

実験数学を AI とやってみる

国立情報学研究所 横山 重俊

Shigetoshi Yokoyama, National Institute of Informatics

1 はじめに

近年、数学の研究・教育の双方において実験数学の重要性が増している。とりわけ、SageMath, Julia/Nemo や LMFDB, Jupyter Notebook といったオープンな数学ソフトウェア群, データベース群, 対話メディアに, 機械学習, 大規模言語モデル (LLM) を組み合わせることで, 数式処理, 数値計算と自然言語理解を統合した新しい研究・教育での実験数学スタイルが実現しつつある。

本稿では数論を題材に, 数体に関する実験数学を AI との協調的な対話を通じて構築・検証する手法を紹介する。特に, 研究者で構成される共同研究体制の中に AI をメンバーとして加える方法について紹介したい。さらに, 教育応用への展望などを含めて「AI と共に数学を探究する」実践の可能性を論じる。

2 背景

オープンサイエンスの推進という枠組みの中で, 論文の再現性確保のために論文のみならず, 研究データや実験用プログラムコードのオープン化への取り組みが進んでいる [1]。これに加えて AI 技術の進展により AI for Science というキーワードのもとに様々な研究分野での AI の活用が進んでいる。研究支援基盤側でも研究データ基盤と AI 基盤の連携が注目されている [2]。

数学の分野においても, arXiv の活用やそこからデータやプログラムコードへのリンクなどを活用した研究再現性を向上させるための取り組みが進んでいる。さらには数学ソフトウェアの活用や LMFDB [3] のような研究データの分野別統合データベース構築も進み, それらを活用した研究も発現している [4, 5]。

オープンデータということで論文に付随するデータの公開というにとどまらず, 今後ますます分野別のデータベースの充実が進み, その豊富なデータを不可欠な要素とする研究が増えると予想できる。このような研究のためのデータベースから論文が生まれて育つ, データの土壌化が益々進むものと考えられる。この動きと AI for Science の動きの融合により数学研究や数学教育における数学ソフトウェアと AI との連携の重要性がますます高まるだろう。

本稿では, 数論研究における実験数学の一例を通してこの動きの具現化の一つの方向について提案し, その実践例について紹介する。

3 課題

AI for Science の具体的な取り組みはまだ始まったばかりで、定型的な方法論の構築はこれからである。様々な分野での様々なトライアルを経て成功事例が蓄積され実用的なレベルの方法論構築に達するものと考えられる。

一方、AI 活用は情報分野でのソフトウェア開発およびサービス構築・運用の分野において急速に普及し、自然言語からのコード生成やマルチエージェント協調型の開発・運用支援が実用段階に達している [6, 7, 8]。この情報分野での動きに学んだ活用方法を考えるのが AI を研究に活用する一つの方向である。しかし、この成功事例をそのまま研究に適用すると、以下の課題に遭遇すると予想される。

1. 研究フローの多様性

研究分野ごとに実験設計，データ収集，解析手法，検証の流れは大きく異なる。

2. 研究フローの未確定性

研究は仮説検証の過程でフローそのものが変化するため、固定的な AI 支援フローを想定すると柔軟性を失うリスクがある。

3. ベストプラクティスの欠如

ソフトウェア開発にはテスト駆動開発やアジャイル開発のように広く共有された開発手法があり、それに準拠した AI の活用方法が考えられる。しかし、研究活動には分野横断的な「ベストプラクティス」が確立していない。

4 課題解決に向けて

4.1 先行研究

数学研究での AI を活用した先行研究としては、先にあげた [4, 5] のように数論データベースと機械学習を活用して数論予想を実証的に作成するアプローチが取られているものがある。これらは研究者が AI 手法を実装したプログラムを作成し、そのプログラムがデータベース活用して数学的予想を導けるデータを生成することで研究が成立している。研究者が AI を使いこなして研究を進めているという関係である。

また、分野を限定せずに研究再現性のために AI を活用している先行研究 [9] もある。この研究は AI を活用して研究の再現性を高めることを目的としている。具体的には、研究推進時に各種研究情報を再現性の観点から検証し、改善すべき点を研究者に対するコメントという形態でフィードバックする。このことで、継続的な研究の再現性向上を目指している。AI が研究者への再現性向上という特定の目的についてのアドバイザーとなるという関係である。

