

本書の目的

私にとっては、バーチャルから現実の地上界に続くひとすじのクモの糸。自分の能力と技術を学術的に売り込むことを最大の目的にしています。しかしながら、同時にそれは応用計量経済学とその周辺分野の学習体系のパラダイムシフトを広く世に問うものとしています。自分も変わらないといけませんが、それを理解してもらう世界そのものも変化をしてもらわないと、私が現実の世界で生きる余地はないからです。

ただし、クモの糸をトリトンの貝に通すのには、ソースをすべてリアルタイムで公開するなど非常に危険なことをしているわけで、狭隘な精神や金銭的野望をもった人物や団体によって「ただ取り」されてはけません。そういうミノスの王的ふるまいをするものには、あらゆる手段を用いて煮え湯を浴びせることになります。

著作権について

英文で Copyright としてメインページに示したとおりですが、日本語で若干補足するとともに、若干の追加を示します。

What is allowed and what is legally prohibited

Legally Prohibited

- 1) 学術用であろうとなかろうと、ほかのソフトウェア開発に本書の内容を使用して新しいコマンドを作成して加えること。間接か直接かは関係ありません。
- 2) 民間、半民間の金融関係の研究所が本書を利用してソフトウェアとして一部の技術を利用して間接的に利益をあげること。
- 3) 本書の内容を **Compile** してソースをわからなくしてしまうこと。

Allowed

- 1) 学術論文や投稿記事に本書の一部または大部分のアイデアを使うこと。
- 2) 大学院などの授業に本書の内容の一部または大部分利用すること。
- 3) 民間の方も含む、研究者および学生が、技能レベルアップに利用すること。
- 4) GAUSS 内部で **compile** しないのであれば、本書の **program** の改造修正版の公開は認めます。ただし WEB 上に公開してください。
- 5) 大学学内や研究者学生間のプリントアウトの回覧等は各自自由に。ただし、誤植やミスをごまめに修正していますので、最新のものを WEB から見ることを推奨します。

要約すると

使用者や読者が民間か教育機関の人間か、はたまたそれ以外の方は問いませんが、「ソフトウェア開発またはそのコマンド開発」以外の学術教育目的や自らのスキルや関連分野の知識増進に御使用ください。現在、公的な立場での発言権がないため、引用などを強制するものではありませんが、本書は準学術ソースまたはそれを越えるものと考えてください。

学術活動や著述活動への利用には一切文句は差し挟む考えはありませんが、最低限「ソフトウェア開発に利用しない」という絶対条件を守ってください。これを守ってもらわないと、私は両親の郷里の英雄、第二の平賀源内になることになります。

勝手に特許にしないこと

本書の内容は、これまでの統計のプログラムブックに載っていたような内容ではない分野や内容を数限りなく扱っています。もし、これを盗んだり借用改良して何かの物理的な装置と組み合わせてドメスチックな特許申請をすれば、おそらく受理されるでしょう。そのような悪行をすれば、どのような手段を使おうと死ぬまで闘います。最後に、すでに、またこれから金融工学の核心に近づいていきますが、申請受理者の誤解に基づいた、あるいは政治的な裁量に基づいてもたらせた、いかなるローカルな特許やビジネスモデルやソフトウェア特許も基本的な著作権には勝つことはできないことを指摘しておきます。私見としては、確率や統計が絡んでくる分野は、多くの先哲が学問的積み上げた実績のうえに存在しているのであって、特許やソフトウェア特許の網をかけるのは誤りと強く考えます。私自身も、著作権以上のことを主張するつもりはありません。

著作権の保護を

GAUSS を知っている特にアメリカ帰りの研究者は、当地での教育方法から GAUSS とはほかの人の Procedure の一部を”Change here”として変更するものだとして体得してしまっている人がほとんどです。プログラムは他人のを丸写しで、暗記力で優秀な成績で日本へ帰られた人に限って、「私は GAUSS ができます」と言う人がいます。そういう方々は Change Here 学派であって、指導教官から譲り受けた program や Web 上に公開されている Program 以外は書けないわけです。そういう人々に限って、実力と学位とのギャップから盗作贋作の incentive が働くことになります。本書は Web 上に公開されていない多くの入門から中級までの計量経済学の内容を掲載しています。Web 上に見当たらなかったり、同じトピックであってもまったくアルゴリズムが異なるものは、100%私の自作であると考えて間違いありません。著作権尊重を。

