルダイブ型の形式で行われる場合、最終的にはヘッドマウントディスプレイなどの機材も自分専用機が調達でき、回線上もどこからでもアクセスできるのであれば学生は自宅にいてもよいことになる。また、教員も同様な環境を整備すれば、特定の教室や教卓に縛られる必要がなくなる。研究出張中の教員が、その現場から学生に教育を行えるのであればこれほど価値の高い授業はないであろう[1]。学生にとってはいわゆる高度なオンライン教育である一方、大学教員の方向からみるとこれはまさに大学教員のテレワークである。義務教育の学校における働き方改革は様々な問題（ブラック企業化など）が露呈してやっと始まったが、大学教員の働き方も実際は非常に問題山積であるはずにも関わらず、その働き方改革は現状ではほぼ全く論じられていない状況のように考えられる。本研究ではこれらを提案することも重要だと考えている。

## 3 VR とロボット

VR とロボットと言うと、目新しいテーマを繋いだけと見えるかもしれない。しかし、この二つは端緒から連携している。たとえばロボットから考えてみよう。これが最も必要とされる場面は何か。その一つに、災害時、人間が直に入っていく場所（崩落の危険、災や煙の中、放射の汚染等）での活動である。ここに自走、自立歩行型ロボットの重要性がある。歩行型で災害現場奥ま

で入って行った時、では人間はどこでそのロボットを操作するのかと言えば、安全性を確保した場所から遠隔操作を行うのである。その操作の際の必要情報は何か。まず映像と音像である。しかし、例えば人が下敷きになっているかもしれない障害物を慎重に動かす場合や、普段は人が直に操作している機器を掴んだり取りまわしたりする場合は、その力加減をどう伝えたらよいであろうか。具体的には、今どの程度の力が働いているかをどうフィードバックすればよいであろうか。すなわち、ロボットのセンサーで感じた内容を人にフィードバックする方法は、まさにVRによるものには他ならない。

#### 4 VRとAI

VRとAIと言うと、目新しいテーマを繋いだだけに見えるかもしれない。しかし、この二つは間違いなく連携が必要である。たとえばこのようなVR教育を考え見よう。VRは体験学習に優れているので、その一つに語学の習得が挙げられる。そして語学の重要な習得方法の一つは留学であるが、これは言わば、その言語の国の環境の中に身を置き、実際に体験する重要性を意味する。しかしこれには費用的・時間的コストが非常にかかるので、VRで行う意義が十二分にある。しかしここで、言語応答のやり取りがゲームやクイズのようになっては意味がない。留学の真の意味は、ゲームやクイズではない、イレギュラーでリアルなやり取りが行われることであるので、VRで語学留学を行う際も、本当の人間とのやり取りや存在感に限りなく近い空間を構築することこそが最重要ポイントになる。すなわち、本当の人間とのやり取りに限りなく近づけるためには、AIを組み込んだVR空間が必要になる。なお、後者の、本当の人間の存在感に限りなく近づけるために、VRにおいて心理学研究が必須となる。

#### 5 今後の課題

VRと教育、テレワーク、ロボット、AIの連携は、実はVR研究を中核とした研究領域の広がり・創生のごく一部である。たとえば、ITC教育を進める際に、普通教室からパソコン教室(端末室)へ改良工事が必要であったのと同様に、本研究で意図するような授業を可能とするためには、机

やイス自体も含めた、教室のつくりを変えていく必要がある[1][2]。PCで作業をするための什器やPCの配置、また着座姿勢や環境(光や音)などの適切性が研究対象となるように、今後はVR利用環境の検討自体も一つの研究領域をなすことが推測される[1][2]。

また、大学教員のテレワークに関連して述べた点にとどまらず、大学の施策の面からもVR研究の重要性は指摘できる。すなわち、パンデミックに対する大学のBCPとしてである。大学教育の中には、教育や授業を行うだけでなく試験の実施もある。たとえば、VRとロボットを連携させれば、いわゆる持ち込み不可型の定期試験も公正に行う施策も提案することができる。このように、VRを中核とする研究は、教育、テレワーク、ロボット、AI等とだけでなく、コロナ後の大学施策の方略とも、ダイナミックに連携させていく意義がある。

謝辞：本研究の一部は、国立研究開発法人情報通信研究機構の高度・放送研究開発委託研究／“革新的な三次元映像による超臨場感コミュニケーション技術”の研究開発の一環としてなされたものである。また一部は、立正大学研究推進・地域連携センター支援費(第1種)の一環としてなされたものである。

#### 参考文献

- [1] 櫻井広幸, 「魔法教室」の構想, 日本テレワーク学会誌, 16(1), 32-34, 2018.
- [2] 櫻井広幸, 大学におけるバーチャルリアリティを用いた教育およびテレワークの構想, 第4回国際ICT利用研究会全国大会講演論文集(CD-ROM), 2019.
- [3] 櫻井広幸, バーチャルリアリティとICT教育, 立正大学情報メディアセンター年報, 1, 47-55, 2011.

# ブロックチェーンの仕組みと課題

鎌田光宣

千葉商科大学人間社会学部

kamata@cuc.ac.jp

**要旨** ブロックチェーン技術の仕組みを活用したものの代表がビットコインやイーサリアムといった仮想通貨（暗号資産）である。このお金はどこかの企業が発行したものではなく、管理している明確な組織がないにもかかわらず、しっかりと信用された取引を行うことができるのである。本稿ではブロックチェーンの概要、歴史、課題について説明したあと、ブロックチェーンの今後について考察する。

**キーワード** ビットコイン ブロックチェーン 歴史 分散台帳システム

## 1 ブロックチェーンの概要

ブロックチェーンでは、ある大きさのデータの塊（ブロック）がいくつもチェーン状に繋がっている。物理的なチェーンが存在するわけではなく、ブロックのヘッダー部に、直前のブロックがどれか分かるような情報が記載されている。ここに暗号技術や分散システムの技術を取り入れることで、一度書き込まれたデータは、たとえシステム管理者であってもあとから改竄（かいざん）することはできないシステムとなる。

ブロックチェーンの特徴を簡単にまとめると、次のようになる。[1]

- 分散型データベースである
- 非中央集権型である
- 書き込み専用・改ざん困難である

ブロックチェーンは、多数のノードが同じデータを共有している。つまりデータを冗長化して保存して稼働しているため、どこかのノードやネットワークに障害が発生しても動き続けることができるシステムである。また、改竄の検出が容易であるという特徴を持つ。

非中央集権型というのは、中央の管理者がいない状態のことを指す。特定役割を持つサーバーが中央にあるわけではなく、大量のユーザーがブロックチェーンを共有する状態である。中央集権型とは違い、管理者が独裁的に管理するということができないため、利用者は信頼できない管理者の存在を気にしなくて良い。お互いに信用できない者どうしであっても、不正があることを心配することなく安心して取引を行える仕組みとなっている。

ブロックチェーンの中を見ると、1つのブロックには多数のトランザクションが含まれており、その中には、どの口座からどの口座にいくら送るか、という情報が書かれている。

ブロックチェーン自体は残高を管理しておらず、利用者が誰なのか、残高はいくらなのか記載されていない。利用者が自分の残高を知りたいときは、自分のウォレットアドレスに関する取引データを集めて、未使用分の残高である UTXO(Unspent Transaction Output) を求める必要がある。ある金額を相手に送りたいときは、送金するのに十分な残高のある UTXO を入力側に置き、送金したい相手のウォレットアドレスと金額を出力側に置く。入力側、出力側ともに複数の宛先を指定することができ、UTXO と送金額の差額は自分のウォレットアドレス宛に送るよう出力側に配置する。ブロックチェーンには、この取引の記録が延々と綴られているのである。

## 2 マイニング

ブロックチェーンでは、マイナー（採掘者）と呼ばれるノードがトランザクションを集めてブロックに詰め、それをブロックチェーンに追加する、マイニングと呼ばれる作業を行っている。マイナーは、ブロック追加の成功報酬と送金の手数料を手に入れることができる。

ビットコインのトランザクションの中身を見ると、必ずしも送る側からの入力と受ける側への出力が一致しておらず、その差額は手数料としてマイナーに支払われる報酬になる。この手数料の金額はトランザクションを作る人、すなわち送金する人が自由に決めて良いことになっている。手数料を0にしたり、安く設定したりすることもできるが、手数料が安いと、トランザクションをブロッ

クチェーンに取り込んでもらえない（いつまでたっても取引が承認されない）可能性が高くなる。

ブロックチェーンでは、保存されているデータの耐改竄性を担保する仕組みとしてハッシュ値が用いられている。ブロックの中には多数のトランザクションが格納され、トランザクションの中身には、送信者の電子署名と公開鍵、送金する金額、送金先のウォレットアドレスなどが含まれる。これらのトランザクションに加え、前のブロックのハッシュ値と、さらにノンス（Nonce）と呼ばれる値を加えてハッシュ値を求める。これらはすべてブロックの中に取り込まれ、永久に保管されることになる。途中のデータの改竄を行おうとしても、1ビットでもデータが変わるとそのブロック以降に含まれるハッシュ値がすべて不整合を起こすことになるため、不正を行うことは事実上不可能である。

ビットコインでは、ハッシュ値の先頭が所定数の「0」の並びになるノンスを求める作業を各ノードで行っている。ノンスを変えて、何度も何度もハッシュ値を求め、ハッシュ値の先頭が所定数の「0」の並びになるノンスを発見すると、ブロックを追加し、コインを新規発行することができる仕組みになっている。この作業はマイニング（採掘）と呼ばれ、時間と電力をかけて行っている処理の正体である。

### 3 ブロックチェーンの歴史

ブロックチェーンの元となったアイデアは、1991年のStuart HaberとW. Scott Stornettaの研究[3]に遡る。デジタル文書にタイムスタンプを付けることによって、日付が遡ったり、改竄されたりを防ぐというものである。このシステムでは、暗号化され、鎖のように繋がったブロックを使用してタイムスタンプの付いた文書を保存し、複数の文書を1つのブロックにまとめることが行われた。

1999年のMarkus JakobssonとAri Juelsの論文にPoW(Proof of Work)という言葉が登場した[4]。サービスを受ける側に、コンピューターによる処理時間を要求することで、DoS攻撃や迷惑メールの送信を抑えようとする仕組みである。2004年にはHal Finneyがリユース可能なProof of Work(RPoW)と呼ばれるシステムを発表した。Webサイトのユーザーが使用したPoWトークンを、新しい未使用のRPoWトークンと交換でき、その後同様にRPoWトークンの受け入れ態勢が整っている第三者のウェブサイトで使用できるというものである。

ビットコインの始まりは2008年にSatoshi Nakamotoと名乗る人物が発表した論文[5]である。最大の特徴は、

中心となる運営者が存在しないということである。

その後、ブロックチェーンにさまざまな機能を持たせて仮想通貨以外の分野にも応用しようとする動きが起こる。イーサリアムは、ループ処理などを含む複雑なアプリケーションを開発し、実行できる機能を実現した。さらに、イーサリアムの機能を利用することで、イーサリアムのブロックチェーン上でオリジナルの仮想通貨を簡単に作成・発行できるようになった。これにより、新たな仮想通貨が爆発的に増えることになった。

## 4 ブロックチェーンの課題

### 4.1 安全性

暗号資産はウォレットの秘密鍵がなければ勝手に使われることはない。しかしながら、秘密鍵が盗難されると全て犯罪者の手に渡ってしまう。ウェブウォレットや取引所のウォレットを利用している場合、秘密鍵がサービス事業者側に知られていることになるため、自分がいくら気を付けていても事業者側で盗難される可能性がゼロではない。実際に、2014年に起こったマウントゴックス事件では、利用者に落ち度がなく、事業者の内部でビットコインが盗まれた。

巨大なマイニングプールの存在も安全性に影を落としている。ビットコインをはじめメジャーなブロックチェーンにおいて、個人がマイニング競争に勝つことは事実上不可能である。どこかのマイニングプールに参加して、チームの一員として報酬を得るしかない。マイニングプールとは、マイナーを集めた巨大なマイナー集団で、マイニングが成功して報酬が得られると、その貢献度に応じて各マイナーに報酬が分配される仕組みになっている。このマイニングプールが巨大になったため、現在のビットコインではいくつかのマイニングプールが談合すれば、不正なブロックを故意に追加することも不可能ではない状態になっている。

### 4.2 スケーラビリティ

多数のユーザーが一度にブロックチェーンを利用すると、ブロックチェーンの処理能力が追い付かなくなることがある。トランザクションの遅延が発生すると、ユーザーはより早くトランザクションを処理してもらうために手数料を高く設定するという状況が起こり、手数料が高騰してゆく。ブロックチェーンでは、システムの処理量が増加したときにノードを増やしても、1つのノードが処理するタスク量は減らないため、システム全体の処

理能力は変わらないままである。

### 4.3 ダブルスペント（二重送金）問題

ビットコインのブロック追加速度はおおよそ10分間隔で、さらに確定するには6ブロックほど待つ必要がある。しかしながら、そこまで顧客を店舗で待たせるわけにはいかず、トランザクションが承認されていないことを承知で顧客にモノを売ることになる。この運用を「ゼロ・コンファメーション（0-comfirmation）」と呼ぶ。

ビットコインでは、手数料が低く設定されたトランザクションは、いつまでたってもブロックチェーンに取り込まれない、という事態が起こる。そのような事態への救済措置として、「RBF」（Replace by Fee）という仕組みが導入された。後から手数料を高く設定しなおしたトランザクションを送ることで、先に送ったトランザクションを取り消せる、というものである。しかし、このRBFを導入したせいで、ビットコインは意図的なダブルスペント（二重送金）ができてしまう状況になってしまった。

### 4.4 消費電力

PoWの仕組みを取り入れているブロックチェーンでは、マイナーはチェーンに新しいブロックを追加して報酬を得るため、無意味ともいえる計算で莫大な電力を浪費している。

ビットコインのマイニングに使われる消費電力が、世界の電力消費の0.25%に達するというデータもある。SDGsの面でも、例えばPoWをPoSに変えて消費電力を減らすなどの対策が求められる。

## 5 ブロックチェーンの今後

即応が必要なものは、いまのところブロックチェーンに向いていない。ブロックチェーン上で動く分散アプリケーション（DApps）は、いちどブロックチェーンに載せてしまうと変更ができないため、頻繁にシステムに変更を加える用途には向かない。また、秘匿性の高い情報については、たとえそれを暗号化したとしても、パブリックに流すというのは不要なリスクを生むことになる。容量の大きなものの流通にも向いていない。

ブロックチェーンの優れたところは、非中央集権型であり、データの改竄ができず、参加者がみんな検証できるというものである。信頼していない者同士で構成されたネットワークで、不正があることを心配することなく安心して取引を行えるのである。

法定通貨の信用度が低い地域、例えば偽札が多く出回っていたり、通貨の価値が乱高下したりするようなところでは、ブロックチェーンによる暗号資産の評価が高い。現状では日常の決済に向いているとは言い難く、決済の高速化を図る仕組みの開発が期待される。

電子投票システムには、ブロックチェーンの耐改竄性がぴったりと当てはまる。いつ、誰が、何に投票したのかをブロックチェーンに記録することで、不正の余地をゼロに近づけることができる。ただし、本人確認をバイオメトリクス認証などを用いて厳格に行う必要があり、また、「誰が、何に」の情報は、個人のプライバシーとして保護されるべきものであるため、記録を検証できる人物や団体は限定される可能性がある。

電子データ化された証憑書類の保存の用途にもブロックチェーンは有効である。現在利用されている電子署名や電子証明書は、それを認証する認証局自体の信頼性や安全性が問われるが、例えばハッシュ値や電子署名をブロックチェーンに記録しておけば、書類自体はブロックチェーンの外にあっても改竄の有無を確認することができる。

ブロックチェーンは新しい技術や仕組みが取り入れられ日進月歩で進化を続けているが、万能ではない。ブロックチェーンが得意とする分野とそうでない分野をしっかりと把握した上で、活用方法を模索していく必要がある。

## 参考文献

- [1] 鎌田光宣, “ブロックチェーン技術の歴史と展望”, 国府台経済研究, 31巻2号（掲載予定）
- [2] 「ブロックチェーンの定義」を公開しました, <https://jba-web.jp/news/642>, 日本ブロックチェーン協会（2021年1月5日参照）
- [3] Stuart Haber & W. Scott Stornetta, “How to timestamp a digital document”, Journal of Cryptology volume 3, pages 99–111 (1991)
- [4] Jakobsson, Markus; Juels, Ari. “Proofs of Work and Bread Pudding Protocols”. Secure Information Networks: Communications and Multimedia Security. Kluwer Academic Publishers: 258–272. (1999)
- [5] Nakamoto, Satoshi, “Bitcoin: A Peer-to-Peer Electronic Cash System”, <https://bitcoin.org/bitcoin.pdf> (24 May 2009) (2021年1月10日閲覧)

# テレビドラマ視聴率と SNS を活用したプロモーション戦略の関係性

佐原 麻衣子

日本大学 法学部

キーワード： 視聴率、SNS、プロモーション

## 1 はじめに

スマートフォンや SNS の普及に伴い、テレビを視聴するという日常の生活様式にも変化が生じているのではないかと感じる。Twitter のハッシュタグ機能で、その瞬間に話題になったことがリアルタイムでトレンド入りをすることでさらに注目を浴びニュースになるなど SNS を通じて新しいコミュニケーション形態が形成されている。

近年では、Twitter や Instagram にテレビドラマの公式アカウントを設けて番宣を行っていることが多い。また、高視聴率を記録しているものは Twitter でのつぶやき回数が多く、トレンド入りをして話題になっている。そこで高視聴率を獲得し、話題となったテレビドラマには SNS での話題性に影響があったのか、その原因を調べていく。そこで Twitter で世界トレンド 1 位を獲得し話題になった二つのドラマを中心にみていく。1 年では異例の 2 クール連続で放送され、最終回に近づくに連れ、ドラマの意外な展開や考察で様々な意見が飛び交った「あなたの番です」と令和に入って放送されたドラマで最高となる平均世帯視聴率 32.7% を最終回で記録した<sup>1</sup>「半沢直樹」を中心に視聴率と SNS の関係を調べ、アンケート調査を行い、SNS ユー

ザーの動向を調査し今後の視聴率とプロモーション効果を考察する。

## 2 研究の背景

### 2.1 視聴率とは

視聴率とは、テレビ番組や CM が「どのくらいの世帯・人々に見られたか」という視聴量の大きさを表す指標の一つである。視聴率には、「世帯視聴率」と「個人視聴率」の二種類があり、「世帯視聴率」は、テレビ所有世帯のうち何世帯でテレビをつけていたかを示す割合である。一般的に視聴率として使われるものはこの「世帯視聴率」を指す。また、「個人視聴率」は誰がどのくらいテレビを視聴したかを示す割合であり、視聴者を性・年齢や職業などの特性で分けし、区分ごとに視聴率を計算している。[1] なお、本稿では「世帯視聴率」を扱う。

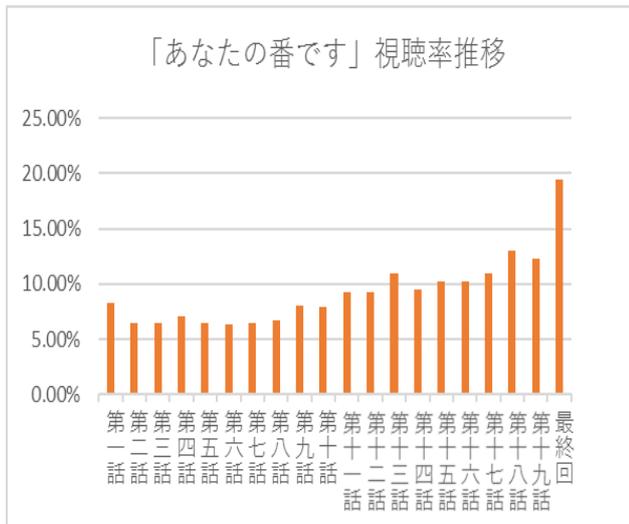
### 2.2 研究対象ドラマの視聴率推移

「あなたの番です」の第 1 話から最終回までの視聴率は、第 1 話から第 10 話までの第 1 章はほぼ横ばいで高いといえるものではないが、第 2 章から最終回に向けて上昇していき、第 19 話から最終回に関しては、約 7% も上がった。(表 1)<sup>2</sup>

<sup>1</sup> 新市場を創る人のデジタル戦略メディア日経 TREND, <https://xtrend.nikkei.com/atcl/contents/watch/0013/01183>, (2020 年 12 月 28 日閲覧)

<sup>2</sup> Drama & Movie, <https://www.oricon.co.jp/news/2144003/full>, (12 月 26 日閲覧)

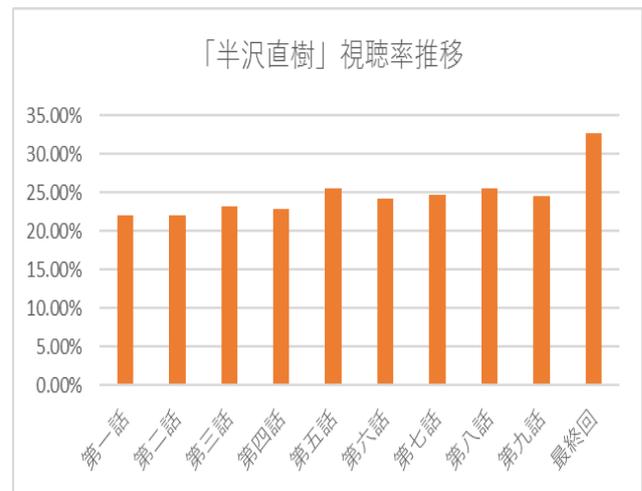
表1 「あなたの番です」視聴率推移



ドラマの視聴率では12%程度を獲得するとそのクール内でトップ3に入ると言われている中で、最終回が19.4%なのだから、高視聴率と言えるだろう。[2]この背景には、第二章に突入してから、Twitterで真犯人の予測に関するツイートで大いに盛り上がったことも要因の一つとして考えられるだろう。

一方、半沢直樹2020年版では全話を通して22%を上回り、最終回では32.7%を記録した<sup>3</sup>。(表2)2013年版の記録を上回ることではできなかったが、Twitter上では、全話を通して世界トレンド1位となるほどの反響だった。[3]

表2 「半沢直樹」視聴率推移



### 2.3 テレビドラマ事例

現在、ほとんどのテレビドラマではSNSで公式アカウントを設けて番宣を行い、そのプロモーション効果に期待をしている。

「あなたの番です」では、最終回放送までにそれぞれInstagramでは575件<sup>4</sup>、Twitterでは1158ツイートもの投稿がされている。<sup>5</sup>

「半沢直樹」では、最終回放送までに、Instagramで160件<sup>6</sup>、Twitterで711件の投稿がされた<sup>7</sup>。「あなたの番です」に比べると少ないと感じるが、「半沢直樹」では、2013年の前作の時はまだTwitterが浸透しておらず、Facebookでの盛り上がりの方が強かった。2020年版を放送するにあたって、Facebookのページの更新を復活させ、9万人の登録者に向け、情報を投稿したところ24時間で4.1万人のリーチを獲得した。[4]

<sup>3</sup> 日経×TREND

<https://www.oricon.co.jp/news/2144003/full/>, (12月28日閲覧)

<sup>4</sup> 「あなたの番です」公式Instagram

[https://www.instagram.com/anaban\\_ntv/?hl=ja](https://www.instagram.com/anaban_ntv/?hl=ja), (12月26日閲覧)

<sup>5</sup> 「あなたの番です」公式Twitter

[https://mobile.twitter.com/anaban\\_ntv](https://mobile.twitter.com/anaban_ntv), (12月26日閲覧)

<sup>6</sup> 「半沢直樹」公式Instagram

[https://www.instagram.com/hanzawa\\_naoki\\_tbs/?hl=ja](https://www.instagram.com/hanzawa_naoki_tbs/?hl=ja), (12月26日閲覧)

<sup>7</sup> 「半沢直樹」公式Twitter

[https://mobile.twitter.com/hanzawa\\_naoki](https://mobile.twitter.com/hanzawa_naoki), (12月26日閲覧)