4.2 研究者と AI の関係

先行研究に見られる研究者と AI の関係の他に AI を共同研究のメンバーとして位置付ける関係も考えられる。これはソフトウェア開発の際に AI を共同開発メンバーとして位置付けた開発体制を考えることに触発されたものである。

研究においてこのような関係を構築する際には、前述の課題である研究フローの多様性と未確定性、ベストプラクティスの欠如に対応する必要がある。これらの課題を解決するための手段として考えられるのは研究のフローを研究者と AI の間で柔軟に制御する仕組みを提供することである。このため、研究プロセスを柔軟に記述可能な Notebook を活用し、研究者と AI が逐次フローを修正しながら研究を遂行する仕組みを導入することとする (Human in the loop)。

人間可読な自然言語を基本とする文書を基点とし、それを AI が解釈して人間可読かつ機械可読な手順書としての Notebook を生成する枠組みを活用するのである。この仕組みにより、研究者は Notebook による手順書ではなく「手順書を生成する知識文書」を作成し、共有することを中心とする活動に移行でき、かつ必要に応じて柔軟にそれを制御できる。

この仕組みにより、研究分野や研究依存のテンプレート管理と比較して、情報が凝縮された知識文書の管理を行うことが主眼となりテンプレートの乱立という課題へ対処できる。また、Notebook 生成時の環境依存性を考慮して AI は Notebook を生成するので、環境依存性も自動的に解消される。それとともに Notebook の保守管理から研究者を解放できる。さらに「手順書を生成する知識文書」は Notebook と比較して長期持続性が高いと考えられる。

本研究では AI を今までの研究体制に加えて、上記の設定でそれぞれの研究に即した研究体制を構築し AI との共同研究手法を考える。

4.3 選択した研究体制の具体化

AI が参加しない共同研究においては研究者同士がメールなどでコミュニケーションする他、共有するコードレポジトリやプログラム実行環境などの外部サービスとしての研究環境を使って研究を進める。この体制に AI を加える際の具体例を示す。

ソフトウェア開発に学んだ AI との協力の形態として、図 1 のように、LLM を対話の相手のフロントに設置し、そこから外部サービスを呼び出す体制とし、研究者は他の研究者との共同作業と同様に自然言語で AI と協力し、その際必要な外部サービスを AI が直接活用する体制とする。

4.3.1 外部サービスとの連携

外部サービスを呼び出す際には MCP 接続を活用する [10]。MCP は、AI アシスタントが外部のツールやサービスと安全に連携するための共通規格である。

従来の AI は対話のみであったが、MCP を使うことで、例えば GitHub でコードを管理したり、Jupyter Notebook で実験を実行したり、実際の作業を代行できるようになり、AI をより実践的な共同研究者として機能させることができる。

4.3.2 長期記憶の共有

先例となるソフトウェア開発分野への AI 適用において長期記憶の利用として実践されているのは、Project Knowledge と呼ばれる AI が意識できる長期記憶領域の活用である。Project Knowledge は自然言語で表現可能であるけれど、できるだけ整理して人間にとっても AI にとっても誤解なく解釈が一つになるような整理された表現とする必要がある。また、自然言語での記述に加えて参考資料を添付できるので、論文やメモ、サンプルプログラムを Project Knowledge に加えることが効果的である。

このソフトウェア開発分野での実践例にならって研究活動においても研究プロジェクト毎に Project Knowledge を共有し、人間の間の情報共有でも活用することとする。Project Knowledge を Human Readable でかつ AI Readable なプロジェクト憲章として活用するのである。

充実した Project Knowledge を共有するという事はプロジェクトの上流情報を共有するという事でもあり、プロジェクトに参加する複数の研究者の間での再現性についての意識共有にも有効で研究の再現性に役立つ。それに加えてプロジェクトに複数の AI を参加させた場合にも上流情報の共有は有効に機能すると考えられる。