自由な回覧配布を前提としていて公共財として使っていただいて結構なのですが、著作権はこの私にあります。さも自分が考え出したように同じような文章を書いて出版したり、官公庁の研修マニュアルに作り変えたりすることのないようにしてください。また、日本語が読める内外の研究者（特に、韓国、台湾などの「優秀な」人たち）は、本書を丸写ししたりアイデアを盗んで、母国で母国語で同じような書物を書くことはやめてください。GAUSS 以外の商業ソフト開発に本書を参考にしたり、WEB 上の他の GAUSS の巨大な code 群を利用することは重大な著作権侵害にあたります。特に金融関係の方々、お気をつけください。GAUSS のアーカイブにアクセスしていると記録は残るもので、金融ソフトがなにかを開発してから数年後に莫大なロイヤリティーや取得利益の請求を（Aptech ではな

くアルゴリズムの作者から)受け取るはめになります。GAUSS のコードは、明文化されていようといまいと、GAUSS という土俵の上で学術上共有される制約の下でフリーなのであって、他のソフトに移植したり商用に利用することは、著作権侵害以上に、天文学的なロイヤリティー支払いや賠償の範囲に入ってくるので、日本で許されてもアメリカでは許されないことはありますので細心の注意をお払い下さい。

ただし、本書は一部、私のアメリカで受けた教育における諸先生の講義その他の著作権(宿題として課題が与えられて、それを私個人またはスタディーグループで解いたプログラム類)では侵すことがあるかもしれませんが、この点については、本書が印刷物になればその際に、私とその方々の間で調整して、個人的に許可を得て解決するものとします。一時的にその旨のことを英語で明示しています。全体のほんのわずかな部分にすぎませんが。なお、シュミレーションによって得られてジャーナルに公開されている Critical Value や統計量には、厳然たる著作権があるという観点から、プログラムには取り入れていません。少々不便かもしれませんが、統計量の結果は、御自分でジャーナル等と対照判断してください。ただし、1990 年前半くらいまでの技術なら本書の読者はシュミレーションで御自分のテーブルを作れるはずです。

著作権侵害免除者

書物を提供してくれたり、訂正上有用な質問をしてくれたりしている国内の研究者(大学院生を含む)が私の文章やプログラムを「発展させる形で」何かにすることを認めます。また、「金融工学ゼミナール」においてリンクを張って頂いているその管理者の方も同様です。世話になっている方々に著作権上のクレームをつけることは信義にもとることになるので多少のことは目をつむります。

コンセプト

My Mission に示しましたように、できるかぎりのところを微分によらない Gird Search により数値計算を行ない、procedure を積み上げる方法で、GAUSS システムのバージョンによらない基本関数だけを用いてプログラムを組み立てていきます。また、Excel 的になっている思考パターンの打破に努めます。表計算的思考の延長上には、繰り返し計算などの学術論文レベルの展開は期待できません。逐次計算の真髄を GAUSS によってお見せします。同時に、そのアルゴリズムが速い遅いは気にせずに、現実の思考や論理構造を重視して、くどいほど、またわかりやすいプログラムに努めます。展開すべきところは n の何乗になろうと展開させます。メモリ上や計算上の効率ではなくて、思考上のわかりやすさやその効率性をめざします。

本書の対象とする人たち

- 1) 経済学およびファイナンスの分野の学部上級および大学院生
- 2) 論文や教育方法にいき詰まりをみせる中堅研究者層
- 3) 経済学や統計学などの理論で道を極めた長老たち

現在のところ、(1)の範疇の支持しか得られていませんが(質問者や書物の提供者からそう想像ができる)、(2)や(3)の層にも受け入れられるような体裁や正確さを徐々に目指したい。特に、GAUSS その他のプログラミング言語は、昔のような OS 上で複雑なファイル操作を前段階で行ない、その上でコンパイルして実行するなどというような学問とは全く関係のないことをやる必要が全くなくなってきています。経済学周辺分野の道を極めた方々に、理解をしていただきたい。この層の支持者を一人でも増やすことが最大の目的です。実際のところ、GAUSS の Tutorial のある Nerlove 教授や、Stata の創始者 Finis Welch 教授たちは、とんでもないシニアであります。年齢は関係ないものと思われます。もし、一連の操作がわけのわからないものであるとするのならば、それは新しい世代の無知や怠慢から生じた非効率性と言えましょう。長老たちのこの分野の参加を大いに求めます。

