

# 教育利用報告

## 「実践的シミュレーションソフトウェア開発演習」

高橋 英男

東京大学大学院工学系研究科非常勤講師

### 1. はじめに

本稿では、大学院工学系研究科機械工学専攻向けの演習科目「実践的シミュレーションソフトウェア開発演習」についての、2020年度の内容について紹介する。本科目は2009年よりS1・S2学期（旧夏学期）に実施しているものであり、スパコン向けに並列化された物理シミュレーションプログラムを対象として、ソフトウェア工学の成果や知見を踏まえてシステムチックに開発するスキルを与えることを目的としている。本演習で利用した実行環境はOakbridge-CXである。

本科目の指導においては、シミュレーションの理論やHPC(High performance computing)の知識については大学教員が、ソフトウェアの開発プロセスや開発ツールについては、企業在籍ないし出身のスタッフが担当している。

シミュレーションの題材としては2つのテーマから受講生が選択する。一つは、分子動力学であり、もう一つは流体力学である。開発は小グループによるプロジェクト形式をとり、やるべきことを漏らさず管理するために、プロジェクト管理サービスのTrelloを利用する。プログラムの設計を検討するためにクラス図やシーケンス図などの図を用いて、全体を俯瞰しつつ設計を進める。グループ開発を実現するためにバージョン管理ツールのgitとgitlabを使う。

本年度は新型コロナウイルス感染拡大の防止のため、初のリモート講義となった。これに対応するために進め方や教材をアレンジする必要に迫られたので、その点についてまず紹介する。コロナとは独立に行った変更点についても紹介する。

本年度の科目登録者数は、例年より少ない目の8人であった。受講生の成果としては例年よりもむしろ充実したものとなった。

### 2. リモート講義に対応した変更点

リモート講義を実現するために、他の多くの科目同様にビデオ会議サービスのZoomを利用した。本科目の後半はチーム開発になるので、チーム別の作業時間においてはZoomのブレイクアウトルーム機能を活用した。科目の時間外においても、受講生同士で決めた時間でZoomは利用されており、大きな問題なく利用できた。

Zoomによるリモート講義となったことで、例年使っている演習室のiMacの代わりに、受講生個人のWindows PCやMacを使うことが必要となった。例年はソースコードの編集と、ある程度のコンパイルとデバッグまではiMacで進めておいて、並列処理がからむデバッグや性能評価をスパコン上で実施していた。それに対して、今年は学生の手元の環境がまちまちなので、ソースの編集に始まる一通りの作業をスパコンのログインノード上で実施することとした。スパコンに特有のssh(secure shell)経由のリモート作業は煩雑なので、これは受講生に負担を掛けるかと心配したが、結果的には杞憂であった。開発ツールのIDE(Integrated Development Environment)

のいくつかで、近年リモート開発のサポートが進んでおり、ローカルマシンでの開発に近い感覚でリモート開発ができるようになってきている。便利な IDE の存在は学生の間でも広く知られているようで、講師から紹介した訳ではないのに、全員がマイクロソフト社の無償 IDE である Visual Studio Code と、そのリモート開発用プラグインをインストールして使っていた。

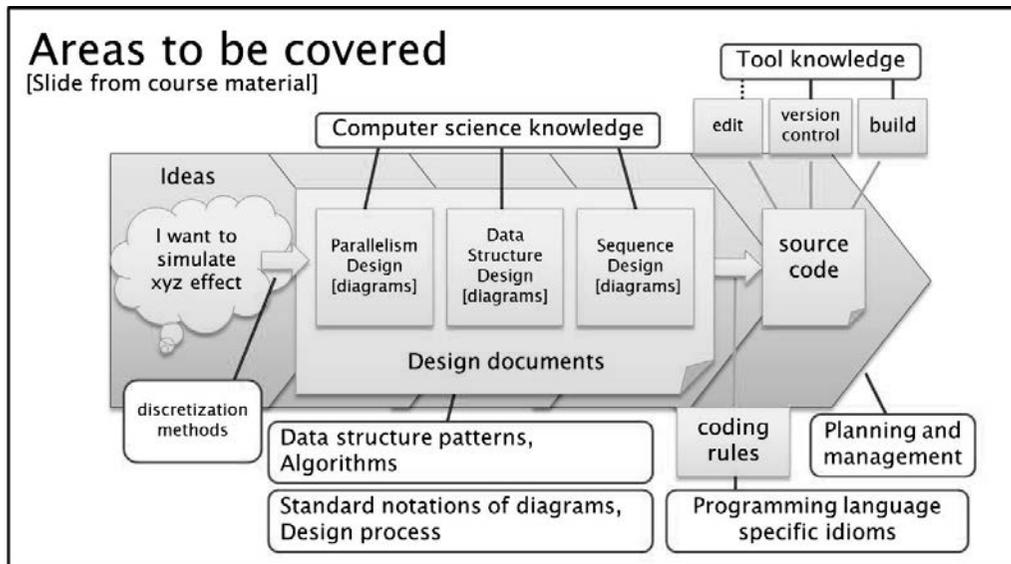


図1 本科目で扱う領域

本科目ではアイデアから始まってソースコードに至る過程で作成する図表や、プログラム開発のプロセスやツールも教えているため資料が多く、講義時間の占める割合も高くなっている。