4.3.3 研究への適用の特徴

前述の課題を解決するために、共同作業のフローを柔軟にする方策として、ソフトウェア開発の場合との差分として、実際に研究データや実験プログラムを扱う際の手順を Notebook として AI と研究者の間で共同執筆・共同利用する。

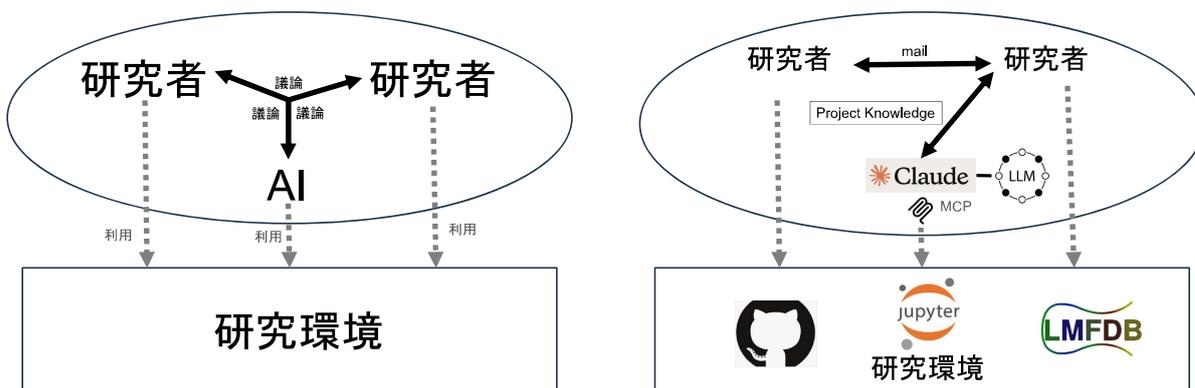


図 1: 選択した研究体制

図 2: 実践例での実際の研究体制

5 実践例

この研究体制で実験数学を AI とやってみる実践例として筆者が以前“実験数学を教育から研究までやってみる” [11] で扱った対象を使う。それは、大域体における素イデアルの偏りについての現象を深リーマン予想との関係で説明した論文“Chebyshev’s Bias against Splitting and Principal Primes in Global Fields” [12] の本文中で述べられている偏りに関する数値実験をさらに進めることであった。

“実験数学を教育から研究までやってみる”においては、元来のチェビシェフの偏りと同様にアーベル拡大での偏りについて、実験数学を研究者のみで実施した。今回は非アーベル拡大についての実験数学を AI と研究者で実施する。ただし、非アーベル拡大一般ではなく、その中でもガロア群が比較的扱いやすい四元数群拡大体についての実験を実施する。

5.1 実践例での研究体制

“実験数学を教育から研究までやってみる”では実験数学プロジェクトのベースになる上記論文を発端に、研究者間のメールベースでの会話、github での情報共有をする研究体制であったけれど、その体制に AI を図 2 の形で取り入れた。

LLM としては Claude [13, 14] を主に使い、対話の相手のフロントに設置し、そこから外部サービスを呼び出す際には MCP 接続を活用して github のプライベートリポジトリと mdx [15] の専有 CPU ノード (152 コア/243.44GB) 内に構築してある SageMath や Julia/Nemo の実行環境を共有する構成とした。また、長期記憶域としては Claude の場合 Project Knowledge 機能を主に用いた。

5.2 Project Knowledge 例

このような研究体制での鍵となるプロジェクト定義の共有の仕組みである Project Knowledge を、この実践の中で SageMath を使った Notebook を生成するために作成した Bias プロジェクトの Project Knowledge を例に紹介する。図 3 に示すようにプロジェクト定義文書として Instructions と呼ばれる文書として記述する。この Instructions から引用する形で既出論文や今後の研究の方向性を示すメモ、それと参考にできるサンプルプログラムなどを Files として登録する。

これらの情報を長期記憶のコンテキストとして活用し、それぞれのフェーズなどに対応したプロジェクト内での会話を順次進めることでプロジェクトを進行する。この会話の中でその都度フェーズに合わせて必要な Notebookなどを生成するよう AI に依頼する。