GAUSS 他、用途別の組み合わせについて

基本的に、本書は GAUSS を素手で扱うことに専念して、制限つき最大化の収束計算にのみ CML を使うスタイルをとっています。また、プログラムの可能な限り大部分を GAUSS Light だけで動作可能なワークスペースを使ったものにしています。以下、実際に GAUSS を実践的に導入するにあたっての私なりのアドバイスを示しておきます。

イ) とりあえず使ってみたい人、ゼミ等でプログラム実習をしたい

GAUSS Light

ロ) 本書と同じく素手で 1 からプログラムしたい人

GAUSS + CML + Maxlik

ハ) 海外の論文付属のプログラムを動かしたい人

GAUSS + CO + Optimization + CML + Maxlik

二) 金融工学 + Wavelet に利用したい

GAUSS + Optimization の上に TSM

ホ) 金融工学 + 時系列に利用したい

GAUSS(+ CML) + FANPAC + Time Series(+ Coint)

ヘ) 時系列に特化した分析につかいたい(GAUSSXver.5 は Census X12 を含む)

GAUSS(+ CML) + Time Series + Coint + GAUSSX5

ト) TSP の手法が手に染み付いて忘れられないが GAUSS も使いたい

GAUSS + GAUSSX

チ) イギリス大好き、C++ ライクにプログラムしたい、

OX

すなわち、CML,Maxlik,CO,Optimaization は目的関数の最適化を行なうもので、その他のライブラリやアドオンは、アプリケーションのようなことをするものです。金融工学をパッケージ的にやろうとするならば、TSM は若干昔の手法でプログラムが書かれているので扱いづらいかもしれません。そうではなくて最新の GAUSS に FANPAC と GAUSSX5 を搭載して分析をすれば、RATS や TSP の代替以上の力を発揮するものと思われます。

GAUSS に向いていないこと

A) 微分や積分、計算展開など記号的に Reduce ライクに代数計算をすること

Matlab(シンボリック Toolbox) または Mathematica

B) 計量経済学ではなくて、より厳密な統計学の計算が関数として必要

S-Plus または R

C) CPS の C D などの大規模データをそのまま動かしたい、表計算のように簡単

Stata

GAUSS に移ってくるべき人たち

Fortran で時系列計算をしている人
金融工学をしたいが S-Plus では荷が重い人
自分の学生しか使わないソフトを作っている統計学者
将来経済の分野のアメリカ留学を考えている人

考えうる最小限のベストなソフトウェア構成は

GAUSS + CML + Maxlik + CO + Optimization

大学院のゼミや若手研究者のグループ学習等ために Legal な GAUSS Light の入手方法

1) 本体とオフィシャルマニュアルのみ取得(現在 ver. 5)

FTP アクセス (詳しくは本書 Reference.pdf を参考のこと)

2) GAUSS Light ver.3.6 + GPE 2 パッケージをその解説本と購入(学部上級 ~ MA 向け)

Kuan-Pin Lin “Computational Econometrics” 79 ドル

3) GAUSS Light ver.3.2 + EF Library + Ruud マニュアル + ベイジアンの上級教科書

Mittelhammer 他 “Econometric Foundations” 70 ドル

2) および 3) については、amazon.com の A Reader のコメント参照のこと (長い方の英文の A Reader の変な英語の文章は私が書いたもの) どちらもパッケージライブラリーにたよっているがプログラム自体は正統的な GAUSS の記述法で書かれている。3) は教科書と CD のソフトとの関連性は薄い。内容的にはかなり高度。EF ライブラリは判読可能。

