計算工学講演会論文集　Vol.29 (2024年6月)　日本計算工学会

計算工学技術者の情報教育で用いられるPythonコードの実用性評価

Evaluation of the Utility of Python Codes used in Information Education for Computational Engineers and Scientists

荻野正雄[1)]，小川哲平[2)]，北川光介[2)]，杉本陸[2)]
Masao Ogino, Teppei Ogawa, Kosuke Kitagawa, and Riku Sugimoto

1) 博(工) 大同大学　情報学部　准教授（〒457-8530 愛知県名古屋市南区滝春町10-3, E-mail: m-ogino@daido-it.ac.jp)
2) 大同大学　情報学部情報システム学科（〒457-8530 愛知県名古屋市南区滝春町10-3）

Computer science education is essential for training engineers in computational engineering and science. The traditional engineering curriculum in university has used Fortran and C as computer programming languages. Recently, however, Python, which is easy to learn and highly functional, has been increasingly introduced into the educational field, which is particularly important in the field of computer science, where complex calculations and analysis of results are required. In this study, we evaluate the utility of Python code in introductory textbooks and other materials.

Key Words : *Computer Science, Engineering Education, Informatics, Python*

1.　はじめに

理工学系出身が比較的多いと推測される計算工学分野の研究者や技術者に対し，情報科学や計算機を体系的に学ぶ仕組みをどう整備するかは計算工学分野の発展において重要である[1]．特に，プログラミング，アルゴリズム，データ構造，コンピュータの構成，数値計算法などの計算科学基礎科目，情報科学の観点から離散数学や確率統計などの数学科目を学ぶことが望ましい．そのような情報学の学びにおいて，プログラミングの知識と技術は重要な素地となる．

計算工学技術者のプログラミング教育においては，プログラミング言語としてFortranやCが長らく用いられてきた．一方，近年は学習のしやすさと開発支援機能の充実からPythonへの注目が高まっており，TIOBE Index [2]などでも上位に位置している．実際に，計算工学に関連する専門基礎科目でPythonを採用する大学等がでてきており，そのための入門テキストが多く出版されている．しかし，インタプリタ型言語であるPythonはコンパイラ型言語であるFortranやCに比べて処理速度が劣るとされており，実システムのカーネル部分をPythonで開発することは少ないと思われる．例えば，大学の講義でクイックソートやガウスの消去法のプログラムをPythonで書いた場合に，そのコード資産にはどのような価値があるのだろうか．本稿では，大学の情報系科目で学ぶことが多いソートアルゴリズム，計算工学技術者として学ぶことが多いポアソン方程式の有限要素解析を例に，Pythonコードの実用性を評価する．

2.　ソートアルゴリズムの評価

(1) Python入門テキストのコード評価

代表的なソートアルゴリズムであるバブルソートとクイックソートについて，Pythonコードの評価例を示す．バブルソートは隣接する2要素の比較を繰り返すことで並び替えを行うアルゴリズムである．そのコードは2重ループ構造になり，入力データサイズNに対して時間計算量は$O(N^2)$である．クイックソートは分割統治法に基づくソートであり，ピボットより小さい要素と大きい要素に2分割する操作を再帰的に繰り返すことで並び替える．そのコードは再帰関数で実装され，時間計算量は$O(N \log N)$である．