### 3 アンケート調査

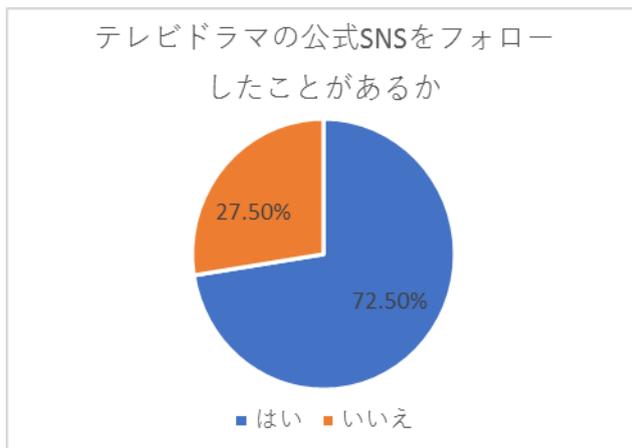
#### 3.1 アンケート目的及び対象

本調査においては10代から20代の男女51名に「テレビドラマを視聴するときその番組のSNSをフォローしたことはあるか」、「テレビを視聴しながらその番組についてSNSでつぶやいたことはあるか」、「SNSで話題になっていたことで後からドラマを視聴したことはあるか」、「『あなたの番です』の考察予想をTwitterにつぶやいたことはあるか」、「『半沢直樹』は視聴していなかったが、Twitterやテレビ広告で見て、“倍返しだ！”や“おしまいDEATH”などのフレーズを知っているか」、「他人のSNSでの情報から興味を持って番組を視聴したことはあるか」の6点についてアンケート調査を実施した。これにより番組のSNS公式アカウントの有用性とTwitterのつぶやきの影響力を検証する。なお、本調査は2021年1月4日に行ったものである。

#### 3.2 アンケート結果

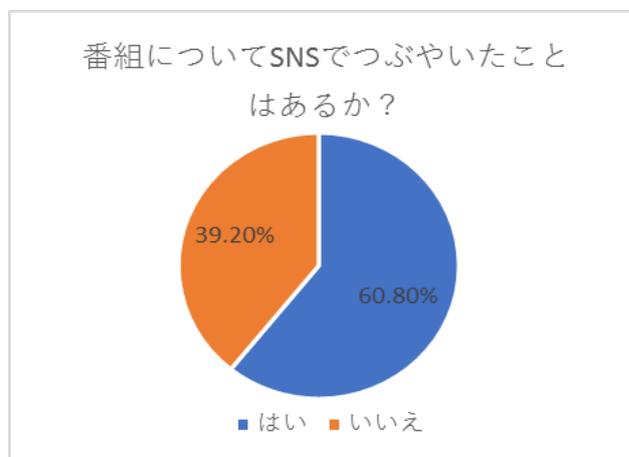
アンケート結果については「テレビドラマを視聴するときその番組のSNSをフォローしたことがあるか」という質問に対して、フォローしたことがある人が72.5%という結果となった。(表3)

表3 テレビドラマの公式SNSをフォローしたことがあるか



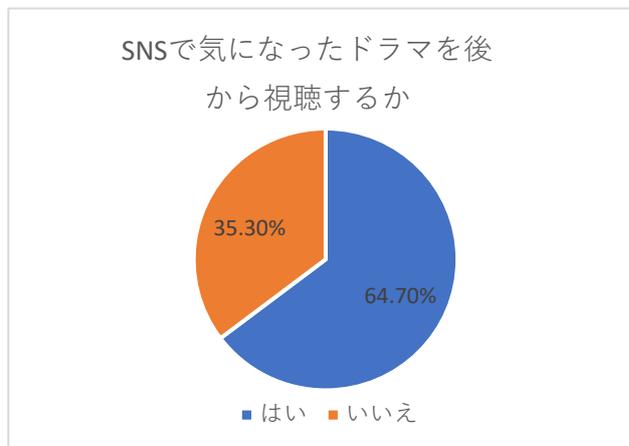
やはり、番組SNSには裏情報やオフショットが多く掲載されていることから注目する人が多いことが分かる。また、Instagramのストーリー機能で放送開始前に告知するのもフォロワーに再度知らせることができるため有効である。「SNSで番組についてつぶやいたことがあるか」という質問には60.8%の人がつぶやいたことがあると答えた。(表4)

表4 SNSで番組についてつぶやいたことはあるか



また、64.7%の人がSNSで話題になったものを後から視聴したことがあると答えたことから、SNSの影響力の大きさが伺える。(表5)

表5 SNSで気になったドラマを後から視聴するか



また、「あなたの番です」では視聴者は全体の49%だったにも関わらず、考察予想をつぶやいたり、見たことがあると答えた人は、55.6%を占めた。番組公式 SNS とはあまり関係がないが、考察予想で Twitter が盛り上がったことも視聴率向上の一つの要因だと考えられる。番組の公式 Instagram に挙げられるキャスト陣による告知にヒントが隠されているような演出からも視聴者を楽しませていることが分かる。

「半沢直樹」は視聴していた人は全体の 37.3%と少なかった。しかし、一度も視聴したことがないという人でも、「倍返しだ!」や「おしまい DEATH」などのフレーズだけは知っていると答えた人が98%と高い数字が出た。また、74.5%の人が他人の SNS での情報から興味を持って番組を視聴したことがあるという結果になった。

#### 4 考察

今回の調査から私が予想していたよりも SNS をチェックしている人は多いと感じたため、テレビドラマの視聴率向上には SNS を用いることによって、多くの人に周知させることができ、その特性を生かせば拡散力も高いことから SNS を利用するプロモーション効果は高いと感じた。実際にテレビ番組だけでなく、企業の商品 PR としても大きな功績を残している点も踏まえ、これからの期待値も高いと考えられる。また、SNS で話題性を呼ぶことで相乗効果を得られるのではないかと考えた。多くの人が SNS で話題となっていることを知り、一定の知名度や人気が出ると人気を呼び寄せるのではないだろうか。Twitter 登録者の投稿頻度は、ほぼ毎日から月に 1 度くらいまでの人が56%、反対に 1 度も投稿したことがない人と最近 1 か月投稿したことがない人合わせて 44%であっ

た。<sup>8</sup>

この数字から、登録者の中でもつぶやく人とただ閲覧のみをしている人に分けられるということが考えられる。私が調査したアンケートでも他人の SNS の情報から番組を視聴したことがあると答えた人が 74.5%いたことから、投稿者は活発的に影響を与えていることもわかる。今や、テレビを視聴することと SNS への投稿を閲覧することが同じ時間軸で楽しまれているのではないだろうか。

また、SNS にもそれぞれ特徴があり、プロモーションの仕方によって使い分けをしていくことがよりよい効果をもたらしていくと考えた。今回の調査で、番組公式 SNS を活発にさせて、話題性を生む方法もあれば、Twitter のトレンド入りをすることによって認知度が上昇する場合もあるからだ。現代は SNS 社会だからこそ話題性があるものに人々は注目しやすいため、話題性に富んだことをどんどん提供しなければ時代の流れについていけないのではないかと思う。SNS でも新しい仕組みがどんどん普及されている現代に合わせたプロモーション効果に今後も期待したい。

#### 参考文献

- [1] FELLOWS Creator's Agent  
[https://job.fellow-s.co.jp/useful\\_info/feature\\_detail/Art-0467](https://job.fellow-s.co.jp/useful_info/feature_detail/Art-0467), (2020 年 12 月 28 日閲覧)
- [2] NIKKEI STYLE  
<https://style.nikkei.com/article/DGXMZO99819520Z10C16A4000000/>, (2020 年 12 月 28 日閲覧)
- [3] 日経×TREND  
<https://www.oricon.co.jp/news/2144003/full/>, (2020 年 12 月 28 日閲覧)
- [4] Screens ,<https://www.screens-lab.jp/article/22933>,(2020 年 12 月 28 日閲覧)

---

<sup>8</sup> 金 正則「ツイッター社会論」P65 (2021 年 1 月 5 日閲覧)

# ポストコロナにおける情報教育の役割

木川明彦

立正大学 非常勤講師

キーワード：オンライン授業/ blended learning /リテラシー/データサイエンス

## 1. はじめに

新型コロナウイルス(COVID-19)感染症の影響を受け、教育機関においてはオンライン授業の実施が進められている。こうした中、令和三年度以降においてもオンライン授業は続いていくことが予想される。昨今、早いもので学術の世界においてもオンライン授業に関する取り組みや成果を問う研究発表を目にする機会が増えてきた。そこで、一年という区切りをもって、検証に当たるのもよいと考え本稿をまとめるものとする。

## 2. 問題の所在

一言にオンライン授業といっても、同期型や非同期型、ハイフレックスと複数パターンが存在する。また、情報教育の役割といっても、扱う内容が基礎教育なのか専門教育なのかというカリキュラム上の扱いによって抱えている問題が違う。

ここで明確にしておくが、本稿で考察していきたいのは、今後多くのオンライン授業が実施される中で情報教育がどのような役割を果たすか(または、果たさなければならないのか)といった事柄を論じることであり、ポストコロナにおける情報教育の在り方とその展開可能性についての考察であ

る。

そこで、オンライン授業の基本的な扱いを整理したい。本稿では、オンライン授業を手法と捉え、その役割を担うのは情報教育であると考え考察を進める。言い換えるならば、他の学問分野から見たらオンライン授業に関することは情報教育の分野でしっかり学んでおいてほしいという一種の要請を感じるわけである。また、他の学問分野から見れば、情報教育は漠然と「情報機器を扱う授業」といったイメージで語られているところもあり、各大学の情報センターと学部系統の講義を区分しているかは定かではない。そして当然にして門外漢というわけにもいかない立ち位置である。

## 3. オンライン授業の捉え方

まず考えたいのは、オンライン授業が非常時の手段かということである。オンライン授業の導入は以前から実施していない(または構想のなかった)大学からしたら、非常時の手段であったということは否めない。多くの大学で緊急対策を余儀なくされた。具体的には、承知の通り、音声付きパワーポイントで講義配信をしたり、Zoom や Microsoft Teams といったアプリケーションを使用し同期型講義をしたりした経緯がある。情報教育に携わる者ならば、その導入

は難しいものでなかったと思うが、他分野の教員は苦勞された部分はあると感じられる。こうした中、これらオンライン授業の取り組みは、兎にも角にも多くの者の関心を情報教育に向けさせ、その学習の機会や可能性を感じさせた。また、話の発端は違うが、内閣府(2020)の報告<sup>1</sup>の中にも「STREM<sup>2</sup>人材の育成に向けて、教育・研究環境のデジタル化・リモート化、研究施設の整備、国内外の大学や企業とも連携した遠隔・オンライン教育を推進するとともに、データサイエンス教育や統計学に関する専門教員の早期育成体制等を整備する。」という記載があり、こうした要請は情報教育にとって好機であると考えられる<sup>3</sup>。

今後は、俗にポストコロナやウィズコロナと呼ばれる時代がやってくる。そこで、「いつでも・どこでも」取り組める学習パッケージが必要になってくると考えられる(“どこでも”といった考え方には今まで否定的であったが、今後の状況を考えるに“どこでも”の視点も考慮しなければならない)。こうした中、今後のオンライン授業を今までの延長線上にととらえて本当に良いのであろうか、今後はよりポジティブな手法としてとらえるべきと感じている。

以降では、こうした期待を受け、情報教育がオンライン授業とどのように向き合うか考えていきたい。

#### 4. ある講義の実践

ここでは、私が令和二年度後期に実施したある大学での講義状況をまとめる。紹介も込めて、下記にまとめる(表1)。

この講義では、社会科学分野の学生を対象に全13回(2回の補修)をもって、主としてMicrosoft Excelの学習を進めてきたものである。講義内容自体は、一年生に対する情報リテラシー教育である。

(表1：講義の実施計画)

授業回	指導内容
第1回	オリエンテーション・受講確認
第2回	イントロダクション・表の作成
第3回	計算の基礎・算術演算子・セルの参照・時刻を使用した計算
第4回	関数の利用①(オートサム機能)
第5回	グラフの作成(主として棒グラフ)
第6回	グラフ作成の続き(pp45-48)
第7回	RANK.EQ 関数・IF 関数
第8回	IF 関数の続き(応用)・ネスト構造
第9回	COUNT に関する関数
第10回	VLOOKUP 関数の操作・中間まとめ問題の実施
第11回	HLOOKUP 関数・SUMIF 関数
第12回	データベースの操作・並び替えと抽出・クロス集計
第13回	秋学期の復習・期末総まとめの解説

<sup>1</sup> 内閣府(2020)『経済財政運営と改革の基本方針2020～危機の克服、そして新しい未来へ～』p28参照。

<sup>2</sup> “Science, Technology, Engineering, Art,

Mathematics”の5つの領域の頭文字をとった標語。理数教育に創造性教育を加えた教育理念。

<sup>3</sup> 個人的には、モチベーションを高めているわけであって、対面講義のほうが講義導入は早い気がする。

#### 4-1. ある講義の感想

情報処理基礎として講義を展開したが、多く感じたのは、このままオンライン授業を展開してしまって良いのかという疑問である。受講生の中には、パソコンに触れたことのないと言い張る学生もあり、本当なのかと思わされることもあったし、学習環境がよく把握できず、中々学習に移行できない者もいた。また、多く問い合わせを受けたのは、授業内容でなく、データの管理、または使用する LMS(Learning Management System)の使用に関してある。更に、課題の添付間違い・添付漏れといった事象も見られた。これらの事象はどの大学においても生じたが、押しなべて言えるのは質問者に偏りがみられたということである。なお、講義計画はある程度決まったものとなっていたため踏み込んだ内容にはなっていない。

成績分布は、A・B 群と D 群に偏りが見られる(表 2)。こうした中、単位不認定の一番の要因は「授業参加の低さ(脱落)」であった。傾向としては、最初からあきらめてしまっている学生もいたが、初回だけ参加しているとか徐々に欠席率が高くなってきたという学生もいた。その要因は学生のモチベーションや学習修練の差があると考えられるが、その理由は複合的であると推察される。

(表 2：最終評価の取りまとめ)(単位：人)

	A クラス	B クラス	合 算	割合 (A-D)
--	----------	----------	--------	-------------

A	9	7	16	26%
B	11	3	14	23%
C	5	5	10	16%
D	14	7	21	34%
合計 (A-D)	39	22	61	100%
F	12	10	22	
参加不良	うち 8	うち 8		

#### 5. データサイエンスとの関係性

こうした中、データサイエンスとの関係性を考えていきたい。内閣府(2019)『AI 戦略 2019』でうたわれるよう、これからのデータサイエンス教育への期待は大きい<sup>4</sup>。このことは、情報教育にかかわらず、全大学教育の課題となっている。その機運は、オンライン授業実施によってますます加速している。また、STEAM 教育は、文理融合とか各分野の横断的な学習の標語となっている。

当然、専門の学部系統であれば、踏み込んだ内容も扱えるだろうが、現状の他分野からしたら、上記のような例もあり、どの程度の実践ができるかは考察が必要になってくる。

#### 6. 今後の情報教育

以上のことから、現在の情報教育が他分野のニーズを充足できているかという点と少々不安を感じる。社会的なニーズとしてデータサイエンスや AI 教育が求められる中、これらの教育はオンライン授業に関係

<sup>4</sup> 具体目標①として「文理を問わず、全ての大学・高専生(約 50 万人卒/年)が、課程にて初

級レベルの数理・データサイエンス・AI を習得」と掲げている(p12 を参照)。

なく実施されるわけであり、問題の根幹は本題と異なる。悩ましいのは、先端的な教育をオンライン授業として実施しないといかない一方で他の分野からのニーズを満たさなければならないということである。つまり、一部の大学においては、目新しいことに目が行っても、それが難しいということである。やはり、根幹を為すは昔から言われる情報リテラシーであるわけで、学習者のデータがどこにあるのか、どのように提出するのかといった根本的操作の学習になってくる。また、コロナ禍にあって学習環境に統一がみられない中で、クラウドやサーバー、ネットワークに関する知識も押さえておきたい。言い換えるならば、オンライン授業禍における情報リテラシーの育成といったところであり、加えて、既存の取り扱い内容も考えていかなければならない。

当然、データサイエンスをはじめとする学習、データやICTの利活用は重要な单元である。昨今のオンライン授業の影響を受けこれらに触れる機会は確実に増えている。これは大変良いことである。大学教育は他の組織体に比べ膨大なミッションを抱えているのは事実であり、体系的なカリキュラム編成の中ですべて行うことは難しい。従って、現状何が間違っているということもない。しかしながら、本稿でまとめたよう状況があるということを踏まえ、今後の情報教育というものを考えていかなければならない。

## 【参考文献】

- [1]. 内閣府(2019)『AI戦略2019』〈参照日：2021年3月17日〉  
<https://www.kantei.go.jp/jp/singi/tougou-innovation/pdf/aisenryaku2019.pdf>.
- [2]. 松原憲治(2020)「資質・能力の育成を重視する教科等横断的な学びとSTEM/STEAM教育の意義と課題」『教育展望6月号』(教育調査研究所,pp17 - 22).
- [3]. 内閣府(2020)『経済財政運営と改革の基本方針2020～危機の克服、そして新しい未来へ～』〈参照日：2021年3月17日〉  
[https://www5.cao.go.jp/keizai-shimon/kaigi/cabinet/2020/2020\\_basicpolicies\\_ja.pdf](https://www5.cao.go.jp/keizai-shimon/kaigi/cabinet/2020/2020_basicpolicies_ja.pdf).

# 画像を題材とした競技プログラミング用コンテンツの開発

松本 貴裕<sup>†</sup>      福井 昌則<sup>‡</sup>      高見 友幸<sup>†</sup>

<sup>†</sup> 大阪電気通信大学大学院 総合情報学研究科

<sup>‡</sup> 兵庫教育大学

キーワード：競技プログラミング，学習サイト，Python

## 1 はじめに

現在，プログラミング学習やプログラミングスキルの証明，高度な IT 人材を採用する手段として，競技プログラミングが活用されている。競技プログラミングとは，複数の参加者全員に同時に同一課題が複数出題され，その課題を素早く要求に応えられるプログラムの作成することを競うものである。

国内の代表的な競技プログラミングサイトとして，AtCoder[1]や Aizu Online Judge[2]などが挙げられる。また，各サイトはオンラインジャッジシステムが採用されており，ソースコードをアップロードすることでプログラムの実行結果の判定を行なっている。

しかし，競技プログラミングでは，コンピュータサイエンスや数学の問題が出題されており，論理的思考力を測ることができるが，画像処理やデータベース，Web スクレイピングといったエンジニアスキルを網羅的に測ることが出来ない。

また，競技プログラミングは C や C++，Java，Python などといった様々なプログラミング言語で問題を回答することが可能であるが，それにより問題がそれぞれの言語でも解ける問題に限定される。そのため，問題はより抽象的になる。

本研究では，画像を題材とした競技プログラミング用コンテンツの開発を行った。従来のコンピュータサイエンスや数学の問題がこの問題と置き換わる。画像を題材とした問題は競技プログラミングやプログラミング学習サイトでは，取り扱われてないため，現在の競技プログラミングシステムでは扱うことができない。そのため，画像を扱うために新たな競技プログラミングシステムを設計・構築を行った。また，プログラミング言語を

Pythonに限定して，より実践的な問題を構築した。

## 2 Python

Python とは，インタプリタ型言語であり，近年話題の Deep Learning や Internet of Things(IoT)，ビッグデータ処理などの開発を得意としている。特徴として C 言語や C++ などと比べシンプルに記述できることや標準で豊富なライブラリがある。競技プログラミングにも使用されている他，プログラミング学習の入門などといった幅広く用いられており，IEEE の調査[3]によれば，Python は近年最も注目されているプログラミング言語である。

本研究では，Python の豊富なライブラリを活用して画像処理，画像認識，Web スクレイピングのコンテンツを開発した。

## 3 競技プログラミングシステム

### 3.1 システム概要

競技プログラミングシステムは，学習サイト[4]との相互連携を目指すため学習サイトの機能を元に拡張を行った。

ユーザーは始めに自身のアカウントの作成を行いログインする。ログインするとフィールドマップに遷移する。ユーザーはそのページ競技プログラミングが開催される時刻まで待機する。開催時刻となるとコンテスト開催機能によりフィールドマップに競技プログラミング用リンクが表示され，ユーザーはそこから競技プログラミングのページへと遷移する。競技プログラミングページでは問題が一覧になっており，そこから回答する問題を選択していく。問題の回答方法は学習サイトと同様に Web サイト上に構築されたプログラミングエディターを用い，Web サイト上でコーディング

を行い回答する。回答ボタンを押すことで回答のソースコードを提出する。競技プログラミングページではこの工程を繰り返すことにより競技を行っていく。また、競技プログラミングページでは全てのページに常時残り制限時間を表示することでユーザーに制限時間を認識させている。

制限時間が終了した時、コンテスト開催機能によりユーザーは強制的にフィールドマップへと遷移させられる。また、制限時間外の回答を防ぐため回答ページは全てアクセス制限を行っている。

競技プログラミングが終了した時に再度リンクを押すことで採点結果とユーザーの順位が確認できるページへと遷移する。

以上の機能を有する競技プログラミングシステムの構築を行った。システム構成図を図1に示す。

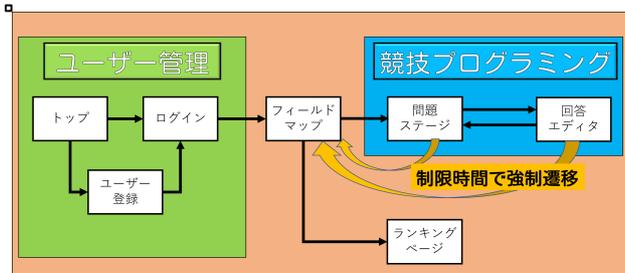


図 1. システム構成図.

### 3.2 コンテスト開催機能

コンテスト開催機能は、競技プログラミングサイトのリンクの表示・非表示と時間外のアクセス制限により実現した。

リンクの表示・非表示は JavaScript を用い実装を行った。Date 関数で開催日時、終了日時を指定し show メソッド、hide メソッドでリンクの表示・非表示を行った。

次に時間外のアクセス制限は Python を用いて実装を行った。datetime モジュールを用い開催時間を指定した。そして時間外にリンクへ接続した場合 Flask の abort 関数を用い HTTP 403 エラーの表示を行った。

### 3.3 制限時間機能

制限時間機能は、競技プログラミングページである問題一覧ページ、問題詳細ページ、回答ページ全てに制限時間の表示を行うものである。

制限時間機能の実装には、JavaScript を用いた。Date 関数を用い現在時刻と終了時間を設定し、制限時間を計算しカウントダウン表示した。制限時間が終了した場合、window.location.href を用い、トップページへと強制遷移させる。これにより制限時間外の接続をアクセス制限と合わせて防ぐ。制限時間機能を図2に示す。



図 2. 問題詳細ページの制限時間表示.

### 3.4 回答機能

学習サイトの回答エディターを改良し回答機能を実装した。Web サイト上のプログラミングエディターは Ace.js の CDN を用いて実装し、実行、回答処理は Python を用いて実装した。また、画像の問題を取り扱うために、ファイル参照ボタンで画像を確認することができる。エディター画面を図3に示す。



図 3. Web 上で動作する回答エディター.

問題に回答するには、実行ボタンでエディター上のソースコードを実行し、プログラムが動作するか確認する。そして回答ボタンでソースコードを提出して問題に回答する。学習サイトでは、不正解の場合は問題に回答することができなかつたが、競技プログラミングサイトでは正解・不正解問わず回答することが可能である。また、回答を行った時、問題一覧のページで回答済みかを確認することができる。問題一覧を図4に示す。



図 4. 問題一覧ページの回答確認.

その他、保存ボタンで現在のソースコードを保存や、リセットボタンで現在のソースコードを初期化することができる。

### 3.5 ランキング機能

ランキング機能では、制限時間終了後に競技プログラミングのサイトで自身のランキングと問題の正解・不正解を確認することが可能である。実装には、NoSQL データベースである TinyDB からユーザーの回答状況、正解状況を元に Python で処理を行い順位と正解・不正解の出力を行った。順位に関しては同率順位の場合も考慮するため pandas モジュールの rank メソッドを用いて出力を行った。ランキング機能を図 5 に示す。

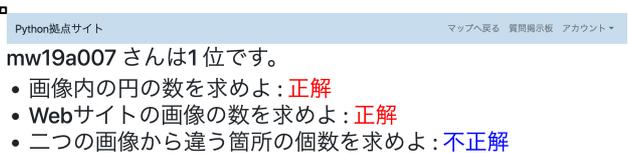


図 5. ランキングページ.

