環境流体解析におけるスクリプト言語の活用

新谷哲也（首都大学東京）・中山恵介（北見工業大学）

環境流体解析には，計算速度が速い静的な型付けを行うコンパイル型言語が広く用いられている．現存する多くの海洋・海岸・陸水を対象とした数値流体モデル(ELCOM, ROMS, FVCOM, UnTRIM, SUNTANS等)は，その代表格であるFORTRANやC言語によって構築されており，研究・実務に幅広く用いられている．これらのモデルは，様々なアプリケーションを想定して豊富なオプションが用意されているが，近年の多様な適用先とその目的を事前にすべて想定することは不可能である．そのため，ソースコードを取得して自ら修正・コンパイルするか，開発者へ依頼して修正がモデルに組み込まれるのを待つことになる．しかしながら，特に研究の場合，改良・修正は変数やパラメータの追加，式の変形を伴うことが多く合理的な結果を出すまで試行錯誤の過程が必要になる．そのため，結局，モデルのソースコードを修正することにが現実的な手段なるが，開発者が自分自身ではない場合は独自に修正した内容はモデルの次期バージョンで採用されるかわからないこと，また部分的に異なる類似のソースコードが氾濫して管理できなくなることなど問題が山積みとなる（実現可能であるが，複雑さが増大する）．本論文では解決策のひとつとして，スクリプト言語を利用した手法を紹介する．

スクリプト言語は，古くはUnixのシェルスクリプト，最近ではPythonやRuby, JavaScriptなど数多く開発されてきている．スクリプト言語は，比較的単純なプログラムを構築する際に利用される他，既存のプログラムを複数組み合わせるためのグルー（糊）言語として用いられることが多い．利用のメリットとしては，コンパイルを伴わないためにプログラム修正と実行が迅速に行えること，動的な型付けに加えて最近では関数型言語やオブジェクト指向言語で重宝される多くの機能が取り込まれ，プログラムデザインが容易になっていることなどが挙げられる．一方，よく知られたデメリットとして実行速度がコンパイル言語に比べて非常に遅いことが上げられる．近年のコンピューターの進歩によって，多くの問題がスクリプト言語で実用的に実行可能になっているが，３次元流体シミュレーションをスクリプト言語で実行するのは今でも現実的ではない．

本研究では，両者の良いとこ取りである，修正が想定される部分だけをスクリプト言語で対応し，速度が重要な流体の基本的な計算部分（修正があまりない部分）はコンパイル型言語(C++)で対応するデザインを提案する．この手法を用いれば，修正の度に再コンパイルすることもなく，保守面で考えても修正部分だけを記述したプログラムを管理すればよく，複雑さを最低限に抑えることが出来る．この試みは，特に新しい手法ではなく，コンピュータゲームの作成過程において，ゲームのストーリー（台本）を担当する部分（変更が多い・プログラムの専門家でない人が修正：スクリプト言語）と３次元描画や計算の部分（負荷が高い：コンパイル言語）を分離することは頻繁に行われている(浜中, 2010)．また，最近の商用流体モデルではマクロ機能（VBAの様なもの）を有しているソフトウエアーもあり，類似の事は出来る．本論文では，提案するコンビネーションが比較的容易に実現可能であること，そして，流体モデルと組み合わせる場合に考えた２種類のデザインを報告する．

スクリプト言語を流体モデルと組み合わせる方法として２種類考えた（図１）．(1)流体モデルにスクリプト言語（Lua, http://www.lua.org）を内部言語として組み込む，(2)スクリプト言語(Python, https://www.python.org)から流体モジュールを呼ぶ．どちらの場合も，コンパイルされたモデル（モジュール）とスクリプト言語が相互に通信できることが必要がある．そのような両者の橋渡しをする（ラッパー）関数を自動的に生成するソフトウエアーSWIG(http://www.swig.org)があり，このSWIGで生成された関数群をモデルと同時にコンパイルすることで，スクリプト言語とやりとりが可能な実行ファイルができる．(1)の手法のメリットとしては，言語自身をモデルに組み込んでいるため単独で実行でき，かつスクリプト言語を全く使用しないことも可能であることが挙げられる．(1)のスクリプト言語としてLuaを用いた理由は，スクリプト言

語自身がコンパクト（実行時のライブラリーが数百KB）であるため組み込みに向いていること，そして他のスクリプト言語に比べてより高速な処理を行うことが出来るからである．一方，(2)の手法のメリットとしては，スクリプト言語上で解析に必要な分だけのモデルパーツ（オブジェクト）を生成し，スクリプト言語の柔軟性を利用して，ブロックのように組み立てながらモデルの実行ができることが挙げられる．また，同じスクリプト言語上で動作するモジュール群ともメモリー上でシームレスに結合することが出来る．このような理由から，(2)の場合には，スクリプト言語として周辺モジュールが最も充実しているPythonを選択した．そのため，プリポスト処理を含めシームレスな計算環境を容易に実現することができる．ただし，(1)の場合と異なり，スクリプト言語(Python)を実行できるなど，実行環境に対する要求が伴う．

