

生成AIの普及によるプログラミング教育への影響 ～プログラミング教育はいかに変わるべきか？～

Impact of Popularization of Generative AI on Programming Education
～How should programming education change?～

山本 誠¹⁾

Makoto Yamamoto

1) 工学博士 東京理科大学 嘴託教授 (〒125-8585 東京都葛飾区新宿6-3-1, E-mail: yamamoto@rs.tus.ac.jp)

Generative AI, including ChatGPT, has become widespread, and many students have begun to use it to write reports and solve exercises. Even in the programming education course "Programming Basics" that the author is in charge of, reports and programs that seem to use generative AI are now appearing here and there. In this paper, I discuss the current use of generative AI and its (likely) impact on programming education under these circumstances and consider what programming education should look like in the future. Programming is an essential subject for computational engineering, and I hope that this paper will help to build a better programming education that matches the times.

Key Words: Programming Education, Generative Artificial Intelligence (AI), New Lecture Methods

1. はじめに

ChatGPTを始めとした生成AIが広く普及し^[1]、多くの学生がレポート作成や演習課題の解決等に生成AIを利用し始めている。著者が担当しているプログラミング教育「プログラミング基礎」においても、生成AIを利用したと想像されるレポートやプログラムが散見されるようになっている。本稿は、このような状況の下、現在の生成AIの利用状況やプログラミング教育に及ぼす（であろう）影響について論じ、これからプログラミング教育のあるべき姿を検討したものである。計算工学にとってプログラミングは必須科目であり、本稿が将来の一層進化した情報化社会にマッチするより良いプログラミング教育を構築するための一助となれば幸いである。

2. 生成AIの普及

2022年11月にOpenAI社からChatGPTが発表されたことに端を発し、ChatGPT (OpenAI)、Copilot (Microsoft)、Gemini (Google)、Claude (Anthropic) など多数の生成AIが無料で利用できるようになっている。また、より高機能な有料版も販売されており、ChatGPTの発表から2ヶ月で1億ユーザを獲得するという驚異的な成長が記録されている。このように生成AIが急速に普及していることは、皆様ご存じの通りかと思う。

学生にも生成AIの利用が浸透しつつある。大森不二雄氏（東北大学高度教養教育・学生支援機構・教授）が2024年5月から6月にかけて実施したアンケート調査^[2]（回答者数4千人超）によれば、ChatGPTに関して学生は以下のよ

- ・大学生全体のChatGPT利用率は約40%であった。
- ・回答者の32.4%に使用経験があった。

- ・使用経験は、男性が44.8%、女性が27.1%（男性優位）
- ・理工農系の学生の利用率が高かった。
- ・学生全体の約15%がレポートや提出物の作成にChatGPTを利用している。
- ・そのうち92%の学生が内容の正確さを確認した。
(逆に言うと、8%の学生が正確さを確認していない。)
- ・また、85%の学生が自分の考えで文章を編集した。
(逆に言うと、15%の学生はChatGPTの出力した文章をコピー・ペーストしてそのまま使用した。)
- ・学生全体の77%が文章力の向上に役立つと回答した。
- ・71%が思考力の向上に寄与したと回答した。

このように、生成AIの普及は急速に進んでおり、今後、ますます加速度的に拡大していくことが予想されている。したがって、大学教育（初等・中等教育も？）においては、生成AIの利用を前提とした教育方法・内容にシフトせざるを得ないのが現状ではないかと思われる。

3. プログラミング教育の目的

プログラミング教育の目的は何であろうか。ChatGPT (OpenAI)、Copilot (Microsoft)、Gemini (Google) に聞いてみたところ、プログラミング教育の目的はかなり幅広く捉えられているようである。上記3つの生成AIの回答をまとめると、プログラミング教育の目的は概ね以下のように表現できる。

（1）論理的思考力の向上

プログラミングにより、問題を論理的に分解し、順序立てて解決する能力（アルゴリズム的思考）が獲得できる。

（2）問題解決能力の涵養

自らプログラムを書くことで「どうすれば効率的に課題を解決できるか？」を考える力が養われる。また、バグ・

フィックス等の試行錯誤を通じて、失敗から学びながら継続的に改善する姿勢・態度が獲得できる。

(3) 創造力の向上

自分のアイデアを形にする手段としてプログラミングを活用できる。ゲームやアプリを作ることで、創造的な発想を自ら実現する力が鍛えられる。

(4) コンピュータやデジタル技術への理解

コンピュータやソフトウェアの仕組みを理解することは、現在の情報化社会を生き抜くために必須である。プログラミングを学習することで、単なる「利用者（User）」ではなく「開発者（Developer）」としての視点を持つことができる。