## 4 画像を題材としたコンテンツ

### 4.1 コンテンツ概要

本研究では、他サイトでは扱うことが出来ない画像を題材としたコンテンツを開発した。コンテンツとして、画像処理、画像認識、Web スクレイピングのプログラミングスキルを確認できるものを用意した。また、他サイトでは扱うことが出来ない Python の外部モジュールを用い Python の特徴が出るコンテンツ構成とした。

### 4.2 画像内の円の数を求めよ

この問題は入力値として与えられた円を含んだ画像から円の数を求める問題である。この問題を解くことで画像処理、画像認識のプログラミングスキルを確認することが可能である。入力画像を図 6、解答ソースコードを図 7 に示す。

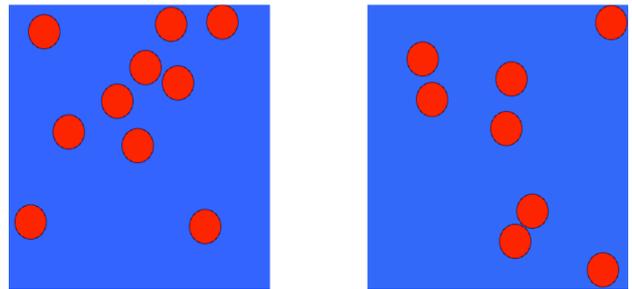


図 6. 左)入力画像, 右)採点時の入力画像.



図 7. 円の数を求めるプログラム.

### 4.3 画像から多種類の図形の数を求めよ

この問題は前問題の応用問題である。前問題では、画像内の円の数を求めるものだったが、この問題は画像内の三角形、四角形、五角形、円の数を求める問題となっている。この問題を解くことでより深い画像処理、画像認識のプログラミングスキルを確認することができる。入力値である多種類の図形を含んだ画像を図 8、解答のソースコードを図 9 に示す。

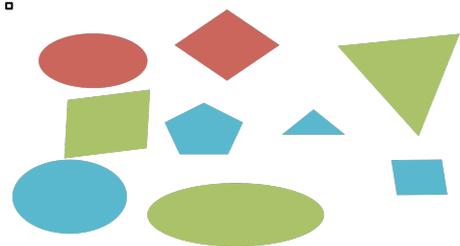


図 8. 多種類の図形を含んだ画像.



図 9. 画像の種類を判別するプログラム.

#### 4.4 Web サイトから白を含む画像の数を求めよ

この問題は入力値として画像を含んだ Web サイトから白色を含む画像の数を求める問題である。Web サイトは画像と文字がランダムに配置されている。この問題を解くことで Web スクレイピング、正規表現、画像処理のプログラミングスキルを確認することが可能である。入力値の Web サイトを図 10、解答のソースコードを図 11 に示す。