バージョンと OS について

- 1) 本書は一貫して Windows 上の GAUSS を対象に、そのなるべく多くを Free で公開されている Light 版でプログラムできるよう努めます。Unix 上のことは扱いません。
- 2) アメリカでもまだ、3.2 から 3.6 までのバージョンが実働していると聞いています。また、WEB 上の膨大なライブラリー群も 3.2 が一応のベースになっています。本書は、3.2 以上ならすべてのバージョンで動くようなプログラムに努めます。もちろん、最新 5 のバージョンでも動くようにします。
- 3) ただし、乱数を発生させる組込み関数のなかで、その内部に引数としてシードを入れる rndns と rndus は ver.4.0 で廃止されたので、「rndseed 数字;」の形で外部でシードを設定してから rndn や rndu を使うようにしてください。唯一、重要な変更点です。
- 4) ライブラリの説明では古い MAXLIK は 4.0、CML は 1.0 を使っているのですが、最新版とは拡張された部分で多少の違いはあります。私の解析では、GPE2 のほとんどは、3.2 版 GAUSS とその時代の MAXLIK と CML ベースにコンパイルされたパッケージライブラリーです。おそらく、本書の環境と大差はないと推察されます。

GAUSS について

- 1) GAUSS は 3.2 以上の「Windows のバージョン」であれば、基本的に、同じような機能を持っています。また根本になるプログラムの構造やコマンドが、バージョンごとにそう変化してはいません。3.2FULL と 3.6Light および 4.0Light の 3 つで動くことを確認しています。Ver.5 でも動くかどうか確認作業を進める予定ですが、マニュアルを通読した限りでは、問題なく動くものと思われます。
- 2) Light 版と Full 版の主な違いは、 100×100 のデータ行列制限があることと若干のワークスペース上の容量の制限があるだけです。使用期限が限られているとか、パフォーマンスが異なるとかいう違いは認められません。なるべく Light 版で動くような説明にするように努力します。少なくとも、シミュレーションには、これで十分です。
- 3) GPE2 は GAUSS のオープンソースの原則に則っていないことと、プログラムの根幹のところで目的関数の 2 値ある引数の左右をひっくり返したり、コンスタントタームを独立変数の最後におくようにデザインされていたりするのでお勧めはできません。しかしながら、GPE2 で計算した結果とライブラリで計算した結果を比べて確かめているとことが本書にはかなりの箇所があります。ただし、Prof.Lin はそれ以外は、非常に標準的でわかりやすい文法で書いておられます。
- 4) UNIX 版はメモリ管理が難しい場合が確認されています。その点、WINDOWS 版はバーチャルメモリを使っても問題なくメモリを自動増加します。Full 版では、マシンの使用可能なメモリに応じて扱えるデータのディメンションは決まってきます。システム管理者を養成したりハッカーを養成したりするのでなければ、UNIX 上の授業および共同研究は、TA や研究補助者の「無用な」負担と時間的な労力を増すだけです。
- 5) Matlab に移植するために perl を学んだり、個人商店ソフトである無料の OX 上で GAUSS のプログラムを実行するために proc の順番を変更したりすることは、非常に非常におろかなことです。賢明な人はそのようなことをしようとも思わないはずです。そのように、英語を旅行用翻訳機で訳させるようなことが一部の日本の大学院で行なわれているとすれば驚きのなにものでもありません。GAUSS 本体を学びましょう。

PDF 形式について

- 1) 最もセキュリティの高い部類のフォーマットである。また、既存のロボット登録のサーチエンジンに内容をテキストとして保存されない。(しかしながら、その反対に、自動登録は拒否される。ただし、最近は PDF を一部自動でテキストに直して登録する技術も開発されたようです。) 無責任な自動翻訳も受け付けない。
- 2) 標準フォントがそのコンピューターに存在しない場合には文字化けや行崩れが起きることがある。たとえば、WordPerfect のフォントなどを使った文章を PDF 形式に変換すると、WordPerfect のソフトウェアを組み込んでいない他人には、その部分が見えな