(1)の手法の例として，鳥取大学を中心に研究が進められている生態系モデルについて説明する．生態系モデルは，考慮する従属変数の選択やパラメーターの調整を伴うため，スクリプト言語の活用には最適な課題である．具体的には，Fantom3DにLuaを組み込み，生態系方程式のうち移流項と拡散項はコンパイルされたソルバーに行わせ，大きな変更が予想される生成項部分をスクリプト言語でモデル化する仕様にした．生態系変数はメモリーの許す限り自由に増やすことができ，またモデル内部のすべての変数値（水温，水深，日射等）がスクリプト言語から参照可能なため，提案されている様々な生態系モデルを組み込んで，手軽に試行錯誤ができる．(2)の手法の例としては，現在開発中の非構造格子環境流体モデル(Fantom Unstructured)がある．詳細は今後どこかで報告する予定であるが，非構造格子モデルを生物的な概念（細胞，細胞膜，神経系等）に基づいて完全オブジェクト化し，複雑になりがちな非構造格子のモデルを単純に表現している．そして，上で述べたように，すべてのオブジェクトがPython上で実体化でき，マクロ的にオブジェクトの入れ換えと組み替えを行いながら流体解析を実行できる．また，大規模計算における速度の低下を避けるために流体解析部分はC++11から正式に採用されたThreadによって存在するコアに対して等分割に処理を任せる仕様となっている．モデルの実行の際に起こる欲求，例えば，移流スキームを変えたい，拡散計算だけしたい，２次元計算をしたい，計算途中で条件を自動的に変更したい，途中で図化したい等はシナリオ（スクリプト）を修正すればすぐに実現できる．

最後に，本論文では概略のみの説明となってしまったが，発表においては詳細と具体例を含めて説明を行う予定である．

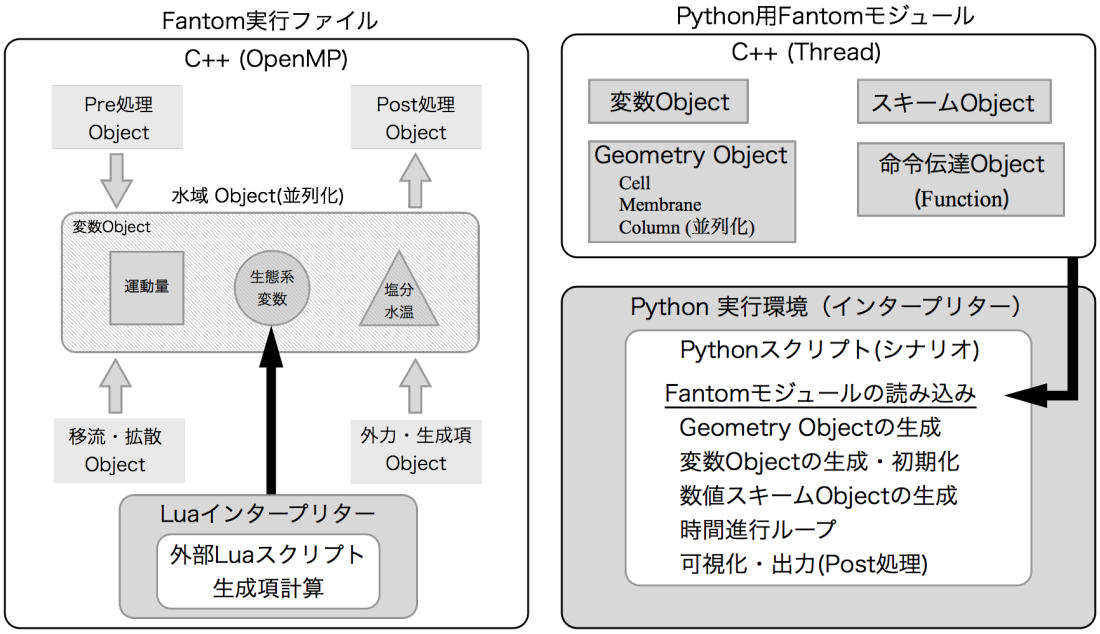

図１: 環境流体モデルにおけるスクリプト言語の利用方法
左側：(1)スクリプト言語組み込み型，右側：(2)スクリプト言語から呼び出し型

参考文献

浜中誠 (2010): スクリプト言語による効率的ゲーム開発, SoftBank Creative, 392p.