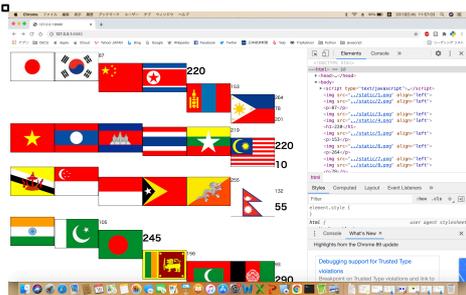


図 10. 入力値の画像を含んだ Web サイト。

```

1 from selenium import webdriver
2 from selenium.webdriver.chrome.options import Options
3 from PIL import Image
4 import re, io, requests
5
6 def getImage(URL):
7     options = Options()
8     options.add_argument('--headless')
9     driver = webdriver.Chrome(options=options)
10    driver.get(URL) # URLのサイトに接続
11    html = driver.page_source # サイトのソースコードを取得
12    ans = re.findall(r'.
- [8] 中村八東, 渡辺稔彦, 田中保史, カワモト・ポーリン Mizar 講義録, 信州大学工学部 情報基礎研究室 <http://markun.cs.shinshu-u.ac.jp/kiso/projects/proofchecker/mizar/Mizar4/index-j.html>

# 保育士養成における相互評価法を活用した工作実践

田中 雅章

Email: m\_tanaka@jc-humanitec.ac.jp

ユマニテク短期大学 幼児保育学科

## ◎Key Words 保育士養成, 工作実践, 相互評価

### 1. はじめに

保育士養成課程において保育者に必要な幼児教育の専門能力の一つとして「個別的・具体的保育知識・実践力」がある。具体的には「ピアノ技術」、「豊かな表現力」、「絵画造形能力」の技術があげられる。本研究では保育士養成科目の一つである「遊び入門」において「豊かな表現力」、「造形能力」を養成する取り組みを行った。「遊び入門」は3人の教員が分担しており、学生の創造力を豊かにする目的で工作を担当した。

### 2. 遊び入門における工作実践



図1 段ボール製の筒

実践した工作の課題は図1の様な長さ34cm直径7cmの段ボール製の筒を材料に使い、学生が自由に3時間で作品にするものである。どうしてもアイデアが創造出来ない学生のために筒を樹木の幹に見立て虫の型紙を準備する作品例を準備した。

作品完成後は学生同士で互いに作品を評価する作品発表会と相互に評価を行った。作品発表会を行った理由は工作で制作しても自分の作品のイメージや想い、世界観を紹介することはあまりない。さらに保育現場では子ども達の自由に表現する工作がこどもの創造力を伸ばす。子ども達の作品とテーマと物語の共感できる共感力を身に着けるためである。

### 3. 作品発表のための視聴覚機器の活用

ICT機器の活用能力は保育士にとって必要な能力のひとつになりつつある。これらの能力は、子どもたちの活動をサポートする際にその威力を発揮するといわれている。ただ、ほとんどの学生はICT機器の活用能力が備わっていない。作品発表会では図2の天井カメラ



図2 天井カメラ

で作品を撮影し、その特徴がよく見えるように2台の大型モニターに拡大表示した。発表する学生

は見やすくなった作品を提示しながら、作品のテーマや世界観、工夫点、留意点などを説明した。視聴する学生は、スマートフォンで作品の評価を行った。翌週に評価結果をまとめた評価シートを発表者へフィードバックした。

### 4. 実践の成果・まとめ

表1 作品発表会の振り返り

| 質問項目                   | 積極群<br>n=28  | 消極群<br>n=28 |
|------------------------|--------------|-------------|
| 次も作品発表会に参加したいですか       | 28<br>100.0% | 0<br>0.0%   |
| 参考になる作品があった            | 27<br>96.4%  | 25<br>89.3% |
| 他人を評価することで得るものがあった     | 25<br>89.3%  | 22<br>78.6% |
| 他人の作品を公平に評価できた         | 24<br>85.7%  | 21<br>75.0% |
| 自分の作品は公平に評価された         | 24<br>85.7%  | 22<br>78.6% |
| 自分の作品を他人に評価してもらって良かった  | 25<br>89.3%  | 15<br>53.6% |
| 他人に評価をしてもらうことで得るものがあった | 26<br>92.9%  | 20<br>71.4% |

評価シートを学生にフィードバックした後に今回の取り組みについて14項目のアンケートを実施した。「次も作品発表会に参加したいですか」の質問に対して「はい」と回答した積極群と「いいえ」「どちらでもない」と回答した消極群の2群に分けた。それぞれの「はい」の回答数と比率を比較したのが、表1 作品発表会の振り返りに示す。消極群と積極群とでやや差があった項目の一覧である。特に「自分の作品を他人に評価してもらって良かった」の差が大きかった。

表2 関連の高い質問

| 質問項目1                 | 質問項目2                  | 相関係数 | 有意確率  |
|-----------------------|------------------------|------|-------|
| 参考になる作品があった           | 他人を評価することで得るものがあった     | 0.63 | 0.024 |
| 自分の作品を他人に評価してもらって良かった | 他人に評価をってもらうことで得るものがあった | 0.62 | 0.044 |

相関関係があり危険率5%以上の項目を表2 関連の高い質問に示す。「参考になる作品があった」と「他人を評価することで得るものがあった」、「自分の作品を他人に評価してもらって良かった」と「他人に評価をってもらうことで得るものがあった」の組み合わせで相関が確認できた。

これまで自由テーマで工作するだけではなく、作品発表会と相互評価まで行う実践は教育効果があることを示唆した。

# 日本の活断層直上の高速道路分布の調査

中村 洋介<sup>†</sup> 曾根 新<sup>†</sup>

<sup>†</sup>福島大学 人間発達文化学類文化探究専攻

キーワード：活断層，高速道路，地震断層，活断層データベース

## 1 はじめに

現在、日本の道路の総延長は120万kmを超えており、人々の移動や貨物輸送などに使用され日々私たちの生活を支えている。国土交通省の調査によると、平成27年の貨物車の輸送量の23%は都道府県間、52%は地域ブロック間の輸送である。そのため都道府県間やブロック間をつなぎ、短時間で離れた地域に向かうことが出来る高速道路は日本の貨物輸送の骨幹を支えており、高速道路を用いた積載量の大きな貨物車による輸送が日本国民の生活を支えるエッセンシャルワークとして、物流ネットワークの基盤になっている。

しかしながら、日本には多くの活断層が存在しており、2016年の熊本地震(M7.3)のような内陸で活断層型の地震が発生した場合、高速道路では地震の揺れによる被害だけではなく地表地震断層がずれることによって断層付近の地面が変位するといった被害が発生し、日本の物流ネットワーク、ひいては私たちの生活に大きな影響が出ると考えられる。そのため、内陸で活断層型の地震が発生した際に大きな被害が出ると考えられる高速道路と活断層が交差、または接近する箇所を調査し、分布の特徴や特に危険な箇所を調査した。今後の日本の地震災害対策に資するデータの提供を目指し、本研究を実施した。

## 2 高速道路について

本研究では研究対象の高速道路は国土交通省によりナンバリングがされた道路とする。高速道路ナンバリングは高速道路に路線番号を付することにより、訪日外国人をはじめ、すべての利用者によりわかりやすい道案内の実現を目指すために導入さ

れた。対象路線は高規格幹線道路網（「高速自動車国道」および「一般国道自動車専用道路」）と、この道路網を補完して地域の高速道路ネットワークを形成する路線、高規格幹線道路網から主要な空港・港湾、観光地へのアクセスとなる高速道路ネットワークを形成する路線であり、首都高速道路や阪神高速道路など独自にナンバリングが行われていた都市高速道路は除外した。

## 3 研究方法

本研究では以下5つの調査を行った。なお、調査にあたり(独)産業技術総合研究所発行の「活断層データベース」、GoogleEarth、政府地震調査研究推進本部のホームページ、高速道路資料室というウェブサイトを利用した。

① 活断層より100m以内に位置する高速道路をリストアップ

② ①で確認した箇所全ての周辺の土地利用(森、田畑、市街地、森および田畑、森および市街地、田畑および市街地)の調査

③ ①で確認した箇所のうち特筆すべき道路の特徴(橋、トンネル、IC・JCT)がある場合の調査

④ ①で確認した箇所全ての活断層の地震の長期的な発生予測についてランク付け(30年以内の地震発生率が3%以内の場合S、0.1~3%の場合A、0.1%未満の場合Z、過去の地震発生に関するデータが少なく地震発生率が不明の場合X)

⑤ ①で確認した箇所全ての高速道路の令和2年12月31日時点の開通後の経過年数

## 4 調査結果

表1 調査1から調査3の結果

| 都道府県 | 森   | 田畑 | 市街地 | 森/田畑 | 森/市街地 | 田畑/市街地 | 合計   | 橋   | トンネル | IC・JCT |
|------|-----|----|-----|------|-------|--------|------|-----|------|--------|
| 北海道  | 20  | 10 | 6   | 0    | 0     | 0      | 36   | 1   | 2    | 2      |
| 青森   | 2   | 7  | 0   | 0    | 0     | 2      | 11   |     |      | 1      |
| 岩手   | 5   | 2  | 0   | 0    | 0     | 0      | 7    | 1   |      | 1      |
| 宮城   | 5   | 6  | 5   | 0    | 0     | 0      | 16   | 3   |      | 3      |
| 秋田   | 10  | 7  | 0   | 0    | 0     | 0      | 17   |     |      | 1      |
| 山形   | 2   | 1  | 1   | 1    | 0     | 3      | 8    |     | 1    | 1      |
| 福島   | 5   | 2  | 0   | 1    | 2     | 0      | 10   |     | 2    | 2      |
| 茨城   | 2   | 0  | 0   | 0    | 0     | 0      | 2    |     |      |        |
| 栃木   | 0   | 0  | 0   | 0    | 0     | 0      | 0    |     |      |        |
| 群馬   | 0   | 1  | 0   | 0    | 0     | 1      | 2    | 1   |      | 1      |
| 埼玉   | 0   | 0  | 0   | 0    | 0     | 2      | 2    | 1   |      | 1      |
| 千葉   | 0   | 0  | 0   | 0    | 0     | 0      | 0    |     |      |        |
| 東京   | 0   | 0  | 2   | 0    | 0     | 0      | 2    | 1   |      |        |
| 神奈川  | 4   | 1  | 7   | 0    | 0     | 1      | 13   | 7   | 1    | 3      |
| 新潟   | 5   | 2  | 4   | 0    | 2     | 2      | 15   | 1   | 2    | 2      |
| 富山   | 1   | 2  | 0   | 0    | 1     | 5      | 9    | 1   | 1    | 2      |
| 石川   | 0   | 1  | 0   | 0    | 1     | 1      | 3    | 1   |      |        |
| 福井   | 9   | 7  | 1   | 4    | 1     | 2      | 24   | 7   | 3    | 7      |
| 山梨   | 1   | 0  | 1   | 1    | 0     | 1      | 4    | 1   |      | 1      |
| 長野   | 12  | 7  | 7   | 10   | 4     | 4      | 44   | 5   | 7    | 3      |
| 岐阜   | 19  | 3  | 2   | 1    | 2     | 2      | 29   | 6   | 7    | 5      |
| 静岡   | 6   | 1  | 5   | 1    | 4     | 0      | 17   | 7   | 7    | 3      |
| 愛知   | 2   | 1  | 5   | 0    | 1     | 2      | 11   | 4   | 1    | 6      |
| 三重   | 9   | 9  | 3   | 2    | 0     | 2      | 25   | 7   | 2    | 9      |
| 滋賀   | 10  | 1  | 2   | 1    | 2     | 0      | 16   | 2   | 1    | 1      |
| 京都   | 12  | 3  | 13  | 0    | 0     | 0      | 28   | 15  | 7    | 7      |
| 大阪   | 7   | 1  | 11  | 0    | 0     | 1      | 20   | 10  | 2    | 2      |
| 兵庫   | 11  | 7  | 6   | 3    | 3     | 0      | 31※  | 5   | 3    | 8      |
| 奈良   | 4   | 4  | 1   | 2    | 1     | 2      | 14   | 3   |      | 3      |
| 和歌山  | 1   | 1  | 0   | 2    | 1     | 1      | 6    | 1   |      | 2      |
| 鳥取   | 0   | 0  | 0   | 0    | 0     | 0      | 0    |     |      |        |
| 島根   | 0   | 0  | 0   | 0    | 0     | 0      | 0    |     |      |        |
| 岡山   | 0   | 0  | 0   | 1    | 0     | 0      | 1    | 1   | 1    |        |
| 広島   | 8   | 0  | 2   | 0    | 0     | 0      | 10   | 2   | 5    | 1      |
| 山口   | 10  | 0  | 0   | 1    | 1     | 0      | 12   | 1   | 3    | 3      |
| 徳島   | 13  | 2  | 1   | 7    | 1     | 1      | 25   | 9   |      | 4      |
| 香川   | 0   | 0  | 0   | 0    | 0     | 0      | 0    |     |      |        |
| 愛媛   | 10  | 2  | 1   | 4    | 2     | 0      | 19   | 10  | 4    | 5      |
| 高知   | 0   | 0  | 0   | 0    | 0     | 0      | 0    |     |      |        |
| 福岡   | 4   | 0  | 2   | 0    | 2     | 1      | 9    | 4   | 1    | 2      |
| 佐賀   | 1   | 2  | 0   | 0    | 0     | 0      | 3    | 1   |      | 1      |
| 長崎   | 0   | 1  | 1   | 0    | 0     | 0      | 2    |     |      | 1      |
| 熊本   | 5   | 3  | 0   | 1    | 3     | 1      | 13   | 5   | 2    | 4      |
| 大分   | 22  | 0  | 1   | 1    | 0     | 0      | 24   | 9   | 3    | 4      |
| 宮崎   | 2   | 2  | 0   | 1    | 0     | 0      | 5    |     |      | 2      |
| 鹿児島  | 0   | 0  | 0   | 0    | 0     | 1      | 1    | 1   |      | 1      |
| 沖縄   | 1   | 0  | 0   | 1    | 0     | 0      | 2    | 1   |      |        |
| 合計   | 240 | 99 | 90  | 46   | 34    | 38     | 548※ | 135 | 68   | 105    |

表2 調査4の結果

ただし「活断層データベース」において発見された活断層の一部には主要活断層として地震本部で評価がされていないものもあった。その場合の評価は『-』としている。

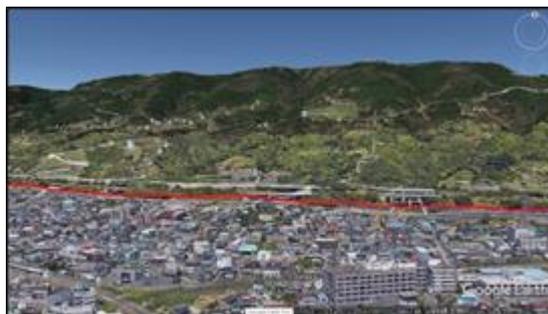
| Sランク | Aランク | Zランク | Xランク | -  | 合計  |
|------|------|------|------|----|-----|
| 70   | 159  | 184  | 58   | 77 | 548 |

表3 調査4の結果

| 0-10年                             | 11-20年 | 21-30年 | 31-40年 | 41-50年 | 51年- | 合計  |
|-----------------------------------|--------|--------|--------|--------|------|-----|
| 68                                | 83     | 120    | 110    | 110    | 57   | 548 |
| 活断層評価がSまたはAランクの箇所みの場合(S-70 A-159) |        |        |        |        |      | 合計  |
| 32                                | 36     | 52     | 44     | 34     | 31   | 229 |

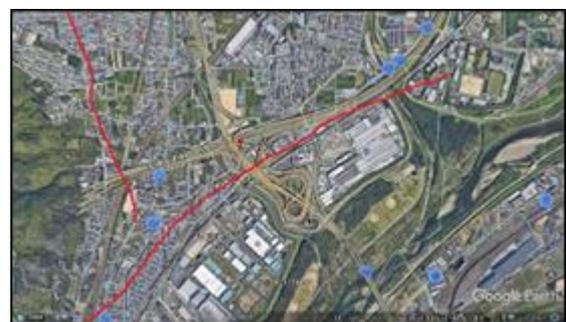
また、調査を進める中で活断層と高速道路の近接の仕方にパターンを確認し、中でも地震が発生したときに被害が大きくなると考えられるものが2パターン確認できた。

○パターン1 活断層と高速道路が3 km以上並行している箇所(計22か所)



(神奈川県 平山-松田北断層帯と東名高速道路)

○パターン2 JCTと活断層が交差する箇所(計17か所)



(京都府 有馬-高槻断層帯、京都西山断層帯と大山崎JCT)

## 5 考察

調査1より、活断層から100m以内に位置する高速道路が計548か所確認できた。地域ブロックごとに分布を比較すると、関西地方に比較的該当箇所が多かった。関西地方は調査2より該当箇所の周辺の土地利用が市街地であることが多く、建物や高架橋の倒壊の可能性があるため相対的に危険度が高い状態にあると言える。また、調査4より地震調査研究推進本部による長期評価がSランクやAランクといった発生確率が比較的高い活断層が関西地方には多く、高速道路に対する耐震工事や発災時の対応をあらかじめ決定しておく必要があると考える。

調査3の結果、橋やトンネル、ICやJCTといった特徴を1つでも持つ調査箇所は268か所あり、調査1で確認できた箇所の約半数近くであった。これらの構造物は高速道路に多用されるものであるが、地震による被害を受けた場合は復旧に時間を要するものであり、迂回のために一般道への交通集中などの影響も考えられる。

国土交通省によると今回の調査対象の高速道路の多くが該当する高速自動車国道の供用延長は一定して伸びているため、調査5の結果から近年は高速道路と活断層が交差しないよう意識して新線の建設がされていると考えられる。しかし調査4の結果SもしくはAランクに分類された箇所のみで集計すると年代ごとの大きな変化は見られなくなり、活断層の長期評価は考慮せずに路線を引いている可能性がある。

活断層と高速道路の近接の仕方にパターンのうち、活断層と高速道路が3km以上並行している箇所は22か所あり、滋賀県や兵庫県、愛媛県では複数箇所確認できた。中には活断層の断層運動によってできた急崖である断層崖や撓曲で生じた斜面である撓曲崖の崖上や崖下にぴったりと張り付くように路線が引かれている場合もあり、実際に地震が発生した際に並行している区間全域で変位が起き、長い区間で壊滅的な被害を受ける可能性や土砂災害の可能性もある。

JCTと交わる箇所は17か所あった。JCTでは立体的に道路が交差しており複雑な構造をしているため高架橋が倒壊した場合は下を走る道路にも被

害を拡大し復旧には時間には相当の期間を要する。また、複数の路線が交わっているためJCTが通行止めになると様々な方面への行き来が制限されることになり、地震が起きた地域以外への物流などの面で被害が出るのが推測できる。

## 6 まとめ

本研究では活断層直上の高速道路について該当箇所、周辺の土地利用、該当箇所の特徴、活断層の長期評価、高速道路の開通からの経過年数の5つの調査を行った。また、調査を行う中で活断層と高速道路の危険な近接パターンを2つ確認できた。このパターンに当てはまる中でも長期評価が高いうえ巨大な断層崖と接する箇所や大規模かつ大都市に近いJCTなど早急に対応が必要な箇所もあった。

高速道路にも大きな被害を及ぼした兵庫県南部地震(M7.3)では高速道路の不通による迂回路の渋滞や通行規制箇所の把握の難航などが発生した。本研究のように危険な箇所をあらかじめ把握し代替路の設定や耐震工事の徹底など平時の備えをすることが何よりも防災・減災に繋がると考えられる。

### [参考文献]

- ・国土交通省 道に関する各種データ集 道路 ([https://www.mlit.go.jp/road/soudan/soudan\\_10b\\_01.html](https://www.mlit.go.jp/road/soudan/soudan_10b_01.html))
- ・国土交通省 我が国の物流を取り巻く現状 (<https://www.mlit.go.jp/common/001263649.pdf>)
- ・産業技術総合研究所 活断層データベース 起震断層・活動セグメント検索 (<https://gbank.gsj.jp/activefault/search>)
- ・政府 地震調査研究推進本部 (<http://www.jishin.go.jp/>)
- ・内閣府防災情報のページ 阪神・淡路大震災教訓情報資料集【01】道路交通 ([http://www.bousai.go.jp/kyoiku/kyokun/hanshin\\_awaji/data/detail/1-6-1.html](http://www.bousai.go.jp/kyoiku/kyokun/hanshin_awaji/data/detail/1-6-1.html))

# 『三才図会』の雙陸盤面図について

木子香<sup>†</sup> 張 超然<sup>†</sup> 徐 恒遠<sup>†</sup> 高見友幸<sup>†</sup>  
<sup>†</sup>大阪電気通信大学総合情報学部

キーワード：三才図会 雙陸 盤双六

## 1 はじめに

中国の古代盤上遊戯である雙陸（ジャンル・盤双六）は歴史が古く、シルクロードを往来したソグド人の商人が西方から伝えたとされている[1]. 雙陸は三世紀に中国に伝来してから十九世紀清の時代に滅びるまで、千年以上に渡り遊ばれ続けてきた[2]. 中国には雙陸に関する古代書籍の文献が多く存在するものの、雙陸遊戯法について書かれているものは殆どない. しかし、1151年、中国南宋の知識人洪遵(1120~1174)によって雙陸専門書『譜雙』（ふそう）が世に出された. 『譜雙』は図版が豊富であり、雙陸初期配置や遊戯法も図と文章で説明されており、当時の雙陸を知る上で大変貴重な資料である. 近年、十七世紀中国明の時代に編纂された『三才図会』の中から9枚の雙陸盤面図が掲載されていることがわかった. これまで『譜雙』以外に雙陸に関する文献には図版が殆どない中、この9枚の盤面図は雙陸遊戯法の変遷の研究において重要な手がかりとなるものである. 本稿では、この『三才図会』に掲載されている盤面図を紹介しながら当時の遊戯法について分析していく.

## 2 雙陸盤面および名称について

中国の雙陸は北雙陸と南雙陸に分れており、遊戯盤盤面のデザインや名称も異なり、同じ事柄に対して、複数の言い方が見られる[3]. 北雙陸盤面にはマス線が描かれておらず、代わりに「梁標」という丸い印で表している. 南雙陸盤面にはマスが線で描かれており、盤の中心にある「六頭屋」マスには、対角線で引かれている. また、南北盤面の各名称も異なるが、どちらも盤の中心から左右に六マスになっており、全部で24マスの盤になっている（図1, 2）. これは日本盤双六の盤面マ

ス数と同様である（図3）. 雙陸の各種名称について日本盤双六名称と対照し、以下の表にまとめる（表1）.

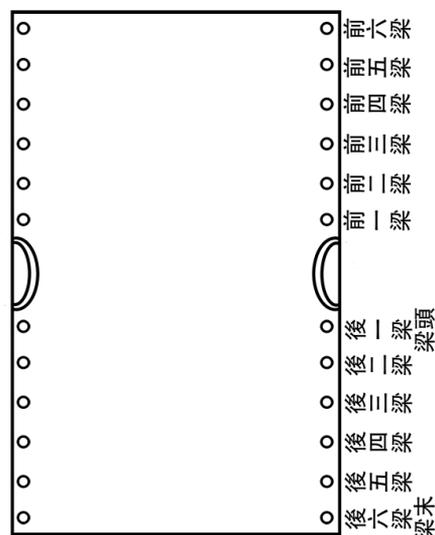


図1 北雙陸盤面

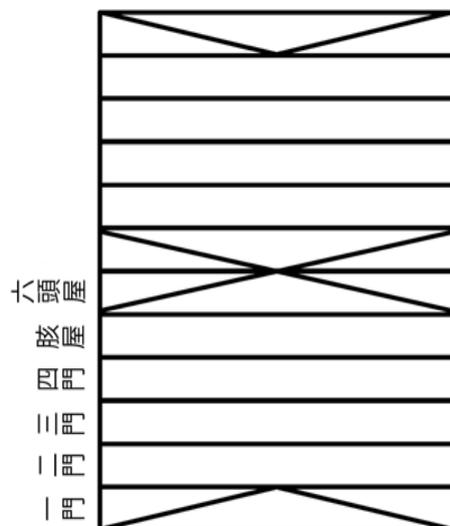


図2 南雙陸盤面

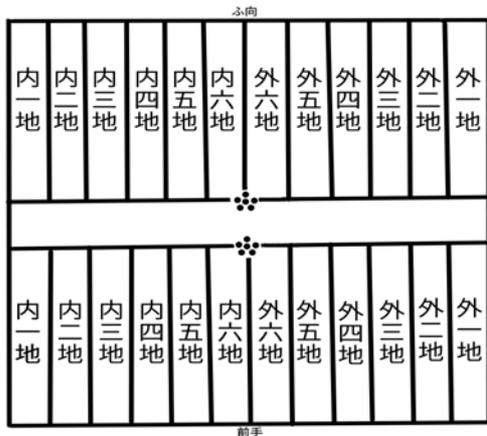


図3 日本盤双六盤面

表1 日中盤双六名称対照

| 日本                         | 中国                          |
|----------------------------|-----------------------------|
| 石 (駒)                      | 馬                           |
| 石飾り<br>(初期配置)              | 布局<br>(初期配置)                |
| 地 (マス)                     | 梁 (北雙陸)<br>門 (南雙陸)          |
| 外地 (外一地～六地)                | 前一～六梁 (北雙陸)<br>外一～六門 (南雙陸)  |
| 内地 (内一地～六地)                | 後一～六梁 (北雙陸)<br>内一～六門 (南雙陸)  |
| 自陣<br>(内一地～六地)             | 宮<br>北雙陸の後一～六梁<br>南雙陸の内一～六門 |
| 賽・賽子・骰子<br>(サイコロ)          | 骰子・彩<br>(サイコロ)              |
| 一荷<br>(一つのマスに自分の駒を二つおいた場合) | 両子相比 (北雙陸)<br>做屋 (南雙陸)      |

また、骰子の出目の呼称について、日中共にゾロ目の場合は数字の前に「重」を加える。例えば、1のゾロ目は「重一」、2のゾロ目は「重二」のように呼ぶ。4のゾロ目は「重紅」と呼ぶが、中国の骰子の4の目は赤く染められているからである。

### 3 『三才図会』の盤面

『三才図会』は、中国明の時代に王圻（おうき、1530-1615）とその子王思義（おうしぎ）により編纂された明時代の経典的類書である。全106巻からなり、「天文」「地理」「人物」など14部で構成されている。『三才図会』は、膨大な内容と多様な版画挿絵があり、中国古代版木の傑作とも言われている[4]。この『三才図会・人事十卷』には雙陸の項があり、9枚の盤面図が描かれている。図の上や図の中に、盤面図を説明する文章や歌もある。先述の『譜雙』の中にも各種初期配置が図で描かれているが、『三才図会』の盤面図には、『譜雙』と異なる部分が見られる。

『三才図会』盤面の「梁標」は「門」の上下に5個描かれ、全部で20マスになっている。このような20マスの盤面図は、現在日中文献の中に類をみない。これに関して次の節で説明する。

#### 3.1 雙陸盤式

図の上部に書かれた文の大意：骰子の出目で先手を決め、一定のルールに従いながら、プレイする(図4)。

図4の盤面の上に「北為内家」「南為外局」と書かれており、これは、北エリアを「内家」と称し、南エリアを「外局」と称するという意味である。盤面の中心部に三日月のような形をしている「門」があり、「門」の上と下に「梁標」が印されている。盤の中央に「中河」の文字が書かれている。更に、盤面には「逢門不作家」「此是外華」の記述がある。

「此是外華」というのは、南エリアは「外華」ともいい、「逢門不作家」は門の場所には馬を置かないと理解するのが正しいであろう。馬は白黒各15個配置され、『譜雙』の常局格制や日本盤双六の「本双六」の初期配置と同じ並べ方である。この「雙陸盤式図」は『三才図会』雙陸の項に描かれた最初の一枚目である。上記の解析により、この図は雙陸の初期配置を表していることが分かる。盤面の

上には南北の方向が示されている。読者から見た盤は、上は北、下は南となっている。盤面図をよく見ていくと、門の上下は「梁標」が5個ずつしかない。この盤面図は前後五梁である。



図4 雙陸盤式

### 3.2 外華平満局

図の上部に書かれた文の大意：前四梁外華平満局、後の梁を「内榮梁」といい、前梁を外華および常作という。骰子の出目が重四の場合は馬を四梁に置く。所謂外華なり（図5）。

この文から分析していく。「門」を中心として南の外華にある梁は前梁、北の内家にある梁は後梁または内榮梁と称するということである。つまり、南を前、北を後ろとする見方である。図5を見ていくと、「外華平満局」は「雙陸盤式」の馬の並び方と違い、前一梁にある五つの馬の二つは前四梁に移動したことがわかる。全体の馬数を見ると、黒は15個あるが、白馬は13個しかない。通常の15個から計算すると馬は二つ足りない。また、前四梁にある黒馬と向かい合わせの場所に、白馬は1個しか描かれていないが、文章から推測すると、ここはもう一つの白馬があるべきである。また、前五梁にある白馬は一つ足りないが、これも上記

と同様である。この盤面図は前後五梁である。



図5 外華平満局

### 3.3 三梁外華定局

図の上部に書かれた文の大意：前三梁の外華で配置。骰子の出目が重三の場合は、馬を外華三梁に配置する（図6）。

図6の盤面には「三のゾロ目の場合は、一梁にある五つの馬の中から前三梁に配置する」と書かれている。

盤面から、前一梁の白黒馬を各二つずつ前三梁の場所に配置したことが分かる。また、図で描かれた馬の数は、白馬が16個、黒馬が14個である。後一梁に白黒の馬が混在している。「雙陸盤式」から考えると二つとも黒馬であるのが正しいと考える。この盤面図は前後五梁である。

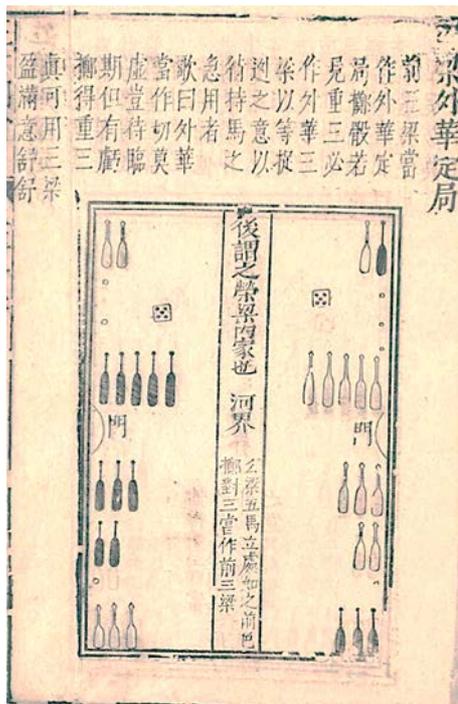


図6 三梁外華局

### 3.4 二梁外華定局

図の上部に書かれた文の大意：前二梁は外華定局にする。骰子の出目が重二の場合は，外華に配置する（図7）。

図7は外華前一梁にある五つの馬のうち二つを前二梁に移動したものである。図6と同様に後一梁に白黒の馬が混在しているが、「雙陸盤式」図から考えると，二つとも黒馬であろう。また，この図では後梁が六つあるようにも見えるが，これも印刷上の間違いだと考える。

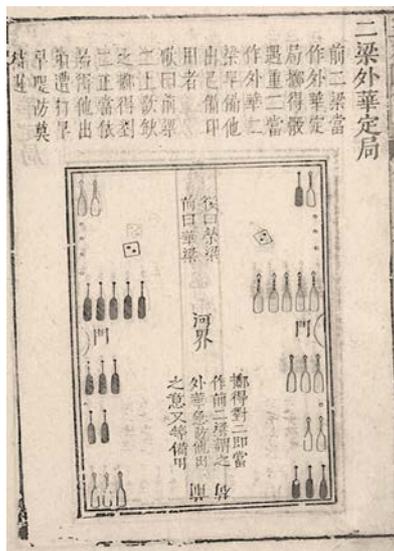


図7 二梁外華定局

### 3.5 蔵馬定局

図の上部に書かれた文の大意は：「把門」（バモン）できてない状況の下では，骰子の出目が五の時，前一梁にある五つの馬の場所に馬を一つ隠し，前一梁に馬を六つ置く。所謂蔵馬定局（図8）。

『譜雙』によると，北方雙陸は前一梁を「門」という，馬二つが門にたどり着くと「把門」という[5]。また，『譜雙』の中に記載されている北雙陸盤の「前一梁」は三日月の形をしている「門」と隣接している梁になる（図1）。この蔵馬定局は，「雙陸盤式」後一梁にある白黒各二つの馬を図8のように，前一梁に移動した。ここは馬が「門」を超えて進むことになっている。「雙陸盤式」に書かれている「逢門不作家」，つまり門の場所には馬を置かないという意味から，「門」を一つの梁だと考える。そうすると，五のゾロ目を一個の馬に対して使い，一個の馬が10梁を進めたこととなり，ちょうど図8の配置になる。前一梁には馬が六つ並んでいる状態になる。この盤面図は前後五梁である。



図8 蔵馬定局

### 3.6 他骸（ハイ）定局

図の上部に書かれた文の大意は：骰子の出目が重

紅(四のゾロ目)の時、後一梁の馬を相手の胷の場所に置く(図9)。

『譜雙』によると北方雙陸は後一梁を胷と称する[5]。ここでの他胷は相手の胷を意味する。図9に描かれたように、後一梁にある白黒馬各二つは後四梁に移動したことが分かる。また、白黒馬の配置は「雙陸盤式」の白黒馬の配置とちょうど逆になっている。この盤面図は前後五梁である。

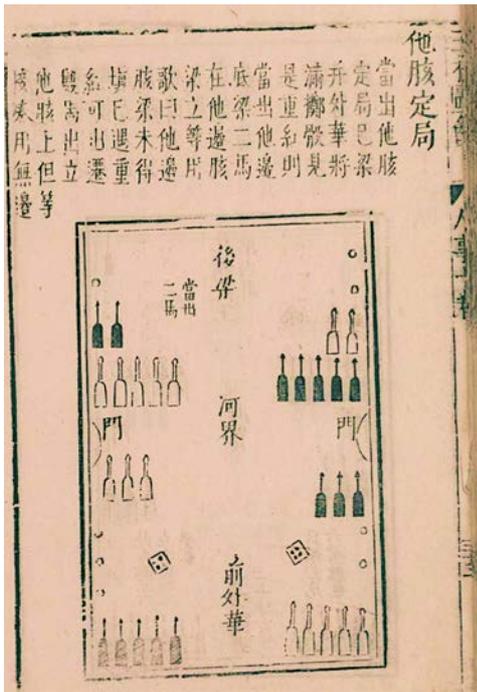


図9 他胷定局

### 3.7 四梁定局

図の上部に書かれた文の大意：相手方の四梁で馬を配置する。骰子の出目が重三の時に、二つの馬を後四梁に置き待機する(図10)。

図10 盤面には「重三は一梁の二つの馬を出すべき」「重三は四梁に置くべき」と書かれている。

「雙陸盤式」の初期配置では、後一梁に白黒馬を各々二つずつ置くが、重三の出目がでた時に、それらが後四梁に移動したことが分かる。また、お互いに相手の梁に置くことから、「他胷定局」とよく似ている。図10は図9と比較すると、図10の「梁標」が図9より一つ少ない以外は、ほぼ同じである。この盤面図は前後五梁である。

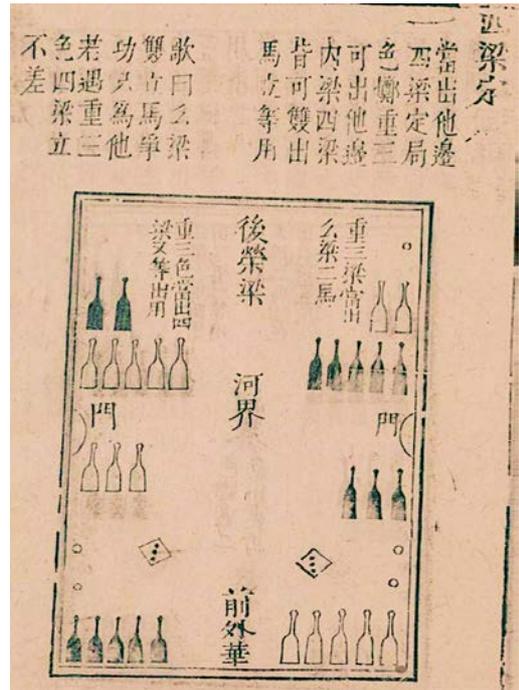


図10 四梁定局

### 3.8 三梁定局

図の上部に書かれた文の大意：骰子の出目が重二の時に、相手の場所に馬を置く。馬二つを三梁に置く(図11)。

図11の盤面上には「么梁両里随色出」「么梁両馬随色出二梁等用」また盤面下部には「擲色重二急出裏馬」の説明が書かれている。

「么梁両里随色出」の「里」を「馬」に置き換えてみると意味が通る。「么梁両馬随色出二梁等用」の文は「二梁」と書いているが、図の上部にある説明文や盤面上に書かれている文の意味から、「二梁」を「三梁」にすると文全体の意味と合う。よって文の大意は「一梁にある二つの馬を骰子の出目に従い三梁に移動する」である。盤面下部に書かれた文の意味は「骰子の出目が重二の時、馬二つを出す」である。

盤面図を見ると、元々内一梁にある二つの馬は、白黒二つ共に三梁に置かれている。文中に「相手の場所に置く」の文言が書かれていたが、「他胷定局」や「四梁定局」のように白黒馬の位置が逆転していない。また、文の中に二のゾロ目と書かれているが、盤面に描かれた骰子の出目は「2, 1」になっている、骰子目の書き方は1の出目の「〇」

は骰子の一面の中央に描くのに対して、この図の「○」がずれている。盤面の骰子の出目は二つとも「2」であろう。この盤面図は前後五梁である。

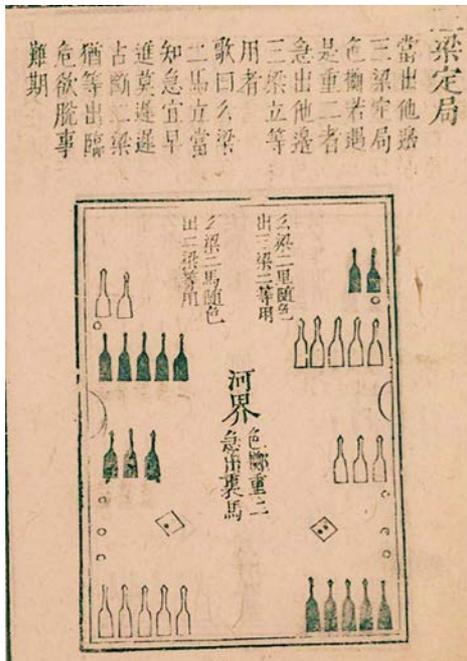


図 11 三梁定局

### 3.9 二梁定局

図の上部に書かれた文の大意：骰子の出目が么（一のゾロ目）の場合は、相手の二梁に馬をおき、定局とする。盤面上に「色擲雙么出在二梁」「色出争二梁」と書かれている。いずれも、一のゾロ目の場合は馬を二梁に置く(図 12)。

盤面図から、「雙陸盤式」で後一梁にある白黒馬各々二つが後二梁に移動したことが分かる。この盤面図は前後五梁である。



図 12 二梁定局

図 4～図 12 は、国立国会図書館デジタルコレクションより

### 4 「五梁」と「初期配置」の問題

まず、「五梁」の問題を見ていく。『三才図会』雙陸の項に書かれた 9 枚の盤面図のうち「二梁外華定局」を除いて、「門」を中心に前後に五梁ずつ描かれている。これは今までに分かっている雙陸文献に類のない盤面図とも言える。盤面のデザインは『譜雙』に記載されている北雙陸の盤面デザインと同様であるが、「梁」の名称は異なる。『三才図会』の盤面の前梁後梁は、『譜雙』の北雙陸盤面の前梁後梁と逆になっている(図 1 を参照)。

次は、初期配置を見ていく。『三才図会』のこの 9 枚の盤面図は、1 枚目の「雙陸盤式」が基本的な初期配置だとすると、その他の 8 枚は骰子を一回振り、その出目で馬の配置を決めている。このような出目によって馬の配置を決めるのは、今までに類がなく、新しい。一方、「蔵馬定局」の初期配置は、前一梁に馬を六つ配置している。一般的な初期配置は一つの梁に置ける馬が最多 5 個までに対して、「蔵馬定局」はこの決まりを破っている。

## 5 おわりに

『三才図会』は、今までにない情報を提供してくれたと同時に図版と文章において不明な箇所も存在している。今後は詳細な解析や他の文献との比較により、中国雙陸遊戯法、日中盤双六の関連性について検証していく必要がある。

### 参考文献

- [1] 増川宏一, 『遊戯』, 法政大学出版局 (1978), p44
- [2] 木子香, 古代盤上遊戯盤双六の復刻, 大阪電気通信大学人間科学研究第 20 号 (2019), pp1-12
- [3] 木子香, 『譜雙』の日本語訳及び盤双六史に関する考察, 大阪電気通信大学人間科学研究第 19 号 (2017), pp79-94
- [4] 何立民, 「王圻父子『三才図絵』的特点与価値」, 『史林』3 期 (2014), pp54-59
- [5] 洪遵, 『譜雙』, 早稲田大学図書館所蔵 (1846)

付録：各盤面図の上にある文章の意味

### 雙陸盤式

おおよそ馬の動かし方の順番は骰子で決める。即ち馬の動かし方は一定のルールがあり、勝手に動かしてはならない。

営するのに一定の道理があり、勝手に動かしてはならない。

歌曰く：馬を動かす時は恐れが要らず、時に骰子の出方で勝負が決まり、左南右西は順序に分かれ、上下はあなたの好きなようにすればよい。

### 外華平満局

前四梁は外華平満局、後の梁は榮梁という。前梁は外華および常作という。骰子の出目が四のゾロ目の時に、四梁にするのがよいだろう。これを外華という。

歌曰く：外華四梁が欠けた時に、ゾロ目の四を得たら、正にそこに配置せよ、他の出方を防ぐため、早目にそう配置した方がよい。

### 三梁外華定局

前三梁は外華定局とする。三のゾロ目が出た時に、必ず外華三梁に配置し、捕まえるか逃げ出すかを待ち、急に馬が必要な時に備える。

歌曰く：外華を空けずに、馬を置くべき、この時が来るのが待ち遠しい。三のゾロ目は本当に役に立つ、三梁が満ちて気持ちが良い。

### 二梁外華定局

前二梁は外華定局とする。二のゾロ目が出た時に、必ず外華二梁に配置する。相手の手を防ぎ、己が馬を使う時に備える。

歌曰く：前二梁を補填したい時に、二のゾロ目を得ればよい。相手の出番なら切られるかもしれない、防備するには早めの方がよい。

### 蔵馬定局

まだ「把門」していないうちに、五の出目が出た時は、馬一つを隠すべし。五つの馬がある前に更に馬を一つ置く、いわゆる蔵馬局。

歌曰く：五六欠けるとき出すべし、遅れないよう

に早々に隠す。

### 他胷定局

他胷に置くべき。已梁外華が満になりそうな時に、四のゾロ目が出たら、相手の梁に置き、出番を待つ。

歌曰く：相手の胷梁がまだ埋まっていない時、四のゾロ目が出たら、そこに移動することができ、両馬は初めて相手の胷に立ち、その使い道は計り知れない。

### 四梁定局

相手の四梁に置き定局する。三のゾロ目が出たら、相手の内梁四梁に両馬を移動し、待機する。

歌曰く：一梁に両馬が立ち、それらは勝利に貢献し、三のゾロ目が出たら、四梁に立つのは間違いない。

### 三梁定局

相手の三梁に置き定局する。二のゾロ目が出たら、相手の三梁に移動し、待機する。

歌曰く：一梁に馬が立っているのは知るべし、早めに進め、遅れることのないように。二梁を占領し出るのを待機、危険から逃げ出そうとしても物事は予測できず。

### 二梁定局

相手の二梁に置き定局する。已梁曰く満になり、一のゾロ目が出たら、相手の二梁に移動しそこで待機する。

歌曰く：一のゾロ目が出たらすぐに使うべし、二梁に配置するのは知っておこう。使うかどうかは分からずとも、早々に備え、遅れぬよう心掛けるべし。

# 日本の古代都城における設計数値の継承（序報）

高見 友幸

大阪電気通信大学 総合情報学部

キーワード 初期平安宮, 藤原宮, 周礼「考工記」, 数値の継承, 都城の設計

## 1 はじめに

周礼「考工記」匠人営国の条に記載された都城モデル（以下、周礼の都城モデル）は、日本の古代都城の研究において強く意識されるべきである。本論文の結論はこの一言に尽きる。周礼の都城モデルは

匠人営國。方九里，旁三門。國中九經九緯，經涂九軌。左祖右社，面朝後市。市朝一夫。

という記述に基づいたモデルである。建てるべき都城の規範として、方九里以下7項目が挙げられている。本論文では、このうち、従来の研究では最も誤解の大きい「方九里」について議論する。「方」の意味については、多くの研究者が、都城の一辺の長さとして解釈している。その長さがほぼ9里に近い長さとなるからである。しかし、第2節と第3節で示されるとおり、「方九里」の9里は正確に9里の長さなのである。しかも、9里に対応する長さは京城の一辺ではなく、宮殿の周囲の長さである。実際、平城宮の周囲長は正確に9里（高麗尺で1350丈）であるにもかかわらず、藤原宮や平安宮の周囲長が9里ではないため、この事実が注目されることはなかった。

方九里の考察から、新しい尺度の存在を導くことができる。藤原宮ではF尺、平安宮ではX尺が、その新しい尺度である。これらの尺度の存在を検証するひとつの手段として、文献[1][2]で提唱された「数値の継承」という考え方がある。第4節では、数値の継承の事例をいくつか紹介し、方九里の成立が確からしいことを示す。

図1に、藤原京、初期平城京、初期平安京の寸法を示した。図1では、藤原京だけが定説に沿うものであり、他の2都城はなお仮説の段階にあり、今後の検証を待たねばならない。さて、宮の周囲長は、藤原宮、平城宮、平安宮で、それぞれ、1200丈（高麗尺）、1350丈（高麗尺）、1526丈（天平尺）である。藤原宮と平安宮の周囲長は9里（高麗尺で1350丈、天平尺では1620丈）ではない。しかし、この設計を方九里に基づく設計であると見るのが本論文の考え方である。

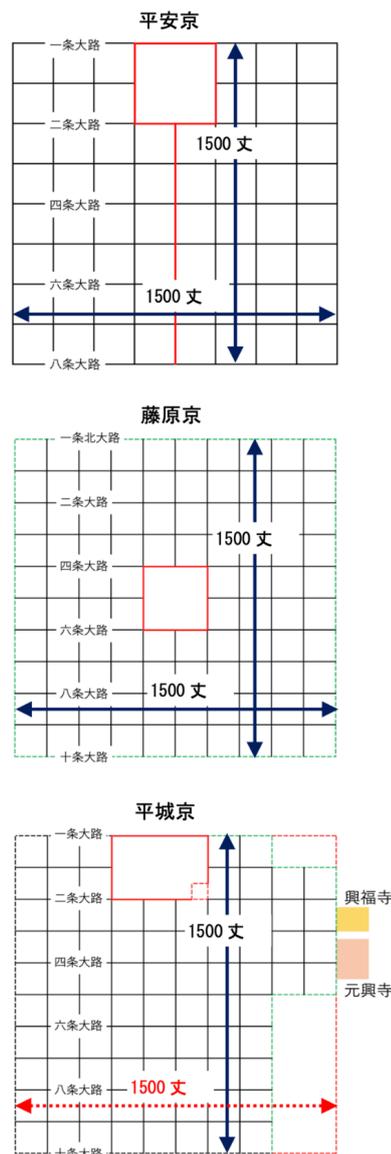


図1. 上) 初期平安京（仮説）、中) 藤原京、下) 初期平城京（仮説）.



平安宮の大極殿院の東西幅は424尺、朝堂院の東西幅は640尺と復原されている。これは天平尺（1尺=29.8cm）による復原であるが、「方九里」を満足させるX尺が使われている可能性の検討が必要となろう。文献[1]では、陽明文庫本「宮城図」をもとに検討がなされており、X尺を用いて、大極殿院および朝堂院の東西幅を400尺、600尺と帰結している。

## 5 おわりに

周礼の都城モデルにおける「方九里」の規範を、藤原宮、平安宮の設計者はかくも厳密にとらえたのである。大雑把に9里なのではなく、正確に9里として「作られた」のである。それは、尺度を都合よく変えて実現しただけの単なる数字合わせと見るべきではない。むしろ数字を合わせるこそが重要と見るべきである。

「方九里」の他に注目すべき都城モデルの規範として、「左祖右社」がある。宮殿の左に、祖先を祀り、宮殿の右には、土地の神様を祀るとというのがその解釈である。本論文では、話題としなかったが、「方九里」がそうであったように、「左祖右社」も非常に厳密に守られていることは注目すべきである[4]。つまり、藤原宮の場合、天子南面して、左には伊勢神宮が、右には淡路島に伊弉諾神宮が建てられているのである。しかも、伊勢神宮、藤原宮、伊弉諾神宮は、藤原宮を中点として、ほぼ同じ緯度にある。周礼の都城モデルをこれほどに厳密にとらえる古代人の信仰心は、古代史を考える上で強く意識すべきであろう。

## 注記

1) 宮殿の周囲に河川を想定することはさほど不自然ではない。実際、平安宮の東西を区切る大宮大路には、大宮川の存在が伝えられている。ただし、発掘例はない。また、平安宮の東西幅を大宮大路を含めて測るとき、400丈となる。この場合は、大宮川の幅4丈を含めない。宮域の長さに川幅を含めないことや川幅を4丈と見るという想定も同様である。

## 参考文献

- [1] 高見友幸, 初期平安京の復原 ～都城の思想と大型将棋の将棋盤～, IIARS 論文誌, Vol.4, 18-28, 2020.
- [2] 高見友幸, 大型将棋の将棋盤と平安京の条坊: 初期平安京の復原, 大阪電気通信大学人間科学研究, Vol.23, 1-13, 2021.
- [3] 高見友幸, 古代都城の設計と天円地方の思想 - 平安京正

方形仮説の考察 - , ゲーム学会第18回合同研究会研究報告, 5-9, 2021.

- [4] 高見友幸, 周礼「考工記」の都城モデルと天照大神, ゲーム学会第19回全国大会講演論文集, 2021. (投稿中)

# 3DCG 動画及び VR の活用による手術学習支援システムの構築

佐藤 礼華<sup>†</sup> 戦 揚

<sup>†</sup>大阪電気通信大学 総合情報学部

キーワード：3DCG 動画，VR，手術学習

## 1 研究背景と目的

### 1.1 背景

医学生教育の一環として、手術室での手術に関する学習が必須であり、その中内視鏡外科手術も含まれている。例えば、泌尿器外科で行う前立腺全摘出手術の場合、「ロボット支援下腹腔鏡手術」で行うことも学習の内容になっている。

3DCG/VR 技術の発展に従い、医療教育での活用も注目されている。治療・手術支援への活用で 3DCG によるリアルな人体及び体内の器官/組織などの構成を表現することも可能になり、様々な視点視角の動的な表現も利用される<sup>[1]</sup>。

### 1.2 目的

前立腺全摘出手術を事例として医学生の手術学習の現状について調査・分析を行い、様々な問題を発見した。主に手術室での手術見学の場、見学環境の不慣れ、手術中内容確認の難点、長時間手術見学の集中力の閑散などの問題を明らかにした。これらの問題解決手法として、3DCG 動画及び VR の活用による手術学習支援システムを構築する。そして、システムの活用によって、医療現場の先端的な手術方法やその特徴などについてさらに理解しやすいよう、手術学習を効率的になることが目指す<sup>[2]</sup>。

## 2 3DCG/VR 技術の応用

### 2.1 3DCG モデル

手術学習システムに導入する 3DCG モデルは、主に手術室内の環境を表現する手術室空間、様々な医療器具、患者とする人体、体内の関連臓器などである。

#### 2.1.1 手術室/医療器具モデル

手術室や医療器具モデルの作成では、実際の手

術室現場の実物（図 1）に参考して作成した（図 2）。



図 1 手術室の現況

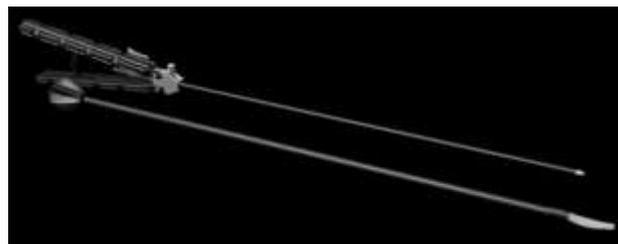


図 2 手術室と医療器具の 3DCG モデル

#### 2.1.2 人体/臓器モデル

3DCG の人体モデルでは、CT 画像から 3DCG モデルを生成修正したもの（図 3）はより精確なモデルである。そして、模型のような説明しやすいため、形や色で強調する内容を示すモデル（図 4）もある。手術のシミュレーションの内容によって使い分けになる。

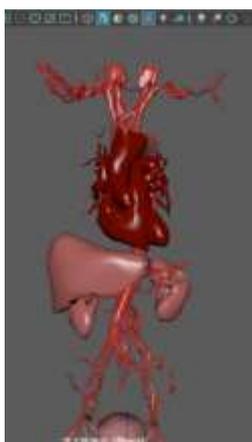


図3 CT画像によって生成した臓器モデル



図4 模型状のモデル

## 2.2 3DCG 動画表現

3DCG動画では、作成された3DCGモデルを、仮想空間中に立体的に動き回ることによって、わかりやすい視点・視角の動的な見方を設定できる。そのため、手術の手技を見やすいような動画表現が可能である。

本研究の事例としては、「内視鏡下前立腺全摘出手術」の各段階の手術内容を3DCG動画で表現するため、内視鏡カメラで撮影した同じ段階の手術内容(図5)を参考し、操作している体内の部位、操作内容などがより明確に把握できる視点を移動しながら確認できる3DCG動画(図6)を作成した。



図5 内視鏡カメラ撮影した脂肪剥離の映像

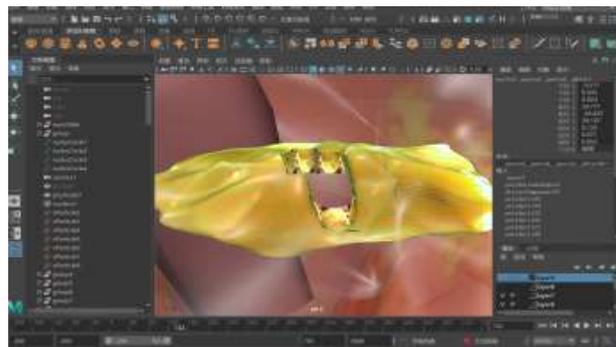


図6 実映像と同じ内容の3DCG動画の作成

視点の移動によって、手術部位の操作の前後関係にも表現可能である。例えば、操作用鉗子の「人体外部から進入の動作」(図7)から、操作が必要な部位である大腸周りまで到達する動作も表現できる。その場合、通過する場所の周辺臓器などが半透明に変化させること(図8)によって、視界を通し、操作部位と全体の臓器との位置関係も確認しやすくなる。操作部位周辺の局部と全体とも表現できる(図9)。



図7 人体外部から進入の動作



図8 通過する周辺臓器の半透明化

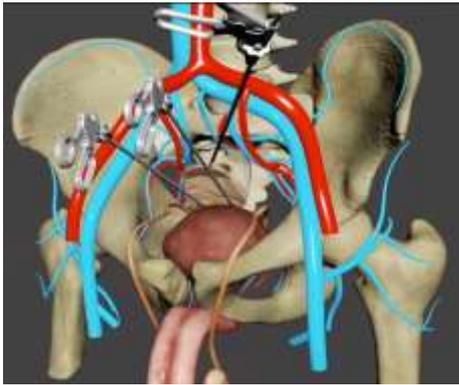


図9 脂肪剥離動作の表現

### 2.3 VR空間内の表現

3DCGモデルを利用してVR空間内の状況を表現できるので、手術室内で手術見学・学習の状況をシミュレートすることも可能である。

#### 2.3.1 手術室の環境

VR空間内では、手術室中にリアルタイムの視点移動が可能なので、様々な医療器具の配置(図10)、患者の手術時の体勢(図11)などの確認ができるように手術室の環境を構成した。



