Pythonによる1次元不定流の研修用プログラムの試作と開発環境の統合化
Trial code for one dimensional unsteady flow and synthesis of development environment

丹治　肇*，桐　博英*，中矢哲郎*，安瀬地一作*，成岡道男*
TANJI Hajime*, KIRI Hirohide*, TETSUO Nakaya*, AZECHI IssakuP, NARUOKA Michio*

1. はじめに 1次元の不定流計算は，1973年に白石が初めて，FORTRANのコードを公開して以来1980~2000年頃まで，水理学の数値計算の研修に，国内はもとより，JICA，タイ，マレーシアなどで広く用いられてきた．その理由としては，伝統的な水理学の初歩は，流体力学と異なり，微分と積分を使わず，定常流を対象とするため，Euler座標系とLagrange座標系の区別がなされないこと，偏微分方程式の初期値問題が理解されないことなどの問題点を抱えており，不定流の研修は，こうした問題を解決する手法として有効であったためと考えられる．2000年以降は，C++, JAVAなどのオブジェクト指向言語が普及し，初歩に教育すべきコンピュータ言語はもはやFORTRANではないが，代替は１つは絞りきれない．例えば，R言語は統計計算には適しているが，速度が遅いループ演算は数値計算には使えない. OSも作ることができる汎用性の高いc言語は機能が多いため初心者には難しい．オブジェクト指向言語では，変数（オブジェクト）が出現した時点で，言語が，変数の属性（スカラー，配列，整数，実数，文字）を判断する．この場合に，宣言文で変数の属性を定義することは邪道である．しかし，この機能を使うと実数型のつもりが文字型などの型エラーが数値計算では頻発する．このためプログラムの冒頭で配列宣言して，オブジェクト指向の利点を殺した数値計算のテキストが多い．

2. Pythonの特徴　Pythonは，1991年にヴァンロッサムが開発した言語で，2000年にバージョン2が，現在は，バージョン3が出ており，現在，２つのバージョンが広く使われている．ふるいバージョン2が使われている理由は，ライブラリーの改定が追い付いていないためである．Pythonの特徴に言語仕様が占める割合は1/3程度であり，残りはラッパ機能とライブラリーが占める．言語仕様の特徴は，予約語が少なく，可読性が高いこと，スクリプト型言語であり，コードが短い点があげられる．ラッパ機能とは，JAVA等の他言語で作成されたライブラリーをユーザがあたかもPythonのライブラリを利用しているかのように，利用できる機能である．この機能を活用したPythonを使ったアプリケーションもある. Pythonを拡張した文法を持つ数式処理システムSAGEはMathematicaなど数式処理システムをフリーに提供する目的で開発された. SAGEは開発コストを節約するために既存の資産を利用する戦略がとられた．SAGEのHPには「NumPy, SciPy, matplotlib, Sympy, Maxima, GAP, FLINT, R and many more」と書かれている．最初の4つはPythonのライブラリーである．Maximaは歴史のある，GAPは群論に強い，FLINTは数論に強い数式処理ソフトである．Rは現時点での標準統計解析ソフトである．SAGEを使えば，ラッパ機能により，１つの文法でこれら全てが使いこなせる．流出モデル作成用に，Python言語を拡張した例にPCRasterがある. Python言語で開発されているQGISには, GRASS,SAGAなどの他のGISソフトを呼び出して使うラッパ機能がある. 数値計算ライブラリーの充実

*農研機構農村工学研究所 National Institute for Rural Engineering, NARO
キーワード：　不定流,Python, 研修, オブジェクト指向, 開発環境

は，Rubyと比べると際だっている．Numpyとその拡張であるScipyは数値計算用のライブラリーであり，c言語で実装されている．このため，Pythonはインタープリタでありながら，比較的速い計算速度を実現できる．包含関係にあるライブラリーが多いことは，Pythonが使いやすい理由の 1 つである．更に，速度を求める場合には，GPUに対応するPyCUDAも利用できる． Matplotlibは非常に多機能で強力なグラフィクライブラリである．Rにも，Tellus, ggplotなどのグラフィックライブラリーがあるが，PythonのMatplotlibはより機能が多く，デファクトスタンダードである点で使いやすい．

PythonにはZenと呼ばれる設計思想書が添付されている．そこでは，オブジェクト指向ではあるが，現実に使いやすいように実装しやすい言語仕様を認めている．Pythonではプログラムを可読性の高いオブジェクト指向にも，実行効率の高い非オブジェクト指向にも書くことができる．このためとりあえずPythonを利用することは，初心者にも容易であるが，効率性と可読性のバランスをとったプログラムを書くためには選択の幅が広く，上級者にも魅力のある言語になっている．

3. 不定流プログラムの移植 Python はオブジェクト指向言語のため変数の自動判定を使うと数値計算でエラーが発生し易いが．Numpy を使うことで数値専属の配列をつかうことができ，この問題が回避できる．オブジェクト指向言語は，コードの可読性は高いが計算効率が悪い．FORTRAN や C のように数値計算を主体に設計された言語は，計算効率は高いが，可読性が低い．Python ではライブラリーを使うことで，1 つのプログラムにこの 2 つの部分を混在させることが可能である． Python に不定流プログラムを移植した． Python では，ループ計算より，ベクトル間の演算の方が高速である．ここでは，この特徴を生かして，ループを減らすとともに，2 重ループを排除した．また，if 文と goto 文を排除した．図-1 に示すように，コード長さは，もとの FORTRAN の 20%程度である．オブジェクト指向を使ってコードを短くし，計算速度をあげる上では，数値計算の開発言語としては C++が最良の選択であろう．しかし，Python は研修用に向いたインタープリタ言語であること，コードがより短く，可読性が高いこと，20 年間の CPU の改善は，インタープリタでも昔のコンパイラ以上のレスポンスを可能にしていることを考えると数値計算の情報処理教育に使う言語として有望と思われる．ラッパ機能が強力なこと，ライブラリーが豊富なこと，GPU ライブラリーや Python を C++に変換してコンパイルすることで実行速度を速める Cython を利用するなど，実行速度の改善の余地が高いことを考えれば，研究開発用言語としても，Python は有望な選択肢と思われる．

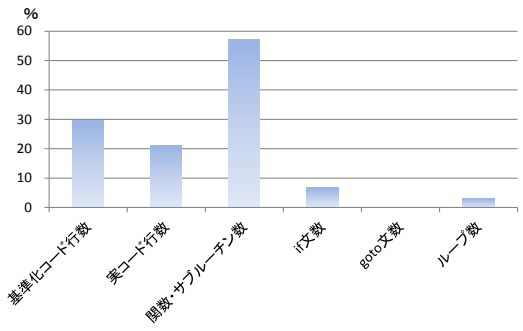

図-1　FORTRAN と Python の不定流計算コードの比較