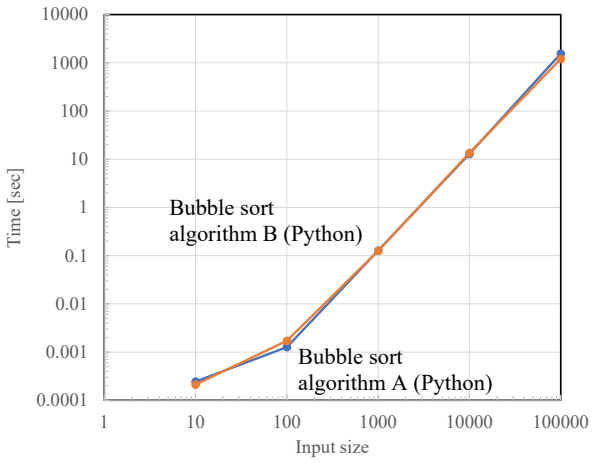

Fig. 1　Computational time for bubble sort algorithm

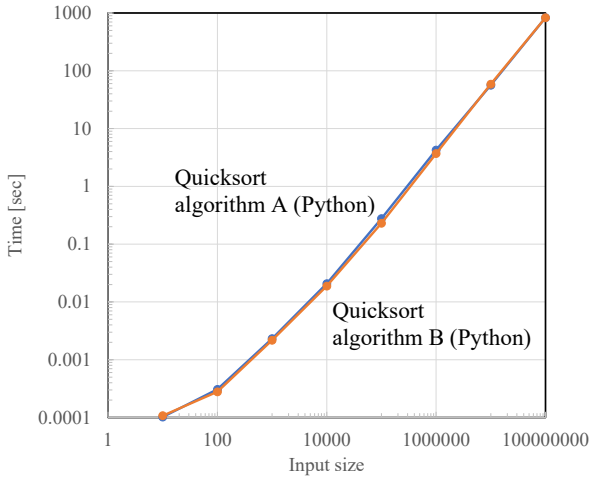

Fig. 2 Computational time for quicksort algorithm

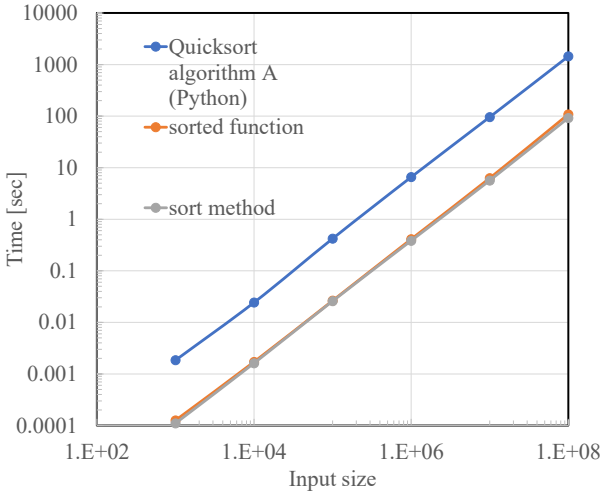

Fig. 3 Comparison between a quicksort algorithm and sort modules

ある入門テキストに掲載されていたバブルソートとクイックソートのPythonコード評価結果をFig.1と2にそれぞれ示す[3]．Pythonコードはテキストによって記述が異なることが多いため，今回は2冊のテキストを適当に選定し，それぞれの結果をalgorithm A，Bとして示している．

まず，教育の観点から評価する．アルゴリズムの理解を深める演習として，入力サイズに対する計算時間の増加割合が時間計算量と同等であることを確認することがある．それらに大きなずれがある場合は，学習に適さないと言える．Fig.1と2より，どちらも時間計算量と同等の計算時間増加割合を示しており，今回用いたPythonコードはアルゴリズム学習に有用であると言える．

次に，実用性の観点から評価する．ソートはそれ自体が目的であることは少なく，複雑な問題を効率的に解くためのデータ操作として行うことが多いため，できる限り短い時間で完了することが望まれる．その基準値を0.1秒とし，実システムでも利用されているクイックソートに着目すると，今回の実験条件ではデータサイズ1万が実用に耐えられる上限であった．これより，今回用いたPythonコードは比較的小規模なデータであれば実用性があると言える．

ここで，Pythonは標準機能として関数sortedやメソッドsortが提供されている．入門テキストのクイックソートプログラムとそれらモジュールの計算時間比較をFig.3に示す．図より，それらを用いることで計算時間を1桁程度下げることができ，数十万の中規模データでも実用性のあるコードにすることができる．しかし，これではソートアルゴリズムを学習してもコードを自作する必要性がほぼないことになってしまい，学習の形骸化につながってしまう恐れがある．

(2) PythonコードからCコードへ

計算工学技術者が情報系科目を学習する場合は，Pythonの方がFortranやCに比べて学びやすい．しかし，前項で述べたように，その際に開発したPythonコードは小規模問題に対してしか実用性がないという問題がある．そこで，資産の有効活用方法として，PythonコードからCコードへの変換を提案する．Fig.4は，Python入門テキストのシェルソートに関するコード，C言語入門テキストのコード，当時学部4年生であった共著者1名がPython入門テキストのコードを見ながら開発したCコードの計算時間を比較したものである．図より，2つのCコードはほぼ同程度の性能を持ち，Pythonコードに対して計算時間を2桁程度下げることができている．これより，計算工学技術者が情報学を体系的に学ぶ場合は，Pythonを用いて効率的に学習し，実システムに必要なアルゴリズムはC言語へ書き換えれば良いと言える．

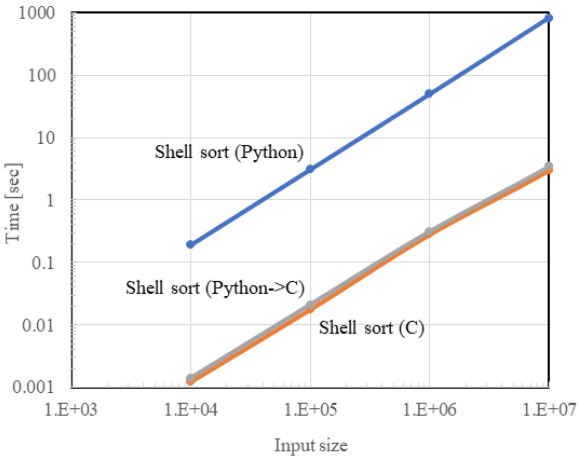

Fig. 4 Comparison of Python, C, Python->C codes in Shell's sort algorithm

(3) PythonコードからCコードへの注意点

PythonコードからCコードへ書き換える場合の注意点について述べる．Pythonには簡潔で読みやすく書くことがPythonらしさ(Pythonic)であるとする考え方がある．そのため，PythonはCとほぼ同様の書き方もできるが，Cではあまり見ない書き方を採用することがある．ここではその例をいくつか示す．