図10 手術室のVR空間



図11 患者の体勢

#### 2.3.2 体内の手術動作環境

人体の内部構成が骨組、筋肉、脂肪、臓器、血管、神経などがあり、非常に複雑である。手術時

実際に内視鏡を通して操作することで、手術学習の場合、内視鏡カメラが撮っている場面しか見れない。そのため、3DCGの臓器などのモデルを表現し、同じ操作の動作内容を見やすい視点から、視線を邪魔しているものを透過できるように動画を編成する(図12)。

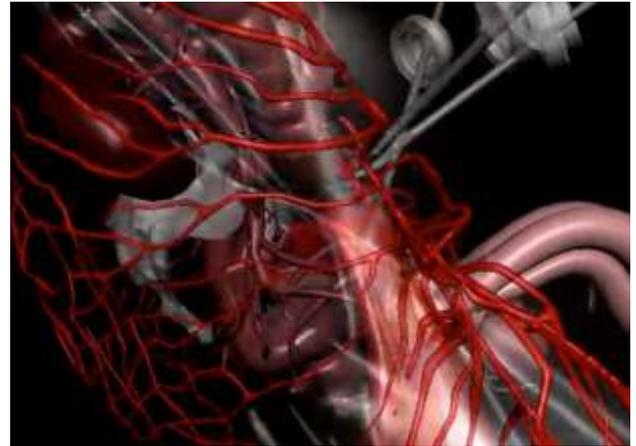


図12 操作内容のシミュレーション

## 3 手術学習支援システムの構築

手術学習支援システムには、VRを用いた仮想空間内に手術室環境、医療器具、患者とする人体と臓器などの3DCGモデル、手術内容を表現する3DCG動画や説明テキストなどを設けている。そして、実際の手術時の内視鏡カメラによって撮影された映像も同時に確認できるように設置している。

### 3.1 ユーザーインターフェース

手術学習支援システムのユーザーインターフェース(UI)は、図13、14のようにデザインされている。UI上の具体的な操作方法について、まず、VRシステムのコントローラボタンをクリックすることによって、UIメニューを開き、関連する手術の手順や各ステップの手術内容を確認できるメニューが示し、その中の表示されるボタンをクリックしコンテンツを選択することができる。

選択できる内容は、内視鏡カメラで撮影された実映像や3DCG動画映像、解説のテキストなどがあり、それぞれの内容を切り替えも簡単にできる。操作メニューを開いて、まず背景とする手術室の環境が表し、その上に各段階の手術内容のボタンや動画再生ボタンがある。



図 13 ユーザーインターフェース (UI)



図 14 選択メニューの画面

### 3.2 システムの構築

手術学習支援システムの構成内容は図 15 に示している。手術のプロセスにおける各段階の操作内容について、実際の内視鏡映像と 3DCG 動画を簡単に切り替えて確認することができる。そして、その部分の内容について解説のテキストも追加している。

VR 空間に映像と解説のテキストを挿入することで、手術室の空間内にリアルタイムで移動し、任意の視点から手術室の状態と手術の内容を同時に確認することが可能になっている。

## 4 手術学習支援システムの応用

手術学習支援システムの応用は、手術学習の問題解決手法として、①手術室内の環境をシミュレートし、どの位置、どの尺度でも確認可能にした。そして執刀医の説明がなくとも、手術内容に合わせた説明内容を学習できる。②内視鏡手術の録画を分析し、具体的な段階での手術内容を抽出し、

その内容の実映像をシステム内で確認できる（現在の事例としては、骨盤深部に入って、前立腺、尿道周りの脂肪を除去する段階の内容である）。

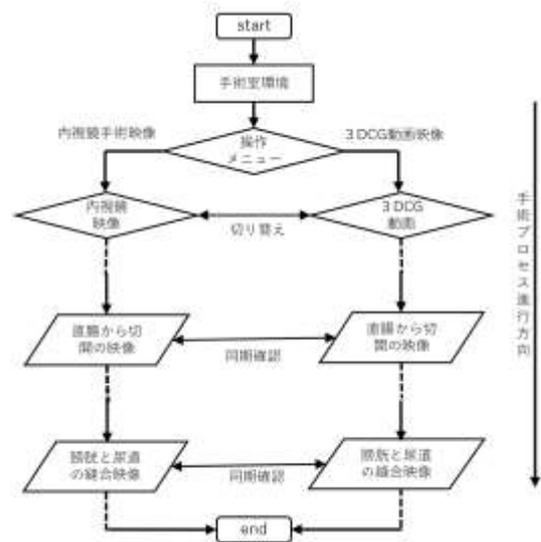


図 15 システムの構成内容

そして、実映像と同じ段階の操作を 3DCG で作成した動画と解説を同時に確認できる。③実際の手術見学では、長時間で同じ場所に立ってみることを、退屈と感じる学生は少なくない。このシステムを使って積極的に学べて、楽しめる見学ができる。それは、「何をどこで切っているか」、「どこをどのように縫うのか」、「この部分の手術のイメージはどうなっているか」などについて調べることや確認することができる。

本システムの構築には、医療関係者の指導を受け、試行錯誤をしながら行った。今後も医療関係者と連携し、医学生の手術学習のため、さらなる学習効率化を目指し、より有意義な手術見学や学習に向けての示唆を得るために、さらに多くの指導内容を支援システムに導入することを目指している。

## 参考文献

- [1]佐藤 嘉 伸 ,外科手術支援システム研究の現状と将来展望 .電子情報通信学会 誌 ,144-150,2006-02-01
- [2] Mingwei Cai, Development and Application of VR Support System for Medical Students, 2019 Nicograph International (NicoInt),pp. 5-7 ,2019

# 畳み込みニューラルネットワークと完全畳み込みネットワークを用いた

## 大腸生検診断支援システムの開発

田中敏幸 小林寛知  
慶應義塾大学 理工学研究科

### 1. 研究背景・目的

高齢化に伴い、現在日本における癌患者の死亡数は増加傾向にある。その中でも、大腸癌は日本における癌の罹患数で1位であり、死亡者数でも女性1位、男性3位と大きな割合を占めている。癌と疑わしい場合は細胞診断することで、癌が進行する前に早期発見することができる。その一方、病理医不足は深刻である。病院には病理医が一人しかおらず、すべての生体組織診断（生検）を診断せざるを得ないケースが多発している。更に、誤診断の防止のために一つの生検に対して、ダブルチェック、トリプルチェックする事が強く求められるが、このような背景から一度しかチェックが行われない状況になっている病院も多い。このままでは、病理医 1 人あたりの負担は増大するばかりか、誤診断率の増加や早期発見できた癌の発見が遅れてしまう等の問題につながってしまう。そこで、以上の問題を解決するべく、ダブルチェックやトリプルチェックに使用可能な大腸癌自動診断支援システムの開発が必須である。先行研究[1]では細胞核細胞質比（NC 比）、腺管偽重層、腺腔形状といった病理組織学的指標を再現した特徴量により正常・腺腫・癌の分類を行っているが、像全体から特徴抽出を行っているために、良性と悪性とが混在する場合に判別結果が過小評価され、良性側に希釈されてしまうという問題があった。もう一つ[2]は画像をパッチ分割し学習を行っている。画像認識の手法として畳み込みニューラルネットワーク（CNN）を用いていたが、訓練データにおいて背景が混在していたり、Groupの特徴を表さない画像が含まれていた。またCNNのネットワーク構造が一律であり、訓練が最適化されていたか疑問が残る。そこで本研究では、大腸の正常細胞(Group1)・腺腫(Group3)・癌(Group5)の3種類の細胞診断用データを用いて、畳み込みニューラルネットワーク(CNN)の更なる最適化と、新手法である全畳み込みニューラルネットワーク(FCN)との比較を行う。そして大腸癌診断のトリプルチェックに使える診断支援シ

ステムの開発を行う事を目的とすると共に、支援システムのウェブアプリケーション化を行う。

### 2. 方法

#### 2.1 入力画像

この研究に使用されている学習画像は、大腸癌病理医によって判別された114枚の画像を元に作成された。CNNが学習する画像として200×200ピクセルの正方形の面積のパッチを作成した。アルゴリズムを使用して背景が混在する画像を取り除くと共に、左右の画像反転による水増しを行った。CNNデータセットの総数は7440枚となった。

次に、FCNの訓練画像としてJPEG形式の入力画像と、対になる分類ラベルを持つインデックスカラー画像を作成した。インデックスカラー画像とは1ピクセル毎にラベル付けされたデータと、RGB値を表すカラーマップで構成されるPNGファイルである。大腸がんの教師画像として採用することにより、各ピクセルには各Groupおよび背景からなる4つのインデックスを持つ。

#### 2.2 CNNアーキテクチャ

CNNは畳み込み層とプーリング層が交互に積層された順伝搬型ニューラルネットワークである。畳み込み層は画像の局所特徴を抽出し、プーリング層は局所領域における画素値を集約させる処理を行う。使用するCNNのアーキテクチャとして、VGG16ベースの転移学を行う。VGG 16は、先行研究で用いられたAlexNetよりもエラー率が大幅に改善したモデルである。且つ、ResNet等よりディープなネットワークでは学習が非常に遅くなるがVGG 16は学習スピードも速い。今回はノードの数と隠れ層数を変更した9パターンのモデルで実験を行った。各全結合層の詳細をTable 1に示す。

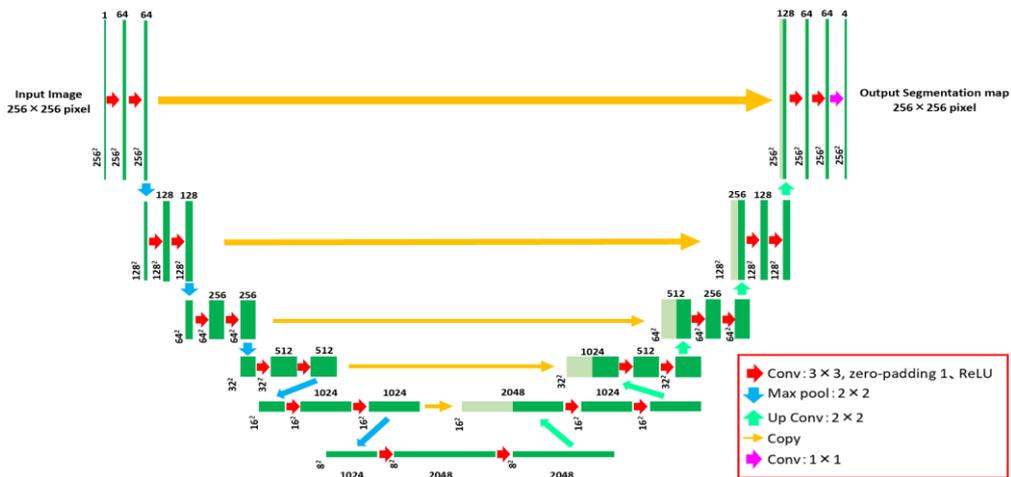


Fig. 1 5層のプーリング層を持つU-NetベースのFCNアーキテクチャ

Table 1 各全結合層におけるノードと隠れ層数

|                               | Number of hidden layers |         |              |
|-------------------------------|-------------------------|---------|--------------|
|                               | 1                       | 2       | 3            |
| Number of nodes in each layer | 64                      | 64/128  | 64/128/256   |
|                               | 128                     | 128/256 | 128/256/512  |
|                               | 256                     | 256/512 | 256/512/1024 |

Table 2 CNNにおける隠れ層と判別精度の関係

| Number of nodes in the last hidden layer | Number of hidden layers |       |       |
|------------------------------------------|-------------------------|-------|-------|
|                                          | 1                       | 2     | 3     |
| 64                                       | 95.09                   | 95.09 | 94.97 |
| 128                                      | 95.22                   | 95.09 | 94.34 |
| 256                                      | 95.09                   | 94.47 | 95.09 |

### 2.3 FCNアーキテクチャ

完全畳み込みネットワーク (FCN) の構造は、畳み込み層と全結合層からなるCNNとは異なり、全てが畳み込み層で構成される。出力として、CNNは各分類における確率が出力されるのに対し、FCNは2次元画像が出力される。FCNのモデルとして、U-Netベースのネットワークを構築した。U-Net [3]は細胞画像の分類に用いられている。

プーリング層数を2~5層まで変化させる事で4種類のモデルについて実験を行った。プーリング層が5層の時のモデルをFig. 1に示す。

## 3. 結果

まずTable 2にCNNの全結合層における隠れ層数の変化と判別精度の関係を示す。各30エポックの学習を行った結果、上記の9つの分類精度は約95%の値を示した。黄色の箇所は最高精度であり、95.22%の精度となった。

次にTable 3にFCNのプーリング層数と判別精度の関係を示す。プーリング層数が多いという事は、よりディープなネットワーク構造であると言える。同じく黄色の箇所が最高精度であり、プーリング層5層の時の最大精度96.52%となった。

Table 3 FCNにおけるプーリング層数と判別精度の関係

| Number of pooling layers | Maximum accuracy in each learning |
|--------------------------|-----------------------------------|
| 2                        | 95.95                             |
| 3                        | 96.17                             |
| 4                        | 96.08                             |
| 5                        | 96.52                             |

この結果から、CNNとFCNの両者において非常に高い判別精度を得る事に成功した。それぞれの強みとしてCNNは局所領域の判別に強く学習スピードが速い。FCNは病理医に対して視覚的に癌領域を提示することが出来る。

次に、上記のCNNを元に開発したウェブアプリケーションをFig. 2に示す。



Fig. 2 CNNにおけるウェブアプリケーション

#### 4. 結論と今後の展望

本研究では、CNNとFCNを使用した大腸生検画像の自動診断システムを提案した。

結果として、VGG16ベースのCNNにおける最高精度は95.22%であり、U-NetベースのFCNの精度は96.52%となり、両手法を用いたトリプルチェックに使用できる診断支援システムの可能性を示した。また、CNNのウェブアプリケーション化を行い、診断支援システムの原型を示した。

今後は、FCNの学習安定性をあげると共に、CNNとFCNを合わせた診断支援システムのウェブアプリケーション開発が必要となる。

#### 参考文献

- [1] Y. Harai and T. Tanaka, "Automatic diagnosis support system using nuclear and luminal features," 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8, 2015.
- [2] M. Nagao and T. Tanaka. "Computer-Aided Diagnosis System of Colon Cancer Biopsy Based on Covolutional Neural Network", The 16th International Conference on Biomedical Engineering, 2016
- [3] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, vol. 9351, pp. 234-241, 2015.

# ハイブリッド形式の国際会議における参加費の設定に関する一考察

次郎丸 沢†

†株式会社カンファレンスサービス

キーワード：オンライン会議，タイムマネジメント，司会，背景動画

## 1 はじめに

COVID-19 の流行により，2020 年は学術会議を対面で行った学会は 2020 年 4 月に行った日本感染症学会[1]など数える程度しかない．学会の開催状況について正確な統計を取ってはいないが，2020 年に開催予定であった学会の多くは中止もしくは第 5 回国際 ICT 利用研究学会全国大会（以下，本会議と表記）と同様にオンラインでの開催となった．また，弊社が担当した学会はすべてオンラインでの開催となった．

2021 年は Virtual の会議と対面方式の会議を同時進行で行う，いわゆるハイブリッド形式の国際会議が増えてきたが，参加費の設定については事例が少ないことから，筆者はハイブリッド形式の国際会議における参加費の設定について調査し，これまでに聞いた各ステークホルダーの意見なども参考に考察を行った．

## 2 ハイブリッド形式での学会開催

本稿では詳しい説明を省くが，ハイブリッド形式での学会開催の場合、ハイブリッド形式を行うための機材や人員配置のためにコストがアップする一方で，Virtual 形式で参加する方に提供する施設利用料や飲食費に関するコストはダウンする．

総合的なコストが上がるか下がるかは，ハイブリッド形式にかかるコストや施設利用料や飲食費の多寡で変わってくるので一概に断言することは出来ない．

## 3 調査方法

本稿では ACM (Association for Computing Machinery) の Calendar of Events を使用した[2]．そこで以下の条件に合致する国際会議 (Workshop を

含む) を抽出した．

- 開始日が 2021 年 3 月 1 日～2021 年 4 月 30 日まで
- 参加費がウェブサイト等から確認できる
- 同系列の会議は 1 種類とカウント

ここで，同系列の会議は「ウェブサイトのレイアウトが全く同じで，かつ参加費が全く同じ会議を同系列の会議」と定義した．

なお，参加費の種類は①著者 (Author)、発表者 (Presenter)、聴講者 (Listener) の 3 種類を抽出した．著者 (Author) とは発表論文を掲載して発表をする者、発表者 (Presenter) とは発表論文を掲載しない、もしくは Abstract のみ掲載して発表をする者、聴講者 (Listener) とは発表をせずに参加をする者のことである．

また，参加区分は募集時期が複数設定している場合は最も早い時期の金額を使用し，参加者区分は学術系で最も高い金額を抽出した．例えば，国際会議 ICCEEG2021 の参加費には Early Bird と Regular の 2 区分が設定されているが，Early Bird の方が募集時期が早いため Early Bird の参加費を抽出し、さらに参加費の区分として Regular と Students の 2 区分が設定されているが，Regular のほうが高額であるため，Regular の料金を抽出する [3]．

## 4 結果と考察

### 4.1 会議形式別会議開催数

調査した結果，合計 106 の国際会議および国際 workshop のデータを得ることが出来た．会議形式別開催数とその割合を表 1 に示す．ハイブリッド

形式での開催が 59.4%となっており、ハイブリッド形式の国際会議が調査期間内では過半数となった。2021年3月から4月ではハイブリッド形式が主流であると言える。

表 1 会議形式別開催数と割合

|         | 開催数 | 割合     |
|---------|-----|--------|
| Hybrid  | 63  | 59.4%  |
| Virtual | 36  | 34.0%  |
| 対面      | 7   | 6.6%   |
| 合計      | 106 | 100.0% |

会議形式別参加費種別数の割合を表 2 に示す。Hybrid では 90.5%の会議が 3 種類の参加費を設定しているのに対して、Virtual では 63.9%の会議が 1 種類しか参加費を設定していないことが分かる。

表 2 会議形式別参加費種別数割合

|         | 1 種類  | 2 種類  | 3 種類  |
|---------|-------|-------|-------|
| Hybrid  | 0.0%  | 9.5%  | 90.5% |
| Virtual | 63.9% | 25.0% | 11.1% |
| 対面      | 14.3% | 42.9% | 42.9% |
| 合計      | 0.0%  | 9.5%  | 90.5% |

#### 4.2 ハイブリッド形式で開催された国際会議の参加費

ハイブリッド形式で開催された 63 会議の参加費を、対面での参加費を基準としたときの Virtual での参加費の割合とその差額の平均を表 3 に、ダイアグラムを図 1 に示す。ハイブリッド形式における参加費は、対面での参加費より Virtual での参加費の方が平均 1 万円～1 万 2 千円程度減額されていることが分かる。

表 3 ハイブリッド形式の会議における対面と Virtual の参加費の差額と割合

|     | 割合    | 差額(円)    |
|-----|-------|----------|
| 著者  | 78.3% | 11,837 円 |
| 発表者 | 75.0% | 10,089 円 |
| 聴講者 | 58.3% | 10,230 円 |
| 平均  | 73.7% | 11,135 円 |

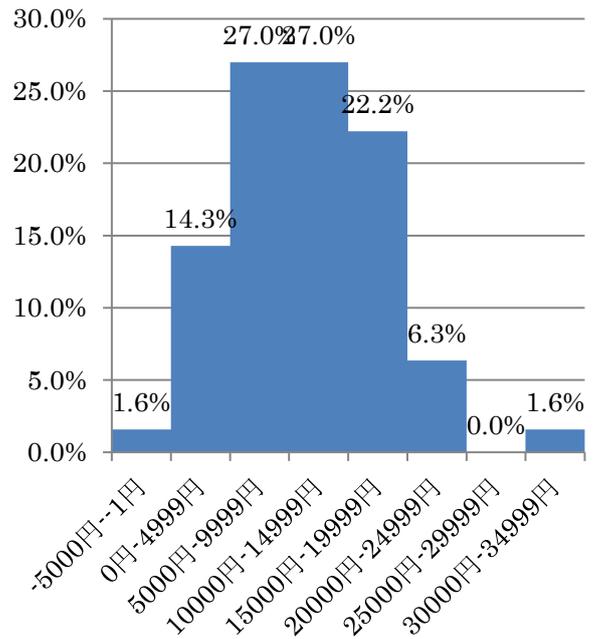


図 1 対面の参加費に対する Virtual での参加費の差額 (全体)

一方で、著者の参加費のみを抽出して作成したダイアグラムを図 2 に示す。最頻値は図 1 と大きく変わらないが、対面の参加費と Virtual での参加費が同一である会議が 8 会議認められた。この 8 会議のウェブサイトには、「来ることが出来ない人のために Virtual の環境も用意している」という文言が書かれていることが共通している。

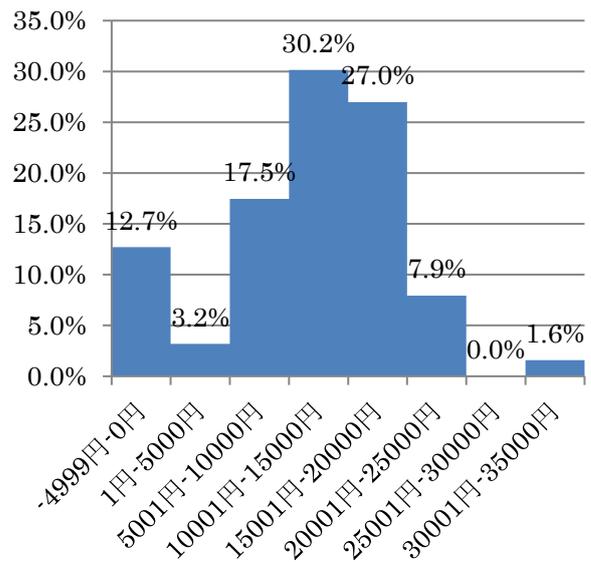


図 2 対面の参加費に対する Virtual での参加費の差額 (著者)

これは、会議としての基本方針が対面方式であるが、来ることが出来ない人に対しても Virtual の環境を用意している、という視点なのだろうと予想している。もしくは、ハイブリッド形式を行うことによるコストと施設利用料および飲食費のコストがバランスしているという事かもしれない。

最後に、聴講者の参加費のみを抽出して作成したダイアグラムを図 3 に示す。聴講者の参加費は著者の参加費よりダイアグラムの山の数が多く、会議の財政面や参加者の増加を見据えながらそれぞれの結論を出したのだと予想している。

ここで「入場制限」と記載している 14.3% (7 会議) のうち 6 会議は、参加者を対面のみに限定しており、1 会議は参加者を Virtual での参加のみに限定している。参加者の参加手法を限定する方法は、COVID-19 の対策になるほか、ハイブリッド方式に掛かる追加費用を抑える事にも役立つ。

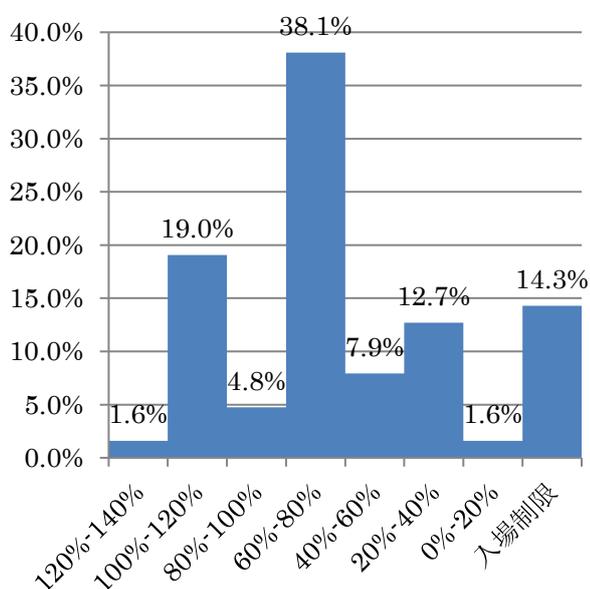


図 3 対面の参加費に対する Virtual での参加費の差額 (聴講者)

## 5 まとめと今後の展望

本稿ではハイブリッド形式の国際会議開催における参加費の設定について調査し考察した。今後は定点観測を続けるとともに、他会議のハイブリ

ッド開催の事例を集めて、より良いハイブリッド形式の国際会議開催方法を検討していきたい。

## 参考文献

- [1] 第 94 回日本感染症学会総会・学術講演会, <https://www.societyinfo.jp/jaid2020/> (2020 年 11 月 10 日閲覧)
- [2] ACM, “Calendar of Event”, <https://www.acm.org/calendar>, (2021 年 3 月 15 日閲覧)
- [3] Registration Instruction, ICEEG 2021, <http://www.iceeg.org/reg.html> (2021 年 3 月 15 日閲覧)

# Eclipse 統合環境における PARI Library を使った C プログラミングとその応用

永田 清

大東文化大学経営学部

nagata@ic.daito.ac.jp

キーワード PARI/GP, データの完全性, 誤り訂正符号, Self-Dual Code

## 1 はじめに

パーソナルコンピュータが普及し、スマートファンやタブレット端末をほとんどの国民が所有するようになった現在、コンピュータ自体というよりもその処理能力が遍在する Ubiquitous Computing が現実となった。また、クラウドコンピューティングを通じた IoT(Internet of Things) の時代には、ハードウェアの存在自体が認識されなくなるのかもしれない。しかし、どこかにはプログラムがあって常に様々な処理を行っているはずであり、AI(artificial intelligence) が再帰的に増殖を繰り返すとしても、それを作成・発展させる必要はあるだろう。

本稿の目的は、発展を遂げてきた数式処理ソフトウェア PARI のライブラリを、既存のプログラミング言語である C 言語に組み込んでプログラム作成を行う手順と、その応用例として筆者らによる誤り訂正符号の生成アルゴリズムの実装例を示すことである。第 1, 2 節では、コンピュータとプログラミング言語の歴史を概観することでさまざまなプログラミング言語の特徴などを把握する。第 3 節ではプログラミング統合開発環境の一つである Eclipse と PARI/GP について、その概略と設定方法などを解説する。第 4 節では、C 言語で PARI Library を使う場合の留意点や関数仕様を見る。第 5 節は、誤り訂正符号の生成アルゴリズムであり、第 6 節で実際のプログラムの一部を紹介する。第 7 節は結論と今後の課題である。

## 2 コンピュータとプログラミング言語の歴史

まず、コンピュータの歴史と、プログラミング言語発展の流れなどと観ていくこととする。

### 2.1 自動計算器としてのコンピュータ

現代のコンピュータの先駆けとなった電子計算は 1940 年代に開発され、当初はその多くが特定の目的で使用さ

れた。英国では Tommy Flowers の設計によって 1944 年に Colossus(Mark II) が完成し、対戦国ドイツの暗号機 Lorenz SZ40/42 による暗号文を解読するために使われた。米国において John William Mauchly と John Adam Presper Eckert Jr. が考案と設計を行い、1946 年に完成した ENIAC(Electronic Numerical Integrator and Computer) は、陸軍弾道研究所の支援を受け砲撃射表と呼ばれる表の作成を目的として作成された。開発メンバの一人であり、ENIAC のプログラムを担当した Adele Goldstine の夫でもある Herman Goldstine によれば、ENIAC はマンハッタン計画で有名なロスアラモス研究所の問題を計算するために使われた ([8])。同じく米国では海軍が「Project Whirlwind」として資金援助し、MIT (Massachusetts Institute of Technology) 研究所の Jay Forrester 達が開発したフライトシミュレーター制御用のコンピュータ Whirlwind が 1951 年に動作しており、リアルタイムで動く世界最初のコンピュータといわれている (<http://museum.mit.edu/150/21>)。Whirlwind は「つむじ風」という意味で、空軍が運用していた旧ソ連の原爆搭載爆撃機を発見、追跡、要撃するために自動化されたコンピュータシステムである SAGE(Semi-Automatic Ground Environment) に使われたとのことである。

Mauchly と Eckert はプログラム内蔵能力のほとんどない ENIAC に対し、水銀遅延線 (delay line memory) によってデータとプログラムを格納するような電子計算機的设计を試み、1951 年に EDVAC(Electronic Discrete Variable Automatic Computer) を稼働させる。彼らはマンハッタン計画にも参加していた John von Neumann を顧問として迎え、彼がまとめた文書「First Draft of a Report on the EDVAC」がケンブリッジ大学の Maurice Vincent Wilkes 達に影響を与え、初期のプログラム内蔵方式による電子計算機となる EDSAC(Electronic Delay Storage Automatic Calculator) を開発した。EDSAC では、Wilkes の書いた 1 から 100 までの二乗と、200 まで

の素数を求めるプログラムが1949年に動いている ([20]).

英国の食品会社 J. Lyons and Co. は、業務上の問題解決の可能性を電子計算機に見出し、EDSAC をベースにしたコンピュータ LEO(Lyons Electronic Office) を開発し、1951年にはベーカリーの経営を評価する Barkery Valuations をコンピュータ化している。LEO のプログラミング言語 CLEO(Clear Language for Expressing Orders) は、COBOL のような言語といわれている。因みに、1954年に設立された LEO Computers Ltd. は、いくつかの経緯を辿り 1990年には Fujitsu に買収された。

ケンブリッジ大学 Computer Lab の資料 ([20]) には、EDSAC が “the world’s first stored-program computer to operate a regular computing service” とあるが、米国マンチェスター大学でやはり von Neumann から影響を受けた Tom Kilburn, Freddie Williams, Geoff Tootill 達によって開発された Manchester Small-Scale Experimental Machine(SSEM, 通称 Manchester Baby) が 1948年6月に稼働していた。SSEM のハードウェアは減算と正負反転だけを実装していたが、そこにプログラムを内蔵して  $2^{18} = 262,144$  を割り切る自分自身でない最大の整数、つまり  $2^{17} = 131,072$  を求める計算を行った ([13])。その後 SSEM は Manchester Mark I などを経て、世界最初の商用汎用コンピュータといわれている Ferranti Mark I に引き継がれることとなった。Manchester Baby に関しては、User Guide が 2001年に Warwick 大の David Sharp によって書かれており、その概要を知ることができる (<https://www.davidsharp.com/baby/babyuserguide.pdf>)。

コンピュータ (電子計算機) の定義要素はさまざま、電気を動力とする、数値表現が 2 進法か 10 進法か、プログラムを内蔵できる (いわゆるノイマン型コンピュータ) かどうか、Alan Mathison Turing による万能 Turing Machine と同じ計算能力を持つ (Turing-Complete) かどうか ([23])、汎用性があるかどうか、等が挙げられている。

上記のものは全て電動であるが、自動で複雑な計算を行えるものならば 1822年に Charles Babbage の設計による階差機関があり、その後詩人として有名な George Gordon Byron の娘でもある Augusta Ada King(Ada Lovelace) が解析機関においてベルヌーイ数を求めるプログラムを書き、世界初のプログラマであるといわれている ([6])。

1950年代にその多くが登場した電動による計算機以外にも、アイオワ州立大学の John V. Atanasoff と Clifford E. Berry(<https://jva.cs.iastate.edu/>) による ABC (Atanasoff-Berry Computer) が 1939年に稼働し、ドイツの Konrad Zuse による Z3 は 1941年に稼働しているが、現在のコンピュータの原型と呼ばれるプログラム内

蔵式 (ノイマン型) の登場には、どのようなハードウェアでそのプログラムを保持するのかといった問題の解決を待たなければならなかった。

## 2.2 コンピュータプログラミングと言語

初期のコンピュータはそれぞれの目的に特化して設計・作成され、各コンピュータに対しさまざまな方法でプログラムがなされており、Ada Lovelace が最初のプログラマだとしても彼女が現在のようプログラム言語を駆使してプログラムを行ったわけではなく、一定のアルゴリズムを実行するために計算機の仕組みを変更していたものと考えられる。Colossus や ENIAC においても、プログラムはプラグ盤やスイッチの操作によって配線を変更することであった。

一方プログラム言語自体は Hilbert の形式主義に触発されたといわれている Konrad Zuse が、論理回路を表すための記法と構造図式を作り、1943年から 1945年頃には “高級” プログラミング言語の先駆けとも言える “Plankalkül” を開発し ([1], [7])、1948年の論文 “概略的に組み合わせられたタスクを定式化する手段としての一般 Plankalkül について” を公表した ([24])。

1940年代の機械語やアセンブリ言語は、判別や改変のやり難さ、表現の煩雑さ、個々のコンピュータ特性に偏っている、などさまざまな問題点を持っていたが、1950年代以降はいわゆる高級プログラミング言語と呼ばれるものが多く開発されてきた。

一般に、世界初の高級プログラミング言語と呼ばれている FORTRAN(FORmula TRANslator) は、John Backus によって IBM のメインフレームコンピュータ IBM 704 用に開発され、主に科学・工学分野の計算に貢献し、さまざまな変更を経て現在 Fortran 2018 となっている。

CLEO の流れを汲む COBOL(Common Business Oriented Language) は、CODASYL(Conference on Data Systems Languages) によって 1959年に開発された共通事務処理用言語であり、自然言語 (英語) による表現が使われている。

IPA が日本国内の IT 企業に対し行った調査によると、2016年から 2017年におけるソフトウェア開発 1,237 件に用いられた言語で COBOL は Java の 42.7% に次いで 2 位 (13.3%) となっている。因みに、3 位以降は C#(8.1%)、C 言語 (7.3%)、Visual Basic.NET(6.8%)、C++(5.5%) である<sup>1</sup>。2020年度の基本情報技術者試験から COBOL が

<sup>1</sup> 「ソフトウェア開発データ白書 2018-2019」, p.50, <https://www.ipa.go.jp/files/000069381.pdf>

なくなり、過去の言語といった印象もあるが、COBOLで教育を受けたプログラマーがまだ多くいることや、使われているシステム自体がCOBOLでプログラムされていることなどが影響していると考えられる。実際、それらの200件近いソフトウェア開発においても、メインフレームにおけるバッチ処理(約60件)だけでなく、クライアントサーバシステムにおけるオンライントランザクション処理(約50件)にも多く使われていた。

1975年以降にBill Gates(William Henry Gates)とPaul Gardner Allenによって、当時販売され出したPC(Personal Computer)に組み込まれたことで有名なBASIC(Beginner's All purpose Symbolic Instruction Code)は、1964年に米国Dartmouth CollegeのJohn George KemenyとThomas Eugene Kurtzによって開発された。これら以外にも、1950年代にKenneth Eugene Iversonによる、配列処理などに特徴を持つ言語APL(A Programming Language)([11])や、FORTRAN, COBOL, Algolなどの特徴を備えた言語としてIBMなどにより1964年に開発されたPL/I(Programming Language One)といった、いわゆる手続き型言語の多くが現れている。

1950年代後半には、構造化プログラミングといった概念が提唱され、主に“go to”文とその行き先を示すラベルなどによって処理を制御することでプログラムが複雑になるといった、いわゆる“Spaghetti Code”への批判が起こった。

Edsger Wybe Dijkstraは、1969年に“Structured programming”というタイトルの論文([4])において“programs as neckless strung from pearls”とあって表現で、構造化プログラミングを譬えている。Dijkstraは構造化プログラミングを一言で表した定義を与えていないようだが、Ole-Johan Dahl, Charles Antony Richard Hoareと共に1972年著わした“Structured Programming”([2])では、1章においてDijkstraが全体像を、第2章でHoareがデータ構造について、第3章ではDahlが構造の継承などに関して記述しており、構造化プログラミングの普及に大きな影響を与えた。

Dijkstraは、1975年当時に使われていた上記のいくつかのプログラミング言語に対し痛烈な文章を残している([5])。例えば、FORTRANを“the infantile disorder”, PL/Iを“the fatal disease”と呼び、COBOLは“cripples the mind”であり“be regarded as a criminal offence”とまで述べている。また、BASICによって教育を受けた学生は“as potential programmers they are mentally mutilated beyond hope of regeneration”となり、APLは“the language of the future for the programming techniques

of the past”であって“a mistake”との主張である。

実際の構造化プログラミングは、順次構造(begin-end)、判別(選択)構造(if-then-else)、繰り返し構造(for, while-do)を使って行なわれ、結果として“go to”のない(または少ない)プログラムとなるが、Donald Ervin Knuthが指摘しているようにDijkstraの論文([4])には“go to”文の記述はなく、彼の率直な関心事は“For what program structures can we give correctness proofs without undue labor, even if the programs get larger?”であると述べている[14]。Richard C. Linger, Harlan D. Mills, およびBernard I. Wittによる1979年の著書“Structured Programming: Theory and Practice”では、構造化プログラムを以下のように定義している([16], p. 118)。

**定義 1** *A structured program is a compound program constructed from a fixed basis set of prime programs.*

ここで、“fixed basis set”となるのは、次の定理([16]pp.118-119)に示され順次構造、判別(選択)構造、繰り返し構造である。

**定理 2 (Structure Theorem)** *Any proper program is function equivalent to a structured program with basis set {sequence, ifthenelse, whiledo}, using functions and predicates of the original program and assignment and tests on one additional counter.*

構造化プログラミングを念頭において開発されたプログラミング言語の草分け的存在は、上記の3人以外にも人工知能やLISP(List Processing language)開発でも有名なJohn McCarthy, PascalやModula-2開発を行ったNiklaus Wirthなど多くの優秀な研究者が関わったALGOL(ALGOrithmic Language)である。ALGOLにはいくつかのバージョンがあり、ALGOL68はPascal, C, およびAdaなどに強い影響を与えたと言われ、またその後Simulaなどのオブジェクト指向プログラミング(OOP; Object Oriented Programming Language)言語の基礎となっている。

OOP言語は、1972年にSmalltalkを開発したAlan Kayが初めて使った言葉だと言われており、データ(フィールド)と処理(メソッド)を併せ持つ構造設計図であるクラスとその実態となるインスタンス(オブジェクト)を扱う。Kristen Nygaardと共にSimulaを開発したOle-Johan Dahl([3])によると、OOPの一般的な定義は存在しないようであるが、C++, Java, C#などのコンパイラ言語が代表格とされるクラスベースのOOP言語ではカプセル化、抽象化、継承、ポリモーフィズムが重要な要素と

されている。また、1994年にver 1.0がリリースされたPython, 1996年に公表されたJavaScript, Rubyなどはインスタンスベース(プロトタイプベース)のOOP言語と呼ばれている。

### 3 C言語と統合開発環境

前節でみたように、ハードウェアとしてコンピュータも、コンピュータを動かすソフトウェア開発のためのプログラミング言語も、さまざまな目的、用途、思想などのもとで発展してきた。本論文の目的は、数式処理を汎用プログラミング言語によって行うことであり、またそのための環境を構築しておくことでもある。数式処理のためのパッケージについては後に述べるが、ここではC言語とプログラミングをスムーズに行うための統合開発環境の一つであるEclipseについて観ておこう。

C言語の系譜は、1966年にケンブリッジ大学のMartin Richardsによって設計された手続き型かつ命令型の構造化プログラミング言語BCPL(Basic Combined Programming Language)に遡る。1969年頃には、AT&Tベル研究所のKen ThompsonによるB言語が登場した。その開発にもかかわったDennis MacAlistair Ritchieが主体となり、AT&Tベル研究所でUNIX OS実装のため、1970年代に開発された言語がC言語である([21])。判別(選択)構造として“if-else if-else”, “switch”, 繰り返し構造“do/while”, “for”, “while”などの制御構文を持っており、その点では高級言語といえるが、ポインタ変数などによってメモリへ直接アクセスできることから低水準言語の特徴も持っている。Brian W. KernighanとDennis M. Ritchieによる“The C Programming Language”の初版が1978年に出版され、1988年の第2版ではANSIでの標準化が反映されている([12])。これらは石田晴久氏によって日本語訳され、C言語の教科書的な役割を果たした。

C言語は、歴史的には古い但现在も使われているプログラミング言語であり、多くの標準ライブラリが提供され、それらをヘッダファイルとして取り込む(#includeする)ことで、様々な機能を簡単に使うことができる。

このようなコンパイル型処理系では、プログラミング言語で記述されたソースファイルを、プリプロセス、コンパイル、アセンブル、リンクなどのいくつかの段階で処理して実行可能はファイルを作成するが、プログラム作成時にそれらを特に意識する必要はない。また、プログラム作成には文字を入力するためのエディタソフトウェアを使い、ソースファイルの保存、編集、削除などの処理も

必要になる。高機能エディタのいくつかはオープンソースとして無料で提供されており、それにC言語のコンパイラを結びつけて言語の学習やプログラム作成が行えるが、現在では統合開発環境(IDE: Integrated Development Environment)として無料のアプリケーションソフトウェアがいくつか提供されている。無料ではあるが、一般的なエディタだけでなく、入力ミスや構文エラー表示、自動補完・修正、実行時の監視など様々な機能が組み込まれており、ここではその一つであるEclipseを使うこととする。

Eclipseは、1990年代初頭にIBMによって開発され、プログラミング言語に対応した統合開発環境VisualAgeに由来しているが、2000年以降は非営利組織であるEclipse Foundationによって無償のオープンソースソフトウェアとしてEclipse Software Development Kit(SKD)がリリースされている。VisualAgeはSmalltalkによって実装されたが、EclipseはJava言語によっているので、Java言語によるプログラム作成に適しているが、最近ではC/C++, PHP, Pythonなど様々な言語をプラグインによって扱える統合開発環境となっている。基本は英語バージョンであるが、Eclipse Foundationにおける抱卵的プロジェクトであるBabel(<https://www.eclipse.org/babel/>)では、多言語対応を行っている。また、Mergedoc Project(<https://mergedoc.osdn.jp/>)では、さまざまなバージョンのEclipseがありC/C++, PHP, Pythonに対応し、Windows版やMac版はAll In One(AIO)パッケージとしてダウンロードして使える。ここでは、後者のWindows版でC/C++パッケージを使うこととし、以下にインストールとC言語によるプログラム作成手順を簡単に解説する。

#### 3.1 EclipseのインストールとC言語によるプログラム作成・実行

WindowsにEclipseのAIOパッケージを解凍するには“7-Zip”を使うので、まずページにあるリンクによって“7-Zip”をダウンロードしておく。次に、Windows64ビットFull Edition C/C++の“Download”(図1)をクリックし、表示されるリンクからZipファイルをダウンロードし、そのファイルを右クリックして7-Zipで適当な場所(MyDocumentフォルダなど)に解凍すればよい。

Eclipseの実行ファイル(eclipse.exe)は、“¥Documents ¥pleiades-2020-12-cpp-win-64bit-jre\_20201101¥pleiades ¥eclipse”フォルダ(図2)にあるので、特別な設定などの必要がなくダブルクリックなどによって起動する。最初

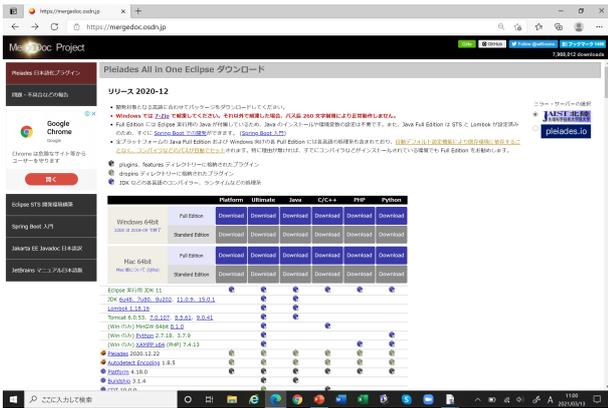


図 1: MerGEDoc Project のトップページ

は Work space(ファイルを保存するフォルダ)を聞いてくるので、適当な場所を指定する。

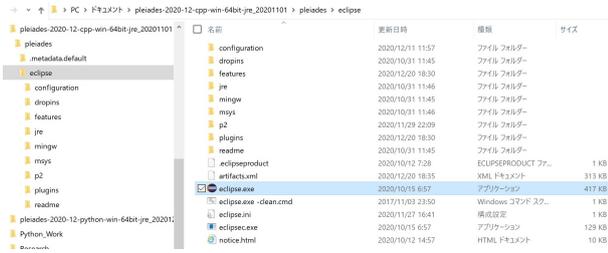


図 2: Eclipse の実行ファイル

Eclipse で C 言語によるプログラムを作成するには、チュートリアルウィンドウを閉じ、[ファイル]-[新規] から“C/C++プロジェクト”を作る(図 3)。

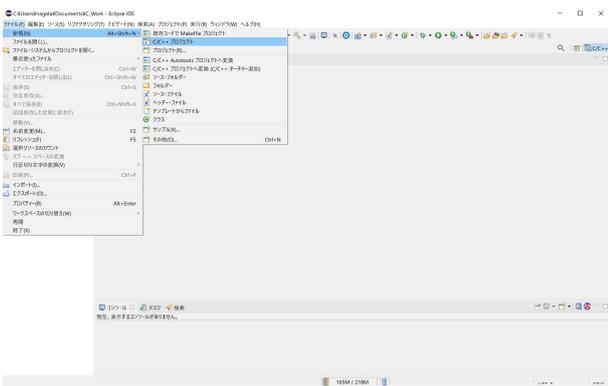


図 3: Eclipse で C/C++プロジェクト作成

次に“C Managed Build”ウィンドウを選び、プロジェクト名を入力すると、「プロジェクト・エクスプローラー」にそのプロジェクトが表示されるので、[ファイル]メニューまたは右クリックによって表示される [新規] から“ソース・ファイル”を選び(図 4)、適当な名前(.c)を付けて作

成するとエディター画面が現れるので C 言語によるプログラムを書いていくことになる。プログラムが書けたら、[保存]-[ビルド]-[実行]の手順で進み、結果は結果は「コンソール」(図 4 では右下)に表示され、実行ファイル自体はプロジェクトフォルダ内の“Debug”フォルダに入る。

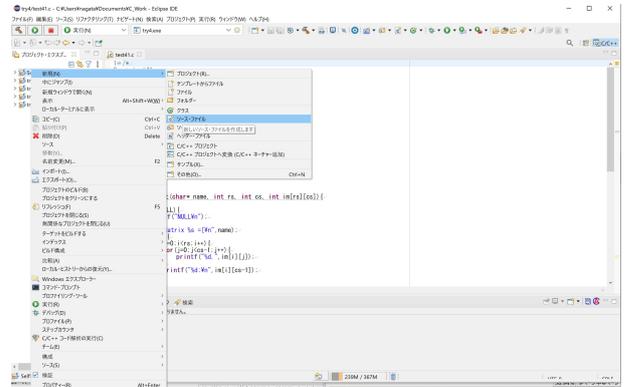


図 4: Eclipse で C 言語のソースファイル作成

### 3.2 数式処理ソフトウェア PARI/GP

コンピュータを使って数式を処理し、数値計算やグラフィック表示などを行うソフトウェアを数式処理システム (Formula Manipulation System) と呼ぶ。1963 年に Stanford 大学で理論物理学のポスドク研究員だった Anthony C. Hearn が、John McCarthy の助言によってプログラミング言語 LISP を使って開発を始め ([10])、数年後に作成された REDUCE が“portable general-purpose computer algebra system”として有名であり、現在も開発が行われている (<http://www.reduce-algebra.com/index.php>)。

商用版としては Mathematica, Maple, Magma, MATLAB などのかなり高価なものもあるが、RARI/GP, Risa/Asir (<http://jp.asir.org/>) のようなオープンソースのフリーソフトウェアもある。最近では、上記の商用版ソフトウェアの代替となるようなフリーソフトウェアを提供しようといった Sage プロジェクトのようなものも立ち上がっている (<https://www.sagemath.org/>)。

本研究では、RARI Library を使うので、ここで RARI/GP について簡単に観ていくこととする。RARI/GP は、1979 年に Bordeaux 大学の Henri Cohen と François Dress による代数計算インタープリタ ISABELLE に由来し、1986 年に Christian Batut, Henri Cohen, Michel Oliver によって PARI アセンブリカーネルと PARI Library の構想が始まった。1995 年に Karm Belabas が維持者となり、フランスの研究者を中心に開発・改良が行われ、



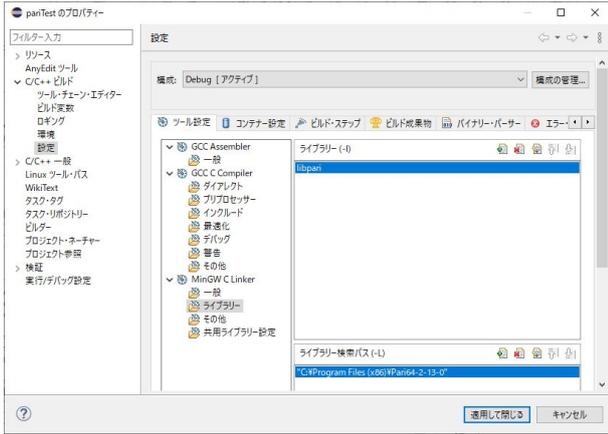


図 8: PARI/GP Download ページ

を使う場合、一定サイズのスタック領域が必要となる。そのために領域を確保しておく関数は、