(5) チームワークやコミュニケーション能力の向上

プログラミングは単独作業だけでなく、チームで協力して開発を進めることが多い。他者と意見を交換しながらプロジェクトを進める経験を通じて、コミュニケーション能力の向上が期待できる。

(6) 将来のキャリア形成

現在、理系・文系を問わず、あらゆる分野においてプログラミングスキルが求められている。ITエンジニアに限らず、様々な分野でデジタル・トランスフォーメーション（DX）が進んでおり、将来のキャリア形成にプログラミング能力を活用できる。

(7) デジタル社会でのリテラシー向上

インターネットや生成AIが発展する中で、デジタル技術を正しく理解することは社会生活においても重要である。プログラミングを学習することにより、情報技術の仕組みを理解し、情報リテラシーを高めることができる。

著者が担当している「プログラミング基礎」という学部1年生向け講義では、「コンピュータの概要を理解し、C言語によって基本的なプログラムが作成・コンパイル・実行できるようになることを目標とする」とシラバスに記載しており、コンピュータリテラシーの涵養とプログラミングスキルの獲得を重視している。恐らく、多くのプログラミング教育担当者は、基本構文・コマンドの理解、課題解決のためのアルゴリズムの構築、それによる論理的思考力の涵養といった要素を重視しているのではないかと思われる。

4. 生成AIのプログラミング教育への影響

著者は、前述のように、工学部機械工学科1年生向けの「プログラミング基礎」という講義を担当している。この講義は、反転授業形式で行われ、ほとんどの講義時間が演習に費やされる。本講義では、標準入出力、for文、if文などの基本構文を利用して、代数方程式の求解、データのソート、熱伝導方程式や振動方程式の求解（差分法）、アニメーション作成などのプログラミング演習を通じて、プログラミングスキルの育成を図ることを主たる目的としている。また、講義時間内に基本的な3～5本のプログラムを各自作成し、その発展的なプログラムの作成を宿題

（復習）として課して提出させている。

最近、提出された宿題プログラムに講義で学習していない構文やコマンドが散見されるようになり、学生との雑談を通じて、一定数の学生が生成AIを利用して宿題に取り組んでいることが分かった。

試しに、演習で学生に作成させる台形法則を用いた数値積分のプログラムをChatGPTに作らせてみた。ChatGPTに「関数 $y=x^2+2.0*x+3.0$ を $x=0$ から $x=1$ まで台形法則を用いて数値積分するC言語のプログラムを作ってください」というプロンプトを入力してみたところ、ChatGPTは瞬時に表1のプログラムを出力した。

表1 ChatGPTが出力した数値積分プログラム

```
#include <stdio.h>
// 被積分関数 y = x^2 + 2x + 3 (单精度版)
float func(float x) {
    return x*x + 2.0f*x + 3.0f;
}
// 台形法による数値積分 (单精度版)
float trapezoidal_rule(float a, float b, int n) {
    float h = (b - a) / n; // 分割幅
    float sum = (func(a) + func(b)) / 2.0f; // 端点の平均値
    for (int i = 1; i < n; i++) {
        float x = a + i * h;
        sum += func(x);
    }
    return sum * h;
}
int main() {
    float a = 0.0f, b = 1.0f; // 積分範囲 [0, 1]
    int n = 1000; // 分割数 (大きいほど精度が向上)
    float result = trapezoidal_rule(a, b, n);
    printf("数値積分の結果 (单精度): %f\n", result);
    return 0;
}
```

このプログラムをコンパイル・実行すると、4.333333という正解が出力され、正しいプログラムであることが確認できた。また、続けて、「倍精度計算ではどうなりますか？」という追加のプロンプトを入力すると、倍精度計算のプログラムが出力され、さらに、「Fortranでは？」、「Pythonでは？」と追加入力すると、それぞれの言語に応じたプログラムが適切に出力された。

ちなみに、ChatGPTに「バックステップ流をナビエ・ストークス方程式を用いて計算するプログラムをFortranで作ってください。ただし、ステップ高さを代表長さ、入口流速を代表速度として定義されるレイノルズ数は100とする」というプロンプトを入力すると、形状および境界条件を自動設定し、MAC法のプログラムが出力された。また、「SIMPLE法ではどうなりますか？」と追加入力する

と、これも妥当なSIMPLE法のプログラムが出力された。なお、Copilotでも同様の結果が得られた（残念ながら、Geminiでは失敗）。

以上のように、現在の生成AIは、初級者向けのプログラミング教育で扱われるような簡単なプログラムであれば、プログラム言語を問わず正しいプログラムを出力できる能力を有しているようである。