いか、文字化けをする。本書は MSWORD の標準フォントだけを使用しています。

- 3) MathType などのアドオンソフトウェアを組み込んだ文書を変換して PDF にした場合も、他人には意図したように見えないことがある。この点を考慮して、不自由ではあるが、本書はアドオンは使わず、MSWORD 内 Object の数式 3.0 を使用しています。
- 4) フォント構成やシステムによっては、行列等の縦の線が若干太くなったり細くなったりして見える不具合がある。この問題は、本書でも一部、確認できる。
- 5) 文章の容量が大きくなった場合や、連続して複数の PDF 形式の文章を見る際に、Acrobat Reader では、メモリ管理に不行き届きがあって、ページが抜けたり、一部のグラフが欠損して見える。(このことは、1 文章ごとに Acrobat Reader を終了させるか、もしくは、一括ダウンロードしてしまって、Ebook Reader で見ると、この問題は解決される。) Ebook Reader で FONT の見え方を変更して見ることを推奨します。
- 6) GAUSS のグラフは PS 形式でも出力できますが、編集上の簡便さと容量上小さくしたために、多少雑に見えますが、そのままカットアンドペーストしたグラフを PDF に張りつけています。実際にはもっと鮮明に見えます。不都合はお許してください。
- 7) 印刷できないようにするなどの設定もできます。しかしながら、本書は万人に読んでもらうために、印刷ロックはかけてはいません。ご自由に印刷してお読みください。

Disclaimer)

ジュニア版および本論は、PDF 形式にして、視覚非健常者への配慮が欠けることをしておりますが、テキスト化または Word 形式での公開は、各国語への無責任な自動翻訳を受けつけてしまうという両刃の剣的性格を帯びているところであり、テキストでの公開は金融工学関係の教科書準拠プログラムのところの公開だけでお許してください。

本書の構成および内容について

- 1) 著者は理論計量や統計が専門でなく、非金融系の実物経済の応用分析を学んだので、細かい議論や不確かな箇所が多少あることはお許し願いたい。ただし、大学院の *Econometrics* が I から V までであるところで学んだのと、数校の大学院をハシゴして武者修業をしたので、たいいていの基本的事項については理解しています。
- 2) 計量経済学の立ち上がった「いわゆる」理論的はことは扱いません。しかしながら、最新の技術も含めて、計量や経済を専門にしない人にも利用されていくであろう実践的なトピックスを GAUSS でプログラムしていきます。また、本書を計量経済学をすべてカバーする「教科書」にするつもりは毛頭ありません。
- 3) 私は GAUSS 本体についてはよく知っています。他の人が難しく説明する内容を簡単に短いプログラムを積み重ねる形で説明していきます。また、一番アクセスフレンドリーな WINDOWS の上で説明します。(UNIX の上などで、UNIX を知らない人にコマンドの羅列を覚えさせて、なにをやっているのかわからないまま他人のプログラムをコピーしたり、少しだけ改造しただけでは何も生まれません。) 解けないような難題を与えてとりくませたり、例題を与えてコピーされるタイプの学習法はプログラム技術習得には不向きです。本書で扱っているプログラムが短いのは、私が短くエッセンスを伝えるためにそうしているのであって、GAUSS が短いプログラムをするものであるというわけではありませんので、誤解のないようにしてください。
- 4) コマンドを使ったような「操作」を本書は扱いません。Stata や TSP のような 1 行コマンドの集まりをけして「プログラムする」とはいいません。単なる「操作」です。行列形式で統計計量経済学を知っている人が、教科書の公式や数式結果を見てすぐに自分の頭で考えてプログラムできるような素地を作るように説明していきます。ただし、行列形式で書かれている教科書を使う必要があります。
- 5) Fortran から移ってきた人がよくやる、ループを goto 文で回したり、サブルーチンを多用したりするような、プログラム上の悪弊を、本書は、取り除くように努めます。また、読者は、エラー処理やリターン処理でもないのに、goto 文で回したりループから抜けたりするプログラムや、独立サブルーチンを return で終わる書き方をしたプログラムを見かけたなら、その人がその道の大家であろうと、GAUSS の練習プログラムを公開していても、実は GAUSS をよくわかっていない人物であると思って間違いありません。本書では GAUSS の真髄、procedure プログラミングを伝授します。
- 6) Fortran と GAUSS の違いは、GAUSS はすでに行列になったものを列単位あるいは行単位に動かしてプログラムできること(したがって、日本的な を使った教科書ではなくて、行列計算で組み立てられた教科書が必要になります)そして、行単位の命令ではなくて原始的な C のような書き方をするもの(トップレベルの main の部分はオープンで、しかも C の { } に相当する部分と型宣言が省略された形)と言えます。これをもう一度 C に戻そうとする OX の策謀は邪悪です。GAUSS の GAUSS たる由縁