```
void pari_init(size_t size, ulong maxprime)

```

であり、第一引数に必要なメモリ値を、第 2 引数には予め求めておく素数の上限を設定する。size は、アドレス bot から top (bot < top) までに入る long の個数、つまり

$$(top - bot) / \text{sizeof}(long)$$

を設定する。素数を使った計算や関数を活用するのでなければ第 2 引数の値は 0 でもよいが、第 1 引数の値は 500000 を下回らないように設定する (p. 13)。

PARI の関数を使うプログラムでは、引数と戻り値に C 言語による型と PARI の型 (タイプ) が混在するが、PARI 関数の整数値のほとんどが long 型となることに注意する。PARI の型宣言は一般的に GEN で、オブジェクトへのポインタとなり、表示や書き込みを行うときは書式指定子 %Ps を使う。

スタックとして確保されたメモリ領域には PARI オブジェクトが入るが、top の位置から下に押し込んでいく形になるで、底 bot まで達するとエラーとなる。したがって、不要なオブジェクトに対しては“garbage collection”によってそのスタック領域を解放しておく必要がある。現在のスタック位置に対するポインタ変数が avma (available memory address) で、一般にスタック位置に対するポインタ変数は pari\_sp (pari stack pointer) タイプなので下のようなコード書けば、あいだで生成された PARI オブジェクトは、これ以降で上書きされることになる (pp.16-17)。

