

大学院生のための情報教育 – シミュレーション教育の試み

中村純¹⁾, 匹田篤²⁾, 稲垣知宏¹⁾

1) 広島大学情報メディア教育研究センター

2) 広島大学総合科学研究科

nakamura@riise.hiroshima-u.ac.jp

概要： シミュレーションは現実の世界の本質的な要素とその関係をモデル化してその動向を調べるものであり、コンピュータの高速化に伴い強力な手法となってきた。文系理系を問わず、将来社会の一線で活躍する大学院生は身に付けておくべきリテラシーである。このカリキュラムはプログラミング教育ではなく、いろいろな事例についてモデル化し、何が重要な変数であるかを学ぶことを目標としている。我々が実際にこれまでにに行い改善してきた教育の実践事例を報告する。

1 はじめに

我々は広島大学総合科学研究科の博士課程前期において、シミュレーションを中心とした授業を行ってきた。総合科学研究科は文理融合を目指した大学院で、情報、物理、化学、生物などの理系分野と文化人類学、歴史、語学、哲学などの文系分野の学生が学んでいる。本コースもどの専攻の学生も受講することができる。

世界のモデルをコンピュータの中に作り、その挙動を調べて行くコンピュータシミュレーションは、情報系の学生のみならず、理系文系全ての学生に研究の重要な道具として学んで欲しい技術である。またシミュレーションを考えることで、理系文系を問わず、論理的な思考力を高めるトレーニングにもなる。

しかし、コンピュータでの研究というとプログラミングや微分方程式を思い浮かべ、多くの学生は敬遠する。このコースでは、まず世界のモデル化からスタートし、最後は一人一人が選んだテーマについて「仕様書」を書くことで終わる。第2章でその具体的な内容を報告する。

2 カリキュラム

2.1 世界のモデル化

コンピュータシミュレーションに限らず、我々の世界の記述はすべて「モデル」を構成することで成立している。情報を表現するためのビットが有限個しかないコンピュータでは真の実数は表現できず

```
float a, b, c;
```

と宣言した変数 a, b, c は実数のモデルである。ビット数が2のコンピュータでは多くの現実世界を表現するには十分とは言えないが、32ビットは符号も含めて多くの日常の現象を表すのに十分である。しかし、科学技術計算ではしばしばこれでは不十分となり、倍精度が導入される。これは実数の32ビットモデルが世界を記述するには不十分であり、より精密化を行ったことになる。



図1 授業風景 (地図の説明1)

ニュートン物理学は、太陽系を記述するのに、まず太陽と惑星のそれぞれの大きさを無視して質量だけを考える。そしてそれらの間の力を導入する。これによって太陽の回りで楕円軌道を描く惑星の配置が説明され、ニュートン以前に膨大な観測から導きだされたケプラーの法則が明快に説明される。すなわち、太陽、惑星を質量を持ち大きさを無視するモデルは、太陽系のモデルとしてその本質を抽出したものになっている。このときに、地球の表面の不規則な形状を始めから考慮することは、世界の良いモデル化とは言えない。

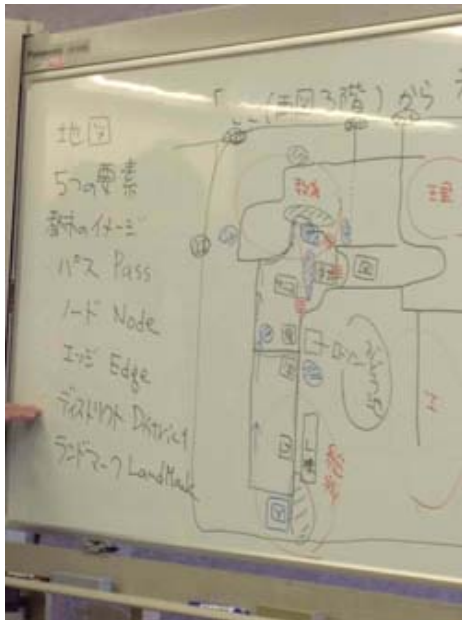


図2 授業風景 (地図の説明2)

ここで、学生たちに紙を渡して地図を描いてもらう。全員が知っているキャンパスの食堂に授業の場所から到達するための地図とする。そこにある全てを描く学生はいないし、またそれは不可能である。そもそも建物などは四角などで描かれており、それは実際のものとは大きく異なる。

学生たちが描いた地図を集めて皆で鑑賞する。なぜこのような簡単なものでも地図として十分に役に立つのかを質問し、これが世界のモデル化になっていることに気がついてもらう。

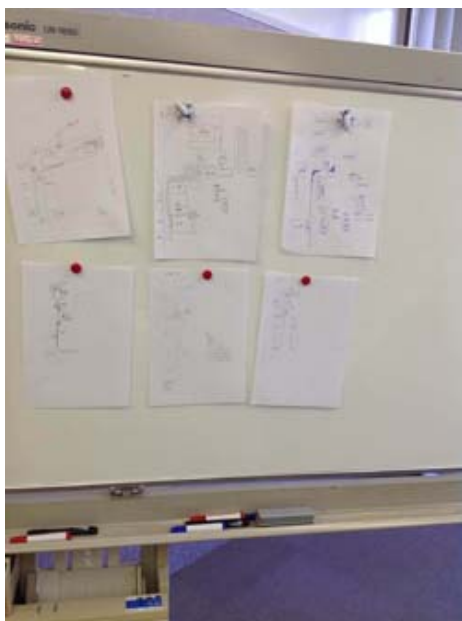


図3 学生が描いた地図を皆で見る

2.2 シミュレーションの基礎

方程式をあらわに出すことはせず、「時間が経つとも (物体の位置, 環境, 性質など) がどのように変わっていくかを考えよう」と問題設定をする。これは、時間発展を微分ではなく差分方程式で書いたものと同様であり、そのことに気がつく学生もいるが特に誘導はしない。例えば、生物系の学生にとって、来年の山の木の状態を今年の状態から推測して次々と年ごとにどうなるかをシミュレートしてみようというのは自然な発想であるが、微分方程式を差分にしてこの t は十分に小さいというような入り方をするとこのような考えかたを制限してしまうようなことが起こりかねない。まずは