の1つには、手続きを極めて簡素化してプログラムのアルゴリズム本体そのもので勝負することにあります。

- 7) こまめにバージョンアップしていきます。アップデートの日付と時間、それとバージョンで判断して、最新のものを手にダウンロードして読んで下さい。細かい間違いは気にしないでアップしていますので、正常に動作するプログラムはバージョンが低いものについては確認されていないと理解してください。
- 8) Update Log のファイルに更新履歴を記述しています。現在の無料プロバイダーでは、ルールが厳しく、ディレクトリーを丸見えにすることは禁止されているので、擬似的に、ディレクトリーの画面をコピーしてそれをホームページにしています。日付や時間が変わっていても、また、容量が小さいのにもかかわらず、その章がほぼ完成していたりするのはそのためです。

私は、Web デザイナーではありません。機敏なアクセスを妨げるような表紙やグラフィックスを付け加えようとは思いません。また、プロバイダーサイドや広告で設置管理されている CGI や Java などの監視機能以外に、私自身が設置している CGI や Java 関数は Web 上にはありません。当方自身は、弓削達もニッコリ CGI や Java 監視フリーです。

Myth

- 1) **OX (牛? オックスフォード?) が GAUSS よりも速いのではないですか?**

そうすることは、「もう」過去のことになっています。コンピューターの世界では数年前のことは(日本の研究者のなかで伝承ゲームでコンセンサスになっているころには遠い昔のことになっています。)特に有名な OX と GAUSS の速度を比べた論文のコードは GAUSS のプログラマーとは決して言えない方法でコーディングされているので、真似をすることはけっしてお勧めできません。確かに、S や S-PLUS のようにべらぼうに価格の高いものに対して R が存在するのならわかりますが、本格的プログラムができてコンパイルもできる言語体系のなかで一番コストパフォーマンスのよい GAUSS のクローンを公共の大学で教えることには大きな疑念があります。クローンではなく、本体を学び、そして教えましょう。ただし R のプロジェクトは偉大であり Mac とかプラットフォームが多目でそれなりに評価できますが、現在は完全分化してますが、開発当初の GAUSS と Ox をめぐる米英の戦いは「逆独立戦争」の様相を呈しており、私個人としては醜い悪意を感じます。

- 2) **CO は数学的に条件付きの最適化問題を解くので、CML よりも数学的でパフォーマンスがよいのではないですか?**

Aptech のサイトのなかで公式に述べられているように、CMLこそが条件付きの最適化をする計量計算をするためにつくられたもので、GAUSS の中では最も技術的に芸術の域にまで達したライブラリと言えます。一方、CO はその数学版と言えます。そこにブートストラップ機能をはじめとする計量経済学のツールを付け加えて完成したのが CML です。それだ

けに、価格的にも他のライブラリに比べ差別化がなされています。CML のライブラリが GAUSS の GAUSS たるゆえんと言えましょう。(本書の読者なら、まっとうな計量経済学の知識がある方は、CML 以外のことは基本的に何でも素手でプログラムできると思います。) 本家アメリカでも、間違ったやり方で、CML ではなくわざわざ CO を使う傾向にあることは悲しい限りです。何人かの先哲が係数とデータの設定がよくわからなくて、単純な CO を使い始め、それとあわせて CML が高価なため研究施設によっては入っていないところがあるため、CML ではなくて CO を使ったプログラムが多く見られるのでしょう。

3) **GAUSS は「乱数に強い」「行列操作の」プログラミング言語ではないですか？**

確かに、ある面では正しいと言えます。「行列操作の」という点では MATLAB の方が行列として操作できます。GAUSS は行列を計量計算と同じように「列ごとに」操作する言語であると言えます。「乱数に強い」という点は、GAUSS にはほとんどの分布の乱数を簡単に作ることができます。そういう意味で乱数に強いのですが、それを利用するにはプログラムの技術を要します。本書では、その点もじっくりと解説します。2002 年の現在の状況では、シミュレーションやブートストラップするのが簡単である意味では、Finis Welch 先生の「魔法の車椅子」「牧場を生み出した黄金の靴」である STATA(ステイタ)が、パッケージ計算としては、1 歩も 2 歩も先を行っています。最近では、この手の分野で STATA は GAUSS と正面から戦う姿勢を見せ始めています。ここで、「乱数に強い」とは GAUSS がプログラミングの前提とする乱数生成に信頼性があり種類が豊富ということを意味します。また Maxlik や CML に Bootstrap 機能が組み込まれていることからそう言えます。