```
pari_sp av = avma;
.....
avma = av;
```

この他にも、関数

GEN gerepile(pari\_sp ltop, pari\_sp lbot, GEN q) は lbot と ltop の間のオブジェクトを削除して avma の位置を上げ、PARI オブジェクト q のアップデートされたアドレスを返す (pp.18-19)。

## 4.2 基本的処理と線形方程式解法に使う関数

本研究で扱う具体的なプログラムは、与えられたバイナリコードから整数の剰余環  $\mathbb{Z}/8\mathbb{Z}$  上の Self-Dual Code を生成するものであり、主に使う PARI の関数は行列に関するものと連立一次方程式の解を全て求めるためのものである。ここでそれらを簡単に見ておく。

以下では PARI のオブジェクトとして gN で整数、gY で列ベクトル、gM で行列、gX で一般のオブジェクトを表すものとする。また、n は C 言語における long 型整数である。

整数の変換：

- stoi(n): 整数 n を PARI の整数に変換
- gmodulo(gX, gN): PARI オブジェクトの成分を gN を法した値に変換

行列関連関数：

- matsize(gM): 行列サイズ (行数→列数の順)
- mkvecs(n): n を要素とする 1 次元ベクトル
- vconcat(gV\_1, gV\_2): 列ベクトルの結合
- shallowmatconcat(gM\_1, gM\_2): 行列の行方向への結合 (“shallow” は簡易版)
- gcoeff(gV, n): ベクトルの第 n 成分 ( $1 \leq n$ )
- gcoeff(gM, n\_1, n\_2): 行列の (n\_1, n\_2) 成分
- det(gM): 行列式
- ginv(gM): 逆行列
- gtrans(gM): 転置行列

ベクトルや行列に関しては四則演算や成分に関する演算もいくつかあるが、PARI のオブジェクトとして扱わなくて済む部分はできるだけ C 言語の処理で行うこととした。

連立一次方程式の解：

- `gaussmodulo2(gM, gn, gY)`: 連立一次方程式に対応する式  $gM \times gX = gY \pmod{gN}$  を満たす列ベクトル  $gX$  を求める。解が存在する時、第1成分としてその代表解列ベクトル、第2成分として不定解列ベクトルからなる行列を返す。解法は剰余環上で行っているが、成分値は PARI の整数となるので  $gN$  の値を超えることがあり、必要に応じて `gmodulo` で変換する。
- `lift_shallow(gMm)`: `gMm` は `gmodulo` で変換された成分を持つ行列で、これにより成分値を PARI の整数に戻す。

### 4.3 繰り返し処理

PARI には合成数を動く `forcomposite`, 素数の動く `forprime`, ベクトルを動く `forvec`, 置換を動く `forperm` などの繰り返し処理がある (pp. 42–44)。ここでは、前述の連立一次方程式の基本解に不定解の組み合わせを足し合わせたものすべてを求めたいので、以下のような1から  $n$  までの集合の部分集合すべてを動く `forsubset` で処理を使う。“garbage collection”も含めた、一般的な処理の流れ次のようになる (p.44)。

```
forsubset_t T;
GEN sub;
pari_sp av = avma, av2;
forallsubset_init(&T, n);
av2 = avma;
while((sub=forsubset_next(&T)){
    .....
    avma = av2;
})
avma = av;
```

## 5 誤り訂正符号と剰余環上の Self-Dual Code

まず誤り訂正符号の簡単な説明と、その例を示す。続いて、整数の剰余環  $\mathbb{Z}/8\mathbb{Z}$  上の Self-Dual Code を生成するための拡張アルゴリズムを示す。

### 5.1 誤り訂正符号

デジタル情報処理における基本的な単位は0と1で表現されるビット情報であり、それらをいくつか並べたものを符号語 (code word), いくつかの符号語の集まりを符号 (code) と呼ぶ。

例えば、次のビット行列の4つの行ベクトルが2進数体  $\mathbb{F}_2$  上で生成する  $2^4 = 16$  個のベクトル全体を一つの符号  $C_0$  と考える。

$$C_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \quad (1)$$

この符号では、16個の内からどの2つの符号語をとっても4カ所以上異なり、Hamming 重みによる最小距離

$$\min\{d_H(\vec{c}, \vec{c}'); c, c' \in C_0\} = \min\{wt(\vec{c}); c \in C_0\} = 4$$

の線形符号となっているので、 $[8, 4, 4]$  型の線形バイナリコードと呼ばれる。

### 5.2 剰余環 $\mathbb{Z}/8\mathbb{Z}$ 上の Self-Dual Code

1994年に Hammons 等は、以前から知られていた非線形符号である Kerdock 符号, Preparata 符号, Goethals 符号などが、整数の剰余環  $\mathbb{Z}/4\mathbb{Z}$  上の Self-Dual Code と呼ばれる符号と対応していることを示し ([9]), それ以降剰余環や有限体の拡大環上の誤り訂正符号の研究が盛んに行われた。我々は2009年に任意の素数  $p$  に対しその3乗による整数の剰余環  $\mathbb{Z}/p^3\mathbb{Z}$  上の Self-Dual Code の構成法を示すことで、符号全体の数を求める Mass Formula と呼ばれる式を与えた ([17])。

Self-Dual Code は自分自身がその直交補空間と一致する符号のことであり、この時はベクトル空間としての次元は  $n/2$  となる。また任意の符号語  $\vec{c}$  に対し1の個数が4で割り切れるとき、その符号を doubly even Self-Dual Code と呼ぶ。我々のアルゴリズムは、2進数体上の doubly even Self-Dual Code を与え、 $\mathbb{Z}/8\mathbb{Z}$  上の Self-Dual Code を構成するものであり、また任意の Self-Dual Code がこの方法で構成されることを示した。

ここでは、与えられた  $[n, n/2, d]$  型バイナリ Self-Dual Code  $C_0$  から、 $\mathbb{Z}/8\mathbb{Z}$  上の Self-Dual Code を生成するための具体的な構成法を示す。まず  $C_0$  の  $n/2$  個の生成ベクトルを doubly even なもの  $k$  個とそれ以外の  $l$  個 ( $n = k+l$ ) のパートに分割し、次のように表す。

$$\begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} I_k & A_2 & A_{30} & A_{40} \\ 0 & I_l & B_3 & B_{40} \end{bmatrix}$$

ただし、 $I_k, I_l$  はそれぞれ  $k$  次,  $l$  次の単位行列であり、 $A_{40}$  は適当な列変換で逆行列を持つようにする。これから、次のような  $\mathbb{Z}/8\mathbb{Z}$  上の  $[n, k+2l]$  型 Self-Dual Code

$C$  を生成する.

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} I_k & A_2 & A_{30} + 2A_{31} & A_{40} + 2A_{41} + 4A_{42} \\ 0 & 2I_l & 2B_3 & 2B_{40} + 4B_{41} \\ 0 & 0 & 4I_l & 4C_4 \end{bmatrix}$$

ここで、バイナリ行列として求めなければならないのは、 $A_{31}$ ,  $A_{41}$ ,  $A_{42}$ ,  $C_4$ , および  $B_{41}$  であるが、最後の2つは (mod 2) における次式から一意的に求まる.

$$\begin{cases} C_4 & \equiv & (-A_{40}^{-1} A_{30})^t \\ B_{41} & \equiv & -A_{40}^{-1}(D + A_{31}B_3^t + A_{41}B_{40}^t) \\ & \text{ただし} & D = \frac{1}{2}(A_2 + A_{30}B_3^t + A_{40}B_{40}^t) \end{cases}$$

$A_{31}$  と  $A_{41}$  は,  $(f_{ij}) = \frac{1}{2}(I_k + A_2A_2^t + A_{30}A_{30}^t + A_{40}A_{40}^t)$  に対し, 次の式を満たすバイナリ行列である.

$$(f_{ij}) + \widetilde{A_{30}A_{31}^t} + \widetilde{A_{40}A_{41}^t} \equiv 0.$$

ただし, ここで  $\widetilde{X} = X + X^t$  である.

[ステップ 1]

$A_{30} = (\vec{a}_i^3)$ ,  $A_{31} = (\vec{x}_i)$ ,  $A_{40} = (\vec{a}_i^4)$ , および  $A_{41} = (\vec{y}_i)$  としたとき, ベクトルの内積  $\vec{x} \cdot \vec{y}$  を使って次の  $(l+k)k = \frac{1}{2}nk$  個の変数を持つ  $\frac{1}{2}k(k-1) + k = \frac{1}{2}(k+1)k$  個の制御条件を満たす連立一次方程式を解くことになる.

$$\begin{cases} (\vec{a}_i^3 + \vec{1}_l) \cdot \vec{x}_i + (\vec{a}_i^4 + \vec{1}_k) \cdot \vec{y}_i & \equiv \frac{1}{2}f_{ii} \\ & (i = 1, \dots, k), \\ \vec{a}_i^3 \cdot \vec{x}_j + \vec{x}_i \cdot \vec{a}_j^3 + \vec{a}_i^4 \cdot \vec{y}_j + \vec{y}_i \cdot \vec{a}_j^4 & \equiv f_{ij} \\ & (1 \leq i < j \leq k). \end{cases} \quad (2)$$

[ステップ 2]

$(h_{ij}) = \frac{1}{2}((f_{ij}) + \widetilde{A_{30}A_{31}^t} + \widetilde{A_{40}A_{41}^t})$  に対し,  $A_{42} = (z_{ij})$  と置いたときに  $k^2$  個の変数を持つ, 次のような  $\frac{1}{2}k(k-1)$  個の制御条件を満たす連立一次方程式を解くことである.

$$\vec{a}_i^4 \cdot \vec{z}_j + \vec{z}_i \cdot \vec{a}_j^4 \equiv h_{ij} + \vec{x}_i \cdot \vec{x}_j + \vec{y}_i \cdot \vec{y}_j \quad (3) \quad (1 \leq i < j \leq k).$$

上記の方法によって,  $C_0$  から派生するの  $\mathbb{Z}/8\mathbb{Z}$  上の Self-Dual Code がすべて求まることになるが, ここで最小距離  $d$  を大きくすることを考えたい. 2進数体上では, 符号語の重さは1の個数であったが, ここでは  $\mathbb{Z}/8\mathbb{Z} = \{0, 1, 2, 3, 4, 5, 6, 7\}$  の各要素の重みを定義する必要がある. その為に各要素に3ビットの Gray Code

$$\{(000), (001), (011), (010), (110), (111), (101), (100)\},$$

を対応させることで, 重さを与える ([18]). つまり, 偶数 2, 4, 6 は重さ“2”, 1, 3, 7 は重さ“1”, 5 は重さ“3”である.

ここでは, 基底となる符号ベクトルの重さをできるだけ大きくするように符号を構成したい. そのためには  $A_{40}$  の  $(i, j)$  成分が1ならば,  $A_{41}$ ,  $A_{42}$  の  $(i, j)$  成分をそれぞれ0, 1となるようにすればよい.  $A_{41}$  を求めるための連立一次方程式 (2) は, 変数が  $\frac{1}{2}nk$  個あり, 方程式の数  $\frac{1}{2}k(k+1)$  との差  $\frac{1}{2}(n-k-1)k$  が自由度となるので, それを使って0の位置を増やすことを考える.

$A_{42} = (z_{ij})$  においては連立一次方程式 (3) に  $\frac{1}{2}k(k+1)k$  の自由度があるので, 対応する部分が1となるように連立一次方程式を設定すればよい.

$A_{30}$  における  $(i, j)$  成分が0ならば  $A_{31}$  の対応する成分を1としたいが, その分制約式が増えてしまい, 場合によると連立一次方程式 (2) が解を持たなくなるので, 今回はその部分の条件は設定しないこととした.

## 6 PARI Library をインクルードした C 言語によるプログラム

プログラム作成の基本的なコンセプトはできる限り C 言語による処理を行い, 特に main 内では PARI Library の使用を最小限にとどめることである. そのため, 行列の基本的な演算用に関数を作成し, PARI の行列やベクトルとの変換も関数を呼び出して行う. 最初に与える行列  $C_0$  は (1) で固定し, 分割を main 内で与えることとした.

### 6.1 PARI Library を使ういくつかの関数

C 言語による行列を PARI 行列へ変換する関数:

```
GEN toPariMat(int r_size,int c_size,+
              int m[][c_size]){
    GEN gm;
    gm=gtovec(stoi(m[0][0]));
    int i,j;
    for(i=1;i<c_size;i++){
        gm=shallowconcat(gm,mkvecs(m[0][i]));
    }
    gm=gtomat(gm);
    for(i=1;i<r_size;i++){
        GEN mm1;
        mm1=gtovec(stoi(m[i][0]));
        for(j=1;j<c_size;j++){
            mm1=shallowconcat(mm1,
                               mkvecs(m[i][j]));
        }
        gm=vconcat(gm,gtomat(mm1));
    }
    return gm;
}
```

PARI 行列を C 言語による行列変換する関数:

```

void toIntMat(int rs,int cs,GEN m,+
              int im[rs][cs]){
    int i,j;
    for(i=0;i<rs;i++){
        for(j=0;j<cs;j++){
            im[i][j]=(int)itos(gcoeff(m,i+1,j+1));
        }
    }
}

```

特定解と不定解集合を与えてすべての解を求める関数：

```

int genAllas(int countF,int cs,int ix[],+
             int imF[][cs],int as[][cs]){
    int i;
    pari_sp av=avma,av2;
    forsubset_t T;
    GEN subsmallM;
    forallsubset_init(&T,(long)countF);
    av2=avma;
    int count=0;
    while((subsmallM=forsubset_next(&T)){
        GEN subM=gtovec(subsmallM);
        int l=(int)itos(matsize(subM)[2]);
        int subS[l];
        toIntVec(l,subM,subS);
        for(i=0;i<l;i++){
            int si=subS[i]-1;
            ppVec(cs,imF[si],as[count]);
        }
        ppVec(cs,ix,as[count]);
        count++;
        avma=av2;
    }
    avma=av;
    return count;
}

```

連立一次方程式の解を求める関数：