$$(\text{未来の状態}) = (\text{今年の状態}) + A$$

というような例を考え、これを実現するにはどのような計算が必要で、それをコンピュータに計算させるためにはどうすればいいかをフローチャートなどを使って皆で考えていく。次に、もしこれが例えばボールの位置で、時間の発展を追うのであれば、

$$(\text{時刻 } t' \text{ の状態}) = (\text{時刻 } t \text{ の状態}) + A$$

という形になり、 t' が t の時は A はゼロでなければならないので、 $A = (t' - t) * B$ という形になることを納得してもらおう。微分について知識が十分にある学生がいるときは、これが差分方程式となることを示す。

さらに、ローレンツモデルを示し、これをプログラム化して結果を gnuplot で見てもらう。ここでは簡単な形の法則からでもカオスが出ること、また可視化が非常に強力であることを体験してもらう。

2.2.1 プログラミング

前節の議論を進めて行く時には、実際にプログラムを書いてもらう。この授業の受講はプログラミングの知識を要求していない。まず、プログラミングの要素として

- 代入。 $A=B$ は B を A に代入することを表す
- 繰り返し。繰り返しの機能がなければ、100万回の計算のためには100万行の記述が必要になる。
- 条件分岐。条件によってプログラムが振舞いを変えられなければ、同じ計算しかできない。
- 入出力

があることを解説し、また逐次型プログラミングではプログラムの記載順に実行が行われることを説明する¹

このコースでは使用言語として Octave[1] を採用した。Octave は MatLab とほぼ同等の機能を持ち、Windows, Mac, Linux 上でフリーで利用可能であるので、学生が自分パソコンで練習し、自分の課題を実行することもできる。また予備知識が無くても簡単なプログラムがすぐ書け、本格的なプログラム開発にも耐える。

上記のようなプログラミングの説明をしたあと、非常に簡単な穴埋め問題をやってもらう。

空白に正しいものを入れなさい

1. $1+3$ と入力すると $ans=(\quad)$ となる。
2. $x=1$ と入力すると $x=(\quad)$ と出力される。
3. $x=1$; と入力すると何も出力されないが x には 1 が代入されている。 $y=2$; $z=x+y$ と入力すると $z=(\quad)$ となる
4. $z=x(\quad)y$ と入力すると $z=-1$ となる。
5. かけ算は* (アスタリスク) 記号で表す。 $x*y$ と入力すると $ans=(\quad)$ となる。
6. 割り算は / 記号で表す。 x/y と入力すると $ans=(\quad)$ となる。

以下同様に、繰り返し、条件分岐、入出力を学ぶ。LMS を利用して何度でも繰り返し受験でき、直ちに採点されてどの問題が正しくなかったかが表示される²。ほとんどの学生が1回ないし2回で満点となる。これは文献 [2] で使われた手法である。

次に説明しながら簡単なプログラムを書いていき、次にそれを自分たちで書いてみる。このような「写経型」のプログラミング学習は、特に文系の学生には効果があることが報告されている [3]。

2.2.2 仕様書の作成

この授業の最終課題は、自分が考えた問題に対してそのプログラムを外注したときに制作すべき仕様書を書きなさいというものである。問題の提案から最初の仕様書案の作成、それに対する教員からの不備の指摘、改善案と数週間をかけて作成する。問題、仕様書案の提示は全学生の前で行う。

卒業後に専門職について問題解決のためにシミュレーションを実施する場合には、プログラマー

として対応する可能性よりは仕様書を書く立場になることの方が多いと思われる。しかし仕様書を書く課題は、そのような理由ばかりではなく、仕様書を書くことで何がインプットであり、何がアウトプットして求められ、どのようなアルゴリズムを使用し、何をパラメータとして動かし、どのような既存のデータを利用すべきかということを明確に指示しなければならない。もちろん、それを他人に分かるように明確に文章にする力も求められる。

3 まとめ

研究の場のみならず、シミュレーションは社会の多くの場所でもっと有効な方法論となるべきである。しかし、そのためには、文理の別なくシミュレーションについて理解している学生たちを育てる必要がある。本稿では、総合科学研究科といういろいろな学生が学ぶ場所で数年に渡って行ってきたシミュレーション教育について報告を行った。

参考文献

- [1] オクターブ
http://ja.wikipedia.org/wiki/GNU_Octave
- [2] JIS FORTRAN 入門 (上) (下) 第3版東京大学出版会 (1984)
- [3] 喜多一, 岡本雅子, 藤岡健史, 吉川直人: 写経型学習による C 言語プログラミングワークブック, 共立出版 (2012).

¹本カリキュラムでは並列プログラミングは取り扱わない

²WebCT (BbLS) CE6 のアセスメント機能