Instructions の中身は構造化された自然言語で記述しており、プロジェクト概要/プロジェクト管理と役割分担/フェーズ記述/成功基準/参考資料/実行環境/技術方針などを定義している (図 4)。この Instructions や参考資料などは AI と研究を進める中で必要に応じて改版して進める。改版の際には AI との共同執筆する。



図 3: Project Knowledge 例



図 4: Instruction 例

5.3 実験数学対象

“Chebyshev’s Bias against Splitting and Principal Primes in Global Fields” [12] の中で Theorem 2.2 の後にその非アーベル拡大における偏りの例として四元数群をガロア群に持つ拡大に関する偏りについて Example 2.1 も含め、以下のように説明されている。

Example 2.1 に述べられている偏りについての実験数学を実践例とする。この偏りを調べる初期プログラムは研究者が与え、プログラムは SageMath の `primes_above(p)` 関数や `artin_symbol(p)` 関数を用いることで記述し、Project Knowledge 内の参考情報として共有している。

L/\mathbb{Q} を有限次ガロア拡大とし、 $G = \text{Gal}(L/\mathbb{Q})$ とおく。 D_L を L の判別式とする。 G の元 σ に対し、

$$S_\sigma = \left\{ p \mid \text{素数}, p \nmid D_L, \left(\frac{L/\mathbb{Q}}{p} \right) \sim \sigma \text{ (} p \text{ は } p \text{ の上にある } L \text{ の素イデアル)} \right\}$$

とおく。ここで、 $\left(\frac{L/\mathbb{Q}}{p} \right) \sim \sigma$ は p の上にある L の素イデアル \mathfrak{p} に関するフロベニウス元 $\left(\frac{L/\mathbb{Q}}{\mathfrak{p}} \right)$ と σ が共役であることを表す。さらに、

$$\pi_{\frac{1}{2}}(x) = \sum_{p \leq x} \frac{1}{\sqrt{p}}$$

$$\pi_{\frac{1}{2}}(x; \sigma) = \sum_{\substack{p \in S_\sigma \\ p \leq x}} \frac{1}{\sqrt{p}}$$

とおく。

Theorem 2.2 (偏り) DRH(深リーマン予想) を仮定すると、 G の任意の元 σ に対し、定数 $c \in \mathbb{R}$ が存在し、

$$\pi_{\frac{1}{2}}(x) - \frac{[L:\mathbb{Q}]}{|c_\sigma|} \pi_{\frac{1}{2}}(x; \sigma) = (M(\sigma) + m(\sigma)) \log \log x + c + o(1) \quad (x \rightarrow \infty)$$

が成り立つ. ここで, $|c_\sigma|$ は σ の共役類 c_σ の位数を表す.

Example 2.1 (四元数群拡大) L/\mathbb{Q} をガロア拡大とし, そのガロア群 $\text{Gal}(L/\mathbb{Q})$ は四元数群 $Q_8 = \{\pm 1, \pm i, \pm j, \pm k\}$ に同型であるとする. $\text{Gal}(L/\mathbb{Q})$ には5つの既約表現が存在する. 非自明な1次元表現を χ_1, χ_2, χ_3 で, 2次元表現を ρ で表す. 係数 $M(\sigma)$ と $m(\sigma)$ は以下のようなになる. ここで, $m_\rho = \text{ord}_{s=\frac{1}{2}} \mathcal{L}_Q(s, \rho) = 0$

σ	1	-1	$\pm i$	$\pm j$	$\pm k$
$M(\sigma)$	1/2	5/2	-1/2	-1/2	-1/2
$m(\sigma)$	$2m_\rho$	$-2m_\rho$	0	0	0

すなわち, バイアスの向きは Artin L 関数の中心点における零点の位数によって決定される.

まず, 四元数群拡大に関する既存研究 [16] で扱われている 23 の四元数群拡大体例を Example 2.1 の具体例として考え, 素数 p に対応するフロベニウス元の共役類を調べ, その偏りについてグラフ化するなどして Theorem 2.2 を支持する実験数学を進めることとした.