学生が生成AIを利用した場合、何が起きるであろうか？学生が生成AIを利用してしまった場合、学生に論理的な思考を深めてプログラミングスキルを向上させてもらうというプログラミング教育の意義がほとんど失われてしまうことは容易に想像できる（人間が安易に流れるのは必然！）。恐らく、かなりの数の学生が生成AIを利用し、その回答に盲目的に従ってしまうのではないだろうか。文部科学省も大学教育における生成AIの利用に関して危惧を表明し、2023年7月という早い段階で「大学・高専における生成AIの教学面の取扱いについて」^[3]という通知を発出し、注意を喚起している。したがって、従来の講義方法に基づくプログラミング教育によって教育目的を達成することは今や困難となりつつあり、講義方法を変更したり、宿題や課題の出し方を工夫するなど、何らかの対応が必要になっているのが現状と言えるであろう。

5. これからのプログラミング教育

ここでは、以上の現状に鑑み、これからのプログラミング教育について考えてみたいと思う。

まず、前提条件として、生成AIの利用を許可するか、許可しないかで講義方法が大きく変わるとと思われる。そもそも学生に生成AIを利用させなければ、従来型の講義が実施可能である（実際に生成AIを利用させない手段があるか否かは不明）。しかし、生成AIの活用スキルがこれから技術者にとって必須なスキルであることを考慮すると、学生が生成AIを利用する前提で新たな講義方法を考えるべきであろうと思われる。

では、これからのプログラミング教育はどうすれば良いであろうか？生成AIを利用して実施するプログラミング教育として著者が思いついた講義形式は、残念ながら、4つしかない。それぞれの利点・欠点を含めて以下に列記する。

- ① プログラムの作成演習（キーボード入力）は従来通りとするが、バグ・フィックスに生成AIを利用させる講義：この方法では、効率的にプログラムを完成させられるが、バグ・フィックスを通じて養われる論理的な思考力は育成されない恐れがある。また、初めから生成AIを利用してプログラムを出力し、それをコピー・ペーストされてしまうと、プログラムの作成演習自体が成立しなくなる。
- ② 生成AIに正しいプログラムを出力させるプロンプトの入力方法に特化する講義：この方法では、生成AIの利用スキルが涵養され、プロンプトを組み立て

る際に一定の論理的思考力が鍛えられる。しかし、アルゴリズムや効率的な処理方法に関するスキルを修得することは難しく、オリジナルのプログラムを作成する能力は育成されがたい。

- ③ 生成AIに正しいプログラムを出力させた後に、中身の解説を行い、アルゴリズムやコマンド等を理解させる講義：この方法では、生成AIへの適切なプロンプトの入力方法を修得でき、ある程度のアルゴリズム等の理解もできる。しかし、生成AIに対する絶対的な信頼感を学生に抱かせてしまうリスクがある。
- ④ バグを多数埋め込んだプログラムを事前配布し、最初に独力でバグ・フィックスを行わせ、最後に生成AIの意見を聞いて正解を確認させる講義：この方法では、アルゴリズムの内容を理解できとともに、論理的思考力の向上が望めるが、学生は最初から生成AIに正解を聞いてしまう可能性が高い。

ここに挙げた例以外の講義形式もあると思うが、それらも含めて、どのような講義形式がこれからのプログラミング教育として望ましく、高い教育効果が得られるかは、今後の議論・検討・試行・検証の結果を待たざるを得ないかもしれない。場合によっては、プログラミング教育の意義が失われたので、プログラミング教育を廃止するということもあり得る（極論であるが、、、(^;))。皆様からの忌憚のない意見をいただければ幸いである。

6. おわりに

本稿では、生成AIの普及によるプログラミング教育への影響と将来のプログラミング教育のあるべき姿を検討した。生成AIの機能は日進月歩で進化している非常に便利なツールであり、生成AIの有効利用はこれからの技術者にとって必須のスキルであると考えられる。しかしながら、生成AIに頼りすぎることは論理的思考力の減退につながりかねず、教育上のリスクを十分に認識しておくべきであろう。最後に、若手の大学教員には、プログラミング教育における生成AIの活用を通じて、将来のより良いプログラミング教育（その他の科目も？）を創出し、計算工学のさらなる発展に寄与していただけることを祈念している。

参考文献

- [1] 総務省, 情報通信白書・特集(2) 進化するデジタルテクノロジーとの共生, <https://www.soumu.go.jp/johotsusintokei/whitepaper/r06.html>, 2024. (2025年4月閲覧)
- [2] 大森不二雄, 大学教育と情報, 2024年度, 第3・4号, p.115, 2025.
- [3] 文部科学省, 大学・高専における生成AIの教学面の取扱いについて, https://www.mext.go.jp/b_menu/houdou/2023/mext_01260.html, 2023. (2025年4月閲覧)