### 3. リモート講義とは独立な変更点

本科目は教えようとしている内容が多く、演習科目としては講義時間の占める割合が高い。図1に、本科目で扱うスキルの領域を示す。この中でも、図の右上の角に示す” Tool knowledge”に関する事項は教材を整備すれば受講生のペースで自習が可能になると以前から考えており、前年度の報告においても課題認識として述べたが<sup>i</sup>、今年度は具体的に以下の取り組みをした。

まず、linuxの初歩的なコマンド類についての早見表を web 上で用意した<sup>ii</sup>。これは、情報基盤センターが教員向けに提供されている web サーバに本科目用のディレクトリを作成いただいて掲載したものである。本資料はすぐに必要となるコマンドの使用例を示し、疑問が湧いたら少し踏み込んだことも調べられるコンテンツとなることを目指している。

次に、UI が複雑で早見表を簡単には書けない2つのツール向けに、動画によるチュートリアルを作成した。2つのツールとは、先ほどの Visual Studio Code と、作図ツールであるチェンジビジョン社の astah UML である。動画は YouTube の非公開動画<sup>iii</sup>として登録しており、URL は授業内で紹介した。どちらのツールも、ツール本来の能力が発揮される正しい使い方の説明を心掛けたところ、各々1時間程度の長い動画となった。同等な内容の説明を図入りのドキュメントとして用意しようとする、大量のスクリーンショットを撮らねばならず、動画作成よりも多くの時間がかかる。動画作成は資料を用意する側としては負担が少ない方法だと感じた。

早見表と動画のそれぞれの効果については、今年度はまだ評価できていない。来年度はアン

ケートなどで受講生からのフィードバックを得たいと思う。

#### 4. 既存コンテンツの活用促進

既存の教材や資料のうちで、受講生が見落としがちになることが分かってきた事項に対して、目を向けてもらう試みを3つほど行った。

一つ目は、OBCX の利用手引書である。本科目は配布資料がとても多いこともあり、手引書を紹介しても受講生にあまり読まれない状況にあった。これを講義の中で度々参照するようにした。例えばジョブスクリプトの内容に関する質問が出た場合に、手引書の目次を開いて該当箇所を辿って、質問された事項を説明している箇所を示して説明するようにした。これにより、受講生が手引書の利用価値をよく認識し、活用につながったと考える。

二つ目は本科目の基礎演習に関するものである。本科目では物理シミュレーションプログラムに取り組む前段階として、大きな行列同士の行列積を求める並列処理プログラムを作成する。並列化の手段として MPI と OpenMP の双方を講義では説明しており、それぞれを用いた並列化を出題するのだが、例年 MPI の方のみ実装して提出する学生がほとんどであった。基礎演習は出題から締め切りまでに数週間の時間が与えられるので、時間不足なのではなく見落とされていると考え、今年は締め切り前の週に出題範囲について強調するアナウンスをした。その結果、ほとんどの学生が MPI と OpenMP 双方による解を提出した。

三つ目は実際に物理シミュレーションに取り組む応用演習に関してのものである。応用演習の最終的な成果物は、小グループごとに開発したシミュレーションプログラムについてのプレゼンである。プレゼンでは開発プロジェクトの評価、プログラムの出来栄の評価、シミュレーション結果の評価について発表を求める。例年、シミュレーションをなんとか動かすところまでで時間切れになるグループが多い。この状況を改善すべく、今年は評価のための時間が残るように、プログラムを動作させる締め切りを1週間早めることにした。その結果、最終日のプレゼンにおいては、3グループ中2グループがシミュレーション結果の評価まで含んだ発表をした。

#### 5. おわりに

本科目の主要教材は書籍として出版している<sup>i)</sup>こともあり、web サイトでの教材公開には踏み出していなかった。本科目開設当初の暗黙の理解では、linux のシェルやコマンド類などの既存ツールの説明は各自調べてもらうものとしていたが、実際には初歩的な操作で躓く学生が跡を絶たなかった。「早見表みたいなものはないのですか」と要望されたこともあった。linux コマンドの入門書を紹介したこともあるが買われることは稀である。そこでようやく、科目で使う範囲の機能については科目自身でリファレンスを提供するか、外部の書籍や資料を具体的に指定する必要があると考えるに至り、今年は教材 web サイトの開設に踏み切った。作るからには、他の科目や研究現場にも知られ、活用されるものを目指したい。

<sup>i)</sup> <https://www.cc.u-tokyo.ac.jp/public/VOL21/No5/15.lec201909-satof.pdf>

<sup>ii)</sup> <https://lecture.ecc.u-tokyo.ac.jp/~hideo-t/>

<sup>iii)</sup> URL を直接指定すれば誰でも見られるが、検索やレコメンデーションには出てこない動画

<sup>iv)</sup> <http://www.utp.or.jp/book/b306662.html> <http://www.utp.or.jp/book/b307123.html>