5.4 並列化・高速化

SageMath の `primes_above(p)` 関数や `artin_symbol(p)` 関数を用いることで計算を進めた場合, どちらの関数も素数が大きくなるほど処理に時間がかかるため, 十分なサイズの実験をするためには処理速度をあげる必要がある. そのためにマルチコアによる並列処理化をするフェーズがあり, これは Instructions に実行環境情報と開発フェーズの作業内容として記述している.

これだけでは十分な高速化が実現できないので, 四元数群拡大の固有な数学的特徴を活用した最適化による高速化も実施するよう指示している. Instructions の中では元の参考プログラムの計算方式を Exact と呼び, この高速化を加えたものを Fast と呼んでいる.

具体的には四元数群拡大の場合, フロベニウス元は, 四元数群拡大体の定義多項式の mod p での因数分解することで分類できる. $[1,1,1,1,1,1,1]$ 型 \rightarrow 恒等元に対応, $[2,2,2,2]$ 型 $\rightarrow -1$ に対応 $[4,4]$ 型 $\rightarrow i, j, k$ のそれぞれの共役いずれかに対応し, 最後のケースについてはさらに部分体での分解パターンをルジャンドルシボルの計算により決定できる. この計算にしたがって, 各素数 p の分解型を調べることで, フロベニウス元の出現頻度を高速に計測できる. 高速版のプログラムの実行例を図 5 に示す.

この高速化が妥当であることは研究者間でプログラムコードを確認し合意することに合わせて, いくつかのサンプルについて Exact 版と Fast 版について計算実行して照合し, 差分の無いことを確認し, 高速化による実験が可能であると判断した.

5.5 言語移行による高速化

前節の並列化・高速化により 10^9 サイズの実験が1日以内に実行可能となった。さらに 10^{10} サイズ以上を実行可能とするための高速化の方法として SageMath/Python 環境から Julia 環境へ移行した。

この移行に際しては、すでに作成・利用している SageMath のコードではなく、5.2 節で述べた SageMath 用に作成した Project Knowledge を活用した。すなわち、bias プロジェクトの隣に bias-julia プロジェクトを作成し、そのプロジェクトの Project Knowledge として bias プロジェクトの Project Knowledge とほぼ同じものを用意した。差分は、使用するプログラム言語と数学ソフトウェア指定および Github 上で使うディレクトリ指定のみである。

新しいプロジェクトで新しい Project Knowledge を利用してプロジェクト新規に立ち上げることで Julia により 10^{10} あるいは 10^{11} のサイズまで実験数学が実行可能なプロジェクトが出来上がった。このプログラム言語と数学ソフトウェアの移行は Project Knowledge の活用による再現性確保効果の一つとして位置付けることができる。

bias_julia Projectでの高速化版のJulia+Nemo版作成

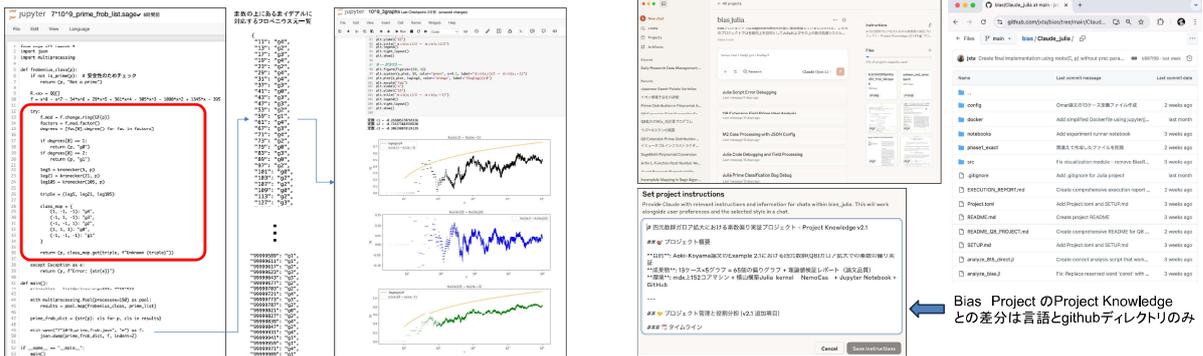


図 5: 四元数群拡大体特化高速版プログラム

図 6: Project Knowledge による言語移行

5.6 実験結果

Omar の論文 [16] で提供されている 23 Case の四元数群拡大について 10^{10} までの計算を実施し、グラフ作成まで実施している. 論文 [12] の Theorem 2.2 および Example 2.1 による理論値と実験数学で得られた計算値の比較をグラフ上で行い、図 7, 8 にその一部の Case の実験結果を示す. 他の Case についても同様に理論値に沿った結果になっている [17].