```

int solveEquation(int rs,int cs,int iy[rs],+
                  int im[rs][cs],int ix[cs],int imF[][cs]){
    pari_sp av=avma,av2;
    GEN x,y,x1,x2;
    GEN m;
    av2=avma;
    y=lift_shallow(gmodulo(toPariVec(rs,iy),+
                           stoi(2)));
    y=gtrans(y);
    m=lift_shallow(gmodulo(toPariMat(rs,cs,+
                                     im),stoi(2)));
    x=gaussmodulo2(m,stoi(iMod),y);
    x1=lift_shallow(gmodulo(x[1],stoi(2)));
    toIntVec(cs,x1,ix);
    x2=lift_shallow(gmodulo(x[2],stoi(2)));
    int matRS=(int)itos(matsize(x2)[2]);
}

```

```

int i,j;
int cF=0;
int ix1[cs];
int imm[(K+L)*K][cs];
toIntVec(cs,gtrans(x1),ix1);
toIntMat(matRS,cs,gtrans(x2),imm);
int im0[cs];
for(i=0;i<cs;i++){
    im0[i]=0;
}
avma=av2;
avma=av;
if(matRS>0){
    for(i=0;i<matRS;i++){
        if(isSameVec(cs,im0,imm[i])!=0){
            for(j=0;j<cs;j++){
                imF[cF][j]=imm[i][j];
            }
            cF++;
        }
    }
    return cF;
}else if(matRS==0){
    for(i=0;i<cs;i++){
        imF[0][i]=0;
    }
    return 0;
}else{
    return 0;
}
}

```

## 6.2 インクルード、定義、および main プログラム

インクルードするヘッダファイルと、定数の定義は以下のようにして、定義の K と L は分割の仕方によって変えることになる。

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <pari.h>

#define N 8
#define iMod 8
#define maxW 3

// #define K 4
// #define L 0

// #define K 3
// #define L 1

// #define K 2
// #define L 2

```

```
#define K 1
#define L 3
```

mainにおいて、分割に合わせて行列を定義しており、ここでは $K=1$ ,  $L=3$ のものが使われる設定となっている。生成されたコードはその Weight Distribution 多項式を求めて、ファイルに書き込むこととした。なお、以下のプログラムにおいて“;”や“(”の後に“+”がある場合は、次行に続くことを示している。

```
int main(void) {
    pari_init(1000000,1000);
    int wt[(iMod)] = {0,1,2,1,2,3,2,1};
    int iK2 = K*K;
    int iL2 = L*L;
    int iKL = K*L;
    int iKpL = K+L;
    int iKpLpL = K+2*L;
    int iKL2 = iL2+iK2;
    /* int iA20;
    int iA30;
    int iA40[K][K]={0,1,1,1},{1,0,1,1},+
        {1,1,0,1},{1,1,1,0}};
    int iB30;
    int iB40;
    int iA20[K][L]={0},{0},{0}};
    int iA30[K][L]={0},{1},{1}};
    int iA40[K][K]={1,1,1},{0,1,1},{1,0,1}};
    int iB30[L][L]={1}};
    int iB40[L][K]={1,1,0}};
    int iA20[K][L]={0,0},{0,0}};
    int iA30[K][L]={1,1},{1,1}};
    int iA40[K][K]={0,1},{1,0}};
    int iB30[L][L]={0,1},{1,0}};
    int iB40[L][K]={1,1},{1,1}};
    */
    // 分割として使われる値は以下の5行
    int iA20[K][L]={0,0,0}};
    int iA30[K][L]={0,1,1}};
    int iA40[K][K]={1}};
    int iB30[L][L]={1,0,1},{1,1,0},{1,1,1}};
    int iB40[L][K]={1},{1},{0}};

    int iC41[L][K];
    int iCT41[K][L];
    int i,j;
    FILE *fp;// 結果書き込みファイル
    int iC0[iKpL][N];
    int invA40[K][K];
    // A40の逆行列計算(ここだけPARI関数使用)
    GEN m = lift_shallow(gmodulo(+
        toPariMat(K,K,iA40),stoi(2)));
    GEN invm = lift_shallow(+
        ginv(gmodulo(m,stoi(2))));
    if((int)itos(det(m))!=0){
        printf("\n determinate is not zero\n");
```

```
        toIntMat(K,K,invm,invA40);
    }else{
        printf("\n determinate of A40 is zero\n");
    }
    // Lが正のときにC4を計算
    if(L>0){
        mulMatrix(K,K,L,invA40,iA30,iCT41);
        transMatrix(K,L,iCT41,iC41);
        iMmod2(L,K,iC41);
        genC0(iKpL,iKpLpL,iA20,iA30,+
            iA40,iB30,iB40,iC0);
    }else if(L==0){
        genC00(iA40,iC0);
    }

    iMmod2(iKpL,iKpLpL,iC0);
    int zeroN = 0;
    int iY1[zeroN+K*(K+1)/2];
    int iM1[zeroN+K*(K+1)/2][(L+K)*K];
    // ステップ1の右辺列ベクトル生成
    genY1(zeroN,iC0,iY1);
    iVmod2(zeroN+K*(K+1)/2,iY1);
    // ステップ1の制約行列生成
    if(L>0){
        genM1(zeroN,iA30,iA40,iM1);
    }else if(L==0){
        genM10(iA40,iM1);
    }

    iMmod2(zeroN+K*(K+1)/2,iKpL*K,iM1);
    fp = fopen("y1_a1s.txt","a");

    int ttE;
    int imF[iKpL*K][iKpL*K];
    int countF;
    int ix[iKpL*K];
    // ステップ1の式(2)の特定解(ix)と不定解(imF)
    countF = solveEquation(zeroN+K*(K+1)/2,+
        iKpL*K,iY1,iM1,ix,imF);
    fprintfVec(fp,"ix",iKpL*K,ix);
    fclose(fp);
    ttE = twoPowers(countF);

    int iA1s[ttE][iK2+iKL];
    for(i=0;i<ttE;i++){
        for(j=0;j<iK2+iKL;j++){
            iA1s[i][j] = 0;
        }
    }
    // ステップ1における解ベクトル全体の生成
    int count = genAllas(countF,iK2+iKL,ix,+
        imF,iA1s);
    for(i=0;i<count;i++){
        iVmod2(iK2+iKL,iA1s[i]);
    }
}
```

```

int iA31s[count][iKL];
int iA41s[count][iK2];
if(L>0){
    devideAs(count,iKpL,iKL,iA1s,iA31s,+
            iA41s);
}else if(L==0){
    for(i=0;i<count;i++){
        for(j=0;j<K*K;j++){
            iA41s[i][j]=iA1s[i][j];
        }
    }
}
fp=fopen("y1_a1s.txt","a");
fprintf(fp,"countF=%d, count=+
        %d \n",countF,count);
fprintfMat(fp,"imFree",countF,iKpL*K,imF);
for(i=0;i<count;i++){
    if(L>0){
        fprintf(fp,"iA31s[%d]=",i);
        fprintfVec(fp,"",iKL,iA31s[i]);
    }
    fprintf(fp,"iA41s[%d]=",i);
    fprintfVec(fp,"",iK2,iA41s[i]);
}
fclose(fp);

int number;
int iA31[K][L];
int iA41[K][K];
int iB41[L][K];
int imF2[iK2][iK2];
int wtdist[(maxW)*N];
for(i=0;i<iK2;i++){
    for(j=0;j<iK2;j++){
        imF2[i][j]=0;
    }
}
int count2;
int ix2[iK2];
int c[iKpLpL][N];

for(number=0;number<count;number++){
    int istr = 100*K+number;
    char str2[10];
    char str3[]=".txt";
    itoa(istr,str2,10);
    strcat(str2,str3);
    fp = fopen(str2,"w");
    fprintf(fp,"test number %d \n",number);
// A31, A41 を固定
vecToMat(K,L,iA31s[number],iA31);
vecToMat(K,K,iA41s[number],iA41);
fprintfVec(fp,"iA31s",K*L,iA31s[number]);
fprintfMat(fp,"iA31",K,L,iA31);
fprintfMat(fp,"iA41",K,K,iA41);

    fprintfMat(fp,"iC41",L,K,iC41);
    fclose(fp);
// ステップ 2 における結果書出し用ファイル
fp = fopen(str2,"a");
zeroN = 0;
for(i=0;i<K;i++){
    for(j=0;j<K;j++){
        if(i!=j){
            if(iA41[i][j]==0){
                zeroN++;
            }
        }
    }
}
// A31, A41 から B41 を計算
genB41(iKpL,iKpLpL,invA40,iC0,iA31,+
        iA41,iB41);
iMmod2(K,L,iB41);
fprintf(fp,"zeroNumber is %d \n",+
        zeroN);

    int iY2[zeroN+K*(K-1)/2];
    int iM2[zeroN+K*(K-1)/2][iK2];
// ステップ 2 の右辺列ベクトル生成
genY2(iKpL,iKpLpL,zeroN,iC0,iA31,+
        iA41,iY2);
// ステップ 2 の制約行列生成
genM2(zeroN,iA40,iA41,iM2);
iVmod2(zeroN+K*(K-1)/2,iY2);
iMmod2(zeroN+K*(K-1)/2,iK2,iM2);
fprintfVec(fp,"Y2",zeroN+K*(K-1)/2,+
        iY2);
fprintfMat(fp,"M2",+
        zeroN+K*(K-1)/2, iK2,iM2);
// ステップ 2 における解ベクトル全体の生成
countF=solveEquation(+
        zeroN+K*(K-1)/2,iK2,+
        iY2,iM2,ix2,imF2);
fprintf(fp,"countF=%d \n",countF);
fprintfVec(fp,"x2",iK2,ix2);
fprintfMat(fp,"mF2",countF,iK2,imF2);

    ttE = twoPowers(countF);
    int a42s[ttE][iK2];
    for(i=0;i<ttE;i++){
        for(j=0;j<iK2;j++){
            a42s[i][j] = 0;
        }
    }
// ステップ 2 における解ベクトル全体の生成
count2 = genAllas(countF,iK2,ix2,+
        imF2,a42s);
fprintf(fp,"count2=%d \n",count2);
for(i=0;i<count2;i++){
    iVmod2(iK2,a42s[i]);
    fprintf(fp,"a2s[%d]=",i);
}

```

```

    fprintfVec(fp, "", iK2, a42s[i]);
}

for(i=0;i<count2;i++){
// 各 A42 に対し符号生成
    genC(iKpL, iKpLpL, iC0, iA31, iA41, +
        iB41, iC41, a42s[i], c);
    iMmod8(iKpLpL, N, c);
    fprintf(fp, "%d:", i);
    fprintfMat(fp, "c", iKpLpL, N, c);
    for(j=0; j<iKpLpL; j++){
        fprintf(fp, "wt[c%d]=%d, ", +
            j+1, wtOfVec(wt, N, c[j]));
    }
    fprintf(fp, "\n");
    wtDist(wt, iKpLpL, N, c, wtdist);

    int minWV = calMinWeight((maxW)*N, +
        wtdist);
    int maxWV = calMaxWeight((maxW)*N, +
        wtdist);
// 生成された符号語の最小, 最大重み
    fprintf(fp, "Minimum Weight=%d, +
        Maximum Weight=%d, +
        \n Weight Distribution:\n", +
        minWV, maxWV);
// Weight Distribution の書込み
    for(j=0; j<maxW*N; j++){
        if(wtdist[j]!=0){
            fprintf(fp, "%d*X^%d", +
                wtdist[j], j);
        }
    }
    fprintf(fp, "\n");
}
fclose(fp);
}
int end;
scanf("%d", &end);
pari_close();
return EXIT_SUCCESS;
}

```

## 7 結果と今後の課題

[8, 4, 4] 型バイナリ線形符号 (1) に対し, 4 通りの分割  $(k, l) = (4, 0), (3, 1), (2, 2), (1, 3)$  を行い, それぞれに対して  $\mathbb{Z}/8\mathbb{Z}$  上の Self-Dual Code で符号語生成ベクトルの重みができるだけ大きくなるようなものを計算した. ただし,  $(k, l) = (2, 2)$  の場合は,  $A_{40}$  を正則とするため, 7, 8 列と 5, 6 列の入れ替えを行った.

$(k, l) = (4, 0)$  の場合は, ステップ 1 で 64 個の  $A_{41}$  を生成し, それぞれに対して  $2 \sim 128 = 2^7$  個の  $\mathbb{Z}/8\mathbb{Z}$  上

Self-Dual Code を得た. また, すべてで, 最小重みは 6 となっていた.

$(k, l) = (3, 1)$  の場合は, ステップ 1 で 64 個の  $A_{31}$  と  $A_{41}$  の組を生成し, それぞれに対して  $2 \sim 32 = 2^5$  個の  $\mathbb{Z}/8\mathbb{Z}$  上 Self-Dual Code を得た. また, すべてで最小重みは 2, 次が 6 となっていた.

$(k, l) = (2, 2)$  の場合は, ステップ 1 で 32 個の  $A_{31}$  と  $A_{41}$  の組を生成し, それぞれに対して  $2 \sim 8 = 2^3$  個の  $\mathbb{Z}/8\mathbb{Z}$  上 Self-Dual Code を得た. また, すべてで最小重みは 2, 次が 4 となっていた.

$(k, l) = (1, 3)$  の場合は, ステップ 1 で 8 個の  $A_{31}$  と  $A_{41}$  の組を生成し, それぞれに対して 1 か 2 個の  $\mathbb{Z}/8\mathbb{Z}$  上 Self-Dual Code を得た. また, すべてで最小重みは 2, 次が 4 となっていた.

$(k, l) = (4, 0)$  の場合が一番時間がかかっているが数分程度であり, それほどストレスを感じることもなく終了する. 扱った連立一次方程式に対応する行列は, 行数 10 前後, 列数 16 であり, この程度ならば PARI の関数は問題なくすべての解を出力してくれることがわかった.

今回は, 特定の [8, 4, 4] 型バイナリ線形符号から出発したものであったが, 他の符号に対しても適応できることを確認する必要がある.

## 参考文献

- [1] Behr, Bernhard, Seminar Report Plankalkül, 31 July 2015, [c1-informatik.uibk.ac.at/teaching/ss15/bob/reports/ss15-BB.pdf](http://c1-informatik.uibk.ac.at/teaching/ss15/bob/reports/ss15-BB.pdf) (2021/03/12 参照)
- [2] Dahl, O. -J., Dijkstra, E. W., and Hoare, C. A. R., *Structured Programming*, A. P. I. C. Studies in Data Processing, No. 8, ACADEMIC PRESS Inc. (London), (1972)
- [3] Dahl, O. -J., The Birth of Object Orientation: The Simula Languages, *Object-Oriented to Formal Methods*, Lecture Notes in Computer Science, 2635, (2004), pp. 15–25.
- [4] Dijkstra, E. W., Structured programming, *Report on a conference sponsored by NATO SCIENCE COMMITTEE*, (1969), pp. 65–68
- [5] Dijkstra, E. W., EWD 498: How do we tell truths that might hurt?, *Selected Writings on Computing: A Personal Perspective*, Springer-Verlag, (1982), pp. 129–131
- [6] Fuegi, J. and Francis, J., Lovelace & Babbage and the Creation of the 1843 ‘Notes’, *IEEE Annals of the History of Computing*, Vol. 25, Issue 4, (2003) pp. 16–26. <https://pdfs.semanticscholar.org/81bb/\f32d2642a7a8c6b0a867379a4e9e99d872bc.pdf> (2021/03/12 参照)
- [7] Giloi, Wolfgang, K., Konrad Zuse’s Plankalkül: The First High-Level, "non von Neumann" Programming

- Language, *IEEE Annals of the History of Computing*, Vol. 19, (1997), pp. 17–24.
- [8] Goldstine, Herman H., *The Computer: from Pascal to von Neumann*, Princeton University Press. ISBN 0-691-02367-0, (1972)
- [9] Hammons, A.R., Jr., Kumar, P.V., Calderbank, A.R., Sloane, N.J.A. and Solé, P., The  $\mathbb{Z}/4\mathbb{Z}$  linearity of Kerdock, Preparata, Goethals and related codes, *IEEE Trans. Inform. Theory* 40 (1994) pp. 301–319.
- [10] Hearn, A. C., REDUCE: The First Forty Years, *Invited paper presented at the A3L Conference in Honor of the 60th Birthday of Volker Weispfenning*, (2005). <http://www.reduce-algebra.com/reduce40.pdf>, (2021/03/12 参照)
- [11] Iverson, K. E., *A Programming Language*, JOHN WILEY AND SONS, INC, (1967).
- [12] Kernighan, B. W and Ritchie, D. M., *The C Programming Language (2nd ed.)*, Englewood Cliffs, NJ: Prentice Hall. ISBN 0-13-110362-8., (1988)
- [13] Kilburn, T., Tootill, G. C. and Williams, F. C., Universal High-Speed Digital Computers: A Small-Scale Experimental Machine. *J.I.E.E.* 98 Part II, No. 61 (1951). <http://curation.cs.manchester.ac.uk/computer50/www.computer50.org/kgill/mark1/ssem.html>, (2021/03/12 参照)
- [14] Knuth, D. E., Structured Programming with go to Statements, *Computing Surveys*, Vol. 6, No. 4, (1974), 261–301.
- [15] Knuth, D. E., and Pardo, L. T., The Early Development of Programming Languages, *A History of Computing in the Twentieth Century*, Academic Press, Elsevier, (1980), pp. 197–273
- [16] Linger, R. C., Mills, H. D., and Witt, B. I., *Structured Programming: Theory and Practice*, Addison-Wesley Publication, (1979)
- [17] Nagata, K., Nemenzo, F., and Wada, H., The number of self-dual codes over  $\mathbb{Z}_{p^3}$ , *Designs, Codes and Cryptography* 50 (2009) pp. 291–303.
- [18] Nagata, K. and Nemenzo, F., Some Properties of Binary Gray Code, *Proceedings of International Conference on Computer Application Technologies*, Aug. 31-Sep. 2 (2015) pp. 72–75.
- [19] Richards, M., The portability of the BCPL compiler, *Software: Practice and Experience*, Vol. 1, Issue 2, <https://doi.org/10.1002/spe.4380010204>(1971), pp.135–146.
- [20] Richards, M., EDSAC Initial Orders and Squares Program, *University of Cambridge Computer Laboratory*, <https://www.cl.cam.ac.uk/~mr10/Edsac/edsacposter.pdf>, (2021/03/12 参照)
- [21] Ritchie, D. M., The Development of the C Language, *Bell Labs/Lucent Technologies*, Murray Hill, NJ 07974 USA, <https://www.bell-labs.com/usr/dmr/www/chist.pdf>, (2021/03/12 参照)
- [22] Stern, Nancy, *From ENIAC to UNIVAC : an appraisal of the Eckert-Mauchly computers*, Digital Press history of computing series, Digital Press, 1981
- [23] Turing, A. M., On Computable Numbers, with an Application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, 2 42: pp. 230–265, (1936) [https://www.cs.virginia.edu/~robins/Turing\\_Paper\\_1936.pdf](https://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf), (2021/03/12 参照)
- [24] Zuse, Konrad, Über den allgemeinen Plankalkül als Mittel zur Formulierung schematisch-kombinativer Aufgaben, *Arch. Math.* 1, (1948/49), pp. 441–449.
- [25] 永田清, PARI/GP による Self-Dual Code の生成, 第 5 回国際 ICT 利用研究会全国大会 *IIARS2020* 講演予稿集, 2020 年 12 月 6 日, pp. 40-43

# 連分数公式を予測するラマヌジャン・マシンの紹介

鈴木 治郎 \*1

\*1 信州大学 全学教育機構

\*1szkjiro@shinshu-u.ac.jp

キーワード 機械学習, 連分数, ラマヌジャン

## 1 はじめに

数学領域において、情報処理技術による自動で未知の公式の作成を試みることはこれまでも試み続けられており、よく知られたものには、たとえば円周率計算を16進数で任意の桁から計算できるBBP公式がある[2].

$$\pi = \sum_{k=0}^{\infty} \frac{1}{16^k} \left( \frac{4}{8k+1} - \frac{2}{8k+4} - \frac{1}{8k+5} - \frac{1}{8k+6} \right)$$

2020年にRaayoniらが発表したRamanujan Machine[1]は、深層学習技術を用いて、数学定数の計算式を与えるものであり、たとえば以下の例は既知の計算式を再発見したものである。

$$\frac{e}{e-2} = 4 - \frac{1}{5 - \frac{2}{6 - \frac{3}{7 - \frac{4}{8 - \dots}}}}$$

以下の例はまだ証明は得られていないものである。

$$\frac{8}{7\zeta(3)} = 1 \cdot 1 - \frac{1^6}{3 \cdot 7 - \frac{2^6}{5 \cdot 19 - \frac{3^6}{7 \cdot 37 - \dots}}}$$

ただし $\zeta$ はゼータ関数を表す。おおまかに言えば、これらの公式もBBP公式の発見と同様、左辺と右辺の近似値をうまくマッチングさせる点では変わらない。以下ではこの公式生成技術の概要を紹介する。

## 2 連分数展開

たとえば円周率 $\pi$ は

$$\pi = 3 + \frac{1}{7 + \frac{1}{15 + \dots}}$$

という連分数展開をもち、途中までの展開を分数に直せば

$$3, \frac{22}{7}, \frac{355}{113}, \dots$$

を得る。連分数展開は、このような実数の近似値を与える方法としても有効である。

連分数を一般に

$$a_0 + \frac{a_1}{b_1 + \frac{a_2}{b_2} + \dots}$$

と表すとき、 $b_n = 1$ とする連分数を正則連分数という。正則連分数を求めるアルゴリズムは

- (1) 整数部  $a_n$  を求める
- (2) 小数部の逆数を求めて、(1)から繰り返す

である。

正則連分数に関しては

- 有理数ならば展開は有限である
- 二次無理数ならば展開は周期的である

などがわかる。正則連分数でなければ、円周率においても興味深い関係式がある。たとえば次はBrounckerが与えたものであり、超幾何級数により証明された。

$$\frac{\pi}{4} = 1 + \frac{1^2}{2 + \frac{3^2}{2 + \frac{5^2}{2 + \dots}}}$$

### 2.1 連分数展開と漸化式

連分数展開を  $n$  番目までで打ち切って得られる(近似)分数を

$$\frac{p_n}{q_n}$$

で表すとき,

$$p_0 = a_0, p_{-1} = 1, q_0 = 1, q_{-1} = 0$$

とにおいて, 次の漸化式で計算できる.

$$\begin{cases} p_{n+1} = a_n p_n + b_n p_{n-1} \\ q_{n+1} = a_n q_n + b_n q_{n-1} \end{cases}$$

この漸化式は, 行列の形で表せば次の通りである.

$$\begin{pmatrix} p_{n+1} & q_{n+1} \\ p_n & q_n \end{pmatrix} = \begin{pmatrix} a_n & b_n \\ 1 & 0 \end{pmatrix} \begin{pmatrix} p_n & q_n \\ p_{n-1} & q_{n-1} \end{pmatrix}$$

### 3 ラマヌジャン・マシンの仕組み

#### 3.1 中間一致手続き

数学的定数などを表す  $c$  に対する有理式 (話を簡単にするために分母・分子とも一次式としたが, 論文では一般の多項式を扱っている)

$$\frac{\alpha c + \beta}{\gamma c + \delta}, \quad \alpha, \beta, \gamma, \delta \in \mathbb{Z} \quad (\text{A})$$

のリストおよび適当な  $\{a_n\}, \{b_n\}$  に対する近似分数

$$\frac{p_n}{q_n} \quad (\text{B}(n))$$

を考える. このとき, 式 (A) と (B(n)) の値の近い組を見つけ出し, さらに計算精度を上げることで計算公式

$$\frac{\alpha c + \beta}{\gamma c + \delta} = \lim_{n \rightarrow \infty} \frac{p_n}{q_n}$$

が実際に成り立つ可能性を高くする (中間一致 (MTTF; Meet-To-The-Middle) 手続き).

先の Brouncker の公式の場合

$$a_n = n^2, b_n = 2$$

である.

#### 3.2 勾配降下手続き

式 (B(n)) の候補を効率よく生成するために機械学習で使われる勾配降下法 (GD; Gradient Descent) を適用する.

### 4 ラマヌジャン・マシンの現在

ラマヌジャン・マシンは 2019 年にオープンソースプロジェクトとして稼働した.

www.RamanujanMachine.com

このサーバ上では

- メルセンヌ素数探索プロジェクト GIMPS と同様の計算時間提供ユーザの管理
- 発見された計算公式候補を公開し, 証明を募集
- アルゴリズムの改善意見の募集

を行っており, 数学的成果は arXiv を通じて継続的に更新されている.

### 参考文献

- [1] Gal Raayoni et al., The Ramanujan Machine: Automatically Generated Conjectures on Fundamental Constants, <https://arxiv.org/abs/1907.00205>, 2020.
- [2] D.H.Bailey, P.B.Borwein, S.Plouffe, On the Rapid Computation of Various Polylogarithmic Constants, Math. Comp. 66, no.218, 903–913, 1997.