配列(リスト)x内のi番目とj番目の値を交換する場合，Pythonでは多重代入を用いることが多い．Cでは一時的

な変数を用いた典型的な値交換プログラムを書く必要がある．

Python	x[i], x[j] = x[j], x[i]
C	tmp = x[i]; x[i] = x[j]; x[j] = tmp;

配列xの要素数nを取得する場合，Pythonでは関数lenを用いることが多い．Cでは同等の関数は提供されていないため，sizeof演算子を用いたプログラムを書く必要がある．ただし，このCコードは静的配列を宣言した関数内で有用であることに注意する．例えば関数へ仮引数で渡した配列に対しては利用できないため，Cでは配列生成時に要素数を変数に記憶するプログラムが望ましい．

Python	n = len(x)
C	n = sizeof(x) / sizeof(x[0]);

変数nの値の大きさだけ配列を宣言する場合，Pythonではリストを使った簡便な書き方があるが，C言語における静的配列宣言では大きさを定数で指定する必要がある．そのため，例えば動的メモリ確保プログラムを書く必要がある．

Python	x = [0] * n
C	x = (int *) calloc(n, sizeof(int));

3. 有限要素解析コードの評価

(1) 評価用コードの概要

2次元ポアソン方程式の有限要素解析を対象に，PythonコードとCコードの性能評価を行う．

Cコードは文献[4]に掲載されているものを用いる．同次Dirichlet境界条件問題を解くものであり，3角形1次要素で離散化することで得られた連立一次方程式をガウスの消去法で解くプログラムとなっている．また，係数行列は密行列に格納される．なお，メッシュデータを外部ファイルから読み込めるよう改良した．以降，このコードをFEM_Cと呼ぶ．

次に，FEM_Cを見ながら，Pythonで書き換えたコードを開発した．それぞれの言語特有の命令を除けば，行単位で比較できるほど一致したものである．ただし，配列生成にはNumPyモジュールを用いている．以降，このコードをFEM_Pyと呼ぶ．

次に，FEM_PyをPythonらしく書き換えたコードを開発した．以降，このコードをFEM_Pythonicと呼ぶ．修正箇所は主にfor文の部分である．修正例を以下に示す．

FEM_Py	for k in range(i+1,m): a[j][k] -= aa * a[i][k]
FEM_Pythonic	a[j,i+1:m] -= aa * a[i,i+1:m]

(2) 実験結果

3つのコードの計算時間を測定した結果をTable 1に示す．ただし，FEM_Pyは節点数1,273のモデルを実用的な時間内に終了することができなかった．前節では，Pythonコードから作成したCコードは最初からCで開発したコードと同程度の性能であった．しかし，表より，Cコードから開発したPythonコードは処理時間が大幅に増加することが分かる．これは，Cで開発してきたシステムをPythonへ変更する場合に留意すべき点である．Pyhonicなコードとすることで計算時間を大幅に短縮することができるが，Cコードよりも数倍遅いことが分かる．

これまで蓄積されてきたCコード資産をPythonコードへ変換する場合，計算時間増加というデメリットが大きく，CコードとPythonコードをどう共存させるかが重要と言える．

Table 1 Comparison of computational time of FEM codes by C, Python, Pythonic, or HPP

Number of nodes	FEM_C [sec]	FEM_Py [sec]	FEM_Pythonic [sec]
335	0.074	10.35	0.254
1,273	0.690	N/A	4.207

4. まとめ

計算工学技術者が情報教育を受ける場合に今後主流となるであろうPython言語について，入門テキスト等で示されているPythonコードの性能評価を行った．学習時に開発したPythonコードは，教育効果はあるものの実用性は低いと言える．しかし，PythonコードからCコードへ書き換えることで，それらコードの実用性は大きく改善できる．また，その書き換え技術を身に付けるためには，Cの知識を有した状態でPythonを学ぶことが良いと思われる．このことから，FortranやCなど実システムで必要となるプログラミング言語の教育を行ったのちに，Pythonを用いた分野に関連のある情報教育を行う，という流れが考えられる．

謝辞: 本研究の一部は大同大学情報学部情報システム学科の卒業研究として実施されたものである．

参考文献

[1] 荻野正雄: 計算工学技術者育成における情報学の役割，第28回計算工学講演会, 2023.
[2] TIOBE Index: https://www.tiobe.com/tiobe-index/
[3] 小川・北川・杉本: Python入門テキストにおけるPythonコードの実用性評価，大同大学情報学部情報システム学科2023年度卒業論文, 2024.
[4] 菊地文雄: 有限要素法概説，サイエンス社, 1999.