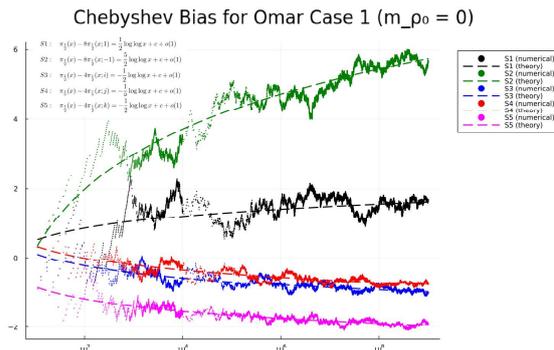


図 7: Omar Case1

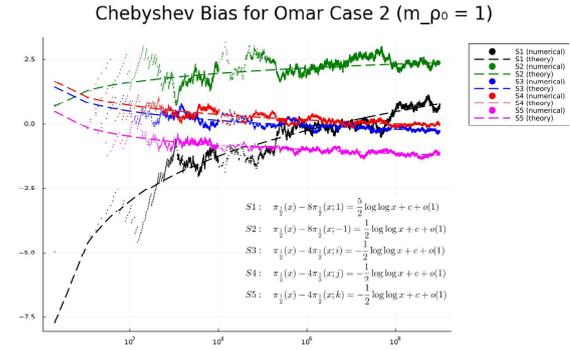


図 8: Omar Case2

6 考察と今後の展開

紹介した実践事例については、AI 活用により実験数学の実装・構築期間が短縮でき、実験結果所得までを短期間で実現可能であった. 主には前章で説明したようにプログラムの並列化、高速化に際して十分な品質で実験プログラムを AI が生成し、github で共有し、それを数学ソフト実行環境で実行し結果を得て、実験プログラムを修正する、というサイクルがスムーズに実行できたためである.

AI の持っているコンテキストが短い・小さいという課題については Project Knowledge や github などの記憶域を共有することで、実用的なレベルに回避可能であり、それが前述の実験プログラムの品質確保とサイクルのスムーズな運用につながった. さらには Project Knowledge などの活用により、プログラム言語・数学ソフトウェア間、さらに AI モデル間での移行が可能であることを体験した.

以上をまとめると、本研究では「自然言語での研究目的記述 → AI による Notebook 生成 → SageMath および Julia/Nemo での実験」という流れを実践した. 研究者は「四元数群拡大における Chebyshev の偏りを数値的に検証したい」といったプロジェクト定義文書を入力するだけで、AI がコードを生成・修正し、実験実行を支援した. これにより、実験数学が有効な数論分野において、以下の可能性があることが確認された.

1. 新しい研究仮説に応じてプロジェクト定義文書を作成するだけで研究プロセスが開始できる.

2. AIの支援により、言語や数学ソフトウェア (Julia, Python, SageMath, Nemo/Oscar) をまたいだ移植性が確保される。
3. プログラミング言語や数式処理ソフトウェアの更新に対して追従可能である手法なので従来のコード配布よりも効率的かつ持続可能な知識共有が実現できる。

これは従来の「コード共有による再現性」から「文書共有による再現性」へのパラダイムシフトを示す。これは、AI技術の進展を背景に、AI Actionable Project Knowledge Document という新しい学術知識の共有形態を確立する可能性を持つと言えるのではないだろうか。

一方、Notebook の活用については実験数学の実行部分に適用したレベルにとどまっておろ、それより大きな研究フローについての適用まで至っていない。課題解決についての評価を行うためにはより大きな研究フローについての Notebook 活用およびその AI との生成と活用についてのさらなる実践を続ける必要がある。

また、数学教育の観点から、今回提案している AI を共同研究者として取り入れた研究体制を大学院教育に導入することでその教育品質を向上させることができる可能性を持っていることが可能であることを実践の実験数学を遂行する中で感じられた。

今後、LMFDB などを活用した今回の偏りに関する実験数学を AI とともにさらに行い、数論研究を進め新しい予想および定理に辿り着き、それを論文化する際に今回提案している AI Actionable Project Knowledge Document を活用した研究再現性を実現した論文の一事例としたい。そのことを通じて数学教育での AI 活用の事例構築の可能性を具現化したい。

7 謝辞

論文 “Chebyshev’s Bias against Splitting and Principal Primes in Global Fields” の Example 2.1 に記載されている数値実験の実施にあたって、この論文の著者である島根大学 総合理工学部 数理科学科 青木 美穂 教授に常に有益な指針とご助言を賜り、ここに深謝する。この指針およびご助言がなければ、5章で説明した実験数学は実現できなかった。

参考文献

- [1] 藤原一毅, 常川真央, 合田憲人, 山地一禎: 研究再現性を支える情報基盤が持つべきデータモデルの検討, 研究報告インターネットと運用技術 (IOT) 2020-IOT-51, pp. 1 - 8, 2020.
- [2] 黒橋 禎夫, データ基盤から知識基盤へ, 情報処理 Vol.64 No.5 , pp.212-213, May 2023.
- [3] Cremona, J. E.; Jones, J. W.; Sutherland, A. V.; Voight, J., The L-functions and modular forms database, Notices of the American Mathematical Society, 68(9), 1520-1522, doi:10.1090/noti2345,2021.
- [4] Saraeb, Ali: Artificial Intelligence in Number Theory: LLMs for Algorithm Generation and Ensemble Methods for Conjecture Verification, arXiv preprint, Available at: <https://arxiv.org/abs/2504.19451>, 2025.
- [5] He, Yang-Hui, et al: Murmurations of elliptic curves. arXiv preprint, Available at: <https://arxiv.org/abs/2204.10140>, 30 July 2024.
- [6] The future of software engineering in an AI-driven world V Terragni, P Roop, K Blincoe arXiv preprint arXiv:2406.07737, 2024.
- [7] Scaffolding Metacognition in Programming Education: Understanding Student-AI Interactions and Design Implications B Ma, H Li, G Li, L Chen, C Tang, Y Xie, C Gu, A Shimada, S Konomi arXiv preprint arXiv:2511.04144, 2025.
- [8] Amplifier: Metacognitive AI Development, <https://github.com/microsoft/amplifier>, Accessed 20 Nov. 2025.
- [9] Adrien Bibal, Steven N. Minton, Deborah Khider, Yolanda Gil, AI Copilots for Reproducibility in Science: A Case Study, <https://arxiv.org/abs/2506.20130>, 2025.
- [10] Singh, A.; Ehtesham, A.; Kumar, S.; Khoei, A Survey of the Model Context Protocol (MCP): Standardizing Context to Enhance Large Language Models (LLMs)., Preprints, 2025040245, 2025.
- [11] 横山 重俊, 実験数学を教育から研究までやってみる, 数理解析研究所講究録 2236 pp.62-71, 2022 年 12 月.
- [12] Aoki, M., Koyama, S. Chebyshev's bias against splitting and principal primes in global fields. J. Number Theory, 245, pp.233-262, 2023.
- [13] Bai, Y., et al. , Constitutional AI: Harmlessness from AI Feedback. arXiv. <https://arxiv.org/abs/2212.08073>, 2022.

- [14] Anthropic, Claude 3.7 Sonnet — System card. <https://www.anthropic.com> , Accessed 20 Nov. 2025.
- [15] T. Suzumura et al., "mdx: A Cloud Platform for Supporting Data Science and Cross-Disciplinary Research Collaborations," 2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, pp. 1-7, 2022.
- [16] Sami Omar: On Artin L-Functions for Octic Quaternion Fields, Experiment. Math. 10(2): 237-246, 2001.
- [17] Github quaternion-bias-project repository, <https://github.com/jxta/quaternion-bias-project>, Accessed 20 Nov. 2025.