4) **GAUSS には WEB 上に豊富なプログラム集が存在するという意見があります。**

しかしながら、GAUSS とは、プログラムがないものを作る、または、パッケージとなっているコマンドがどのようになっているのかを再確認するという意味合いの強い言語であって、模倣してプログラム集を使うというタイプの言語ではないように思われます。アメリカン大学にアーカイブがありますが、年々衰退をしてリストからはずれる人が目立ってきています。Finance 関係で圧倒的なプログラム能力を誇った方がプライベートセクターに転出して公開プログラムを消してしまったこと、Lin 教授や OX が何でもかんでもコンパイルしてしまったこと、そして他の STATA や Eviews の開発競争にただ取りされてしまって、プログラムを新たに公開しようというインセンティブが働かなくなっているように思われます。アメリカン大学のリストに載っていないで、個人で GAUSS のプログラムを公開しているアジア系およびラテン系の方がおられますが、半数以上のプログラムが完全には動かなかったり、公開しているものはある種のベイトで故意にそのままでは動かなくさせているものも多く存在します。本書は、少なくとも既存の状態動くプログラムを提供しており、どんなテクニックで、どのようにプログラミングするかを示しています。

5) **GAUSS は、mixed logit やページアンのためだけのものではありませんか？また、GMM をするためだけのものではありませんか？**

列車に乘せられて日本に送りこまれた研究者がアメリカでやっているものと同じプログラ

ムと講義ノートをもとに、伝道師のように広めているにすぎません。おそらく、logit や probit の分野は将来的には STATA に完全に制圧されて、より信頼性のあるパッケージとなることと思われます。なぜなら、そのために STATA は誕生し発展しているからです。ただし、Structural Economics をやっておられる方々、GMM が必要になる方々にとっては、未知のコードを書くという意味で、GAUSS は重要なツールとして残っていくでしょう。著作権を侵さないという大原則から、それらの日本で「伝道」されている事柄については本書では、おもだってとりあげることはしません。ただし、GMM の大家（または将来そうなるべき方）がその分野に論文等の業績で世界的に秀でているということと、その方のプログラムが GAUSS の文法にのっとって誰にでもわかる手法で書かれているかどうかという問題はまったくの別物です。わざと難解に書かれているその方のプログラムを初心者が手本にして勉強をしようものなら、東海道線の途中の駅で永遠に立ち往生して洪水の渦のなかに埋没することになるでしょう。存在しないプログラムを書くという意味で GAUSS は有効な手段ですが、特定の分野の計量をするのに GAUSS を使うという手法は邪道です。本書では、むしろ、基本的古典的な計量経済学と論文等に付随して公開されている高度なレベルのコードとのギャップを埋めるために、1つ1つステップを踏んでテクニックを詰めていきます。

6) そもそも GAUSS は「やさしい」のですか？

これは「ラジオやさしいビジネス英語」がアメリカ帰りにしか完全には理解できない水準で、初心者にはむしろ害毒になると同じように表面上は理解したふりはできるかもしれませんが、きっちりとステップをおって理解しないと完全には体得できない言語です。私はもともと SAS と SHAZAM 系なのですが（もちろん日本人のなかではもっとも STATA を知っていると思われます）、何か別のソフトウェアを知っていれば、この言語の簡単明解さ、そして使い勝手がよりわかるでしょう。行列を列ごとに計算できるという本来の GAUSS の使い方をすれば、計量経済学や統計学の理論を順にあとづけることができ、理論の理解を深めるのに役立つと思われます。そういう意味で、今アメリカの計量経済学で GAUSS が必須になっているのだと思われます。Wavelet に関するツールボックスがあるからだとかあるマクロの教科書がそれを用いているからだとかという外生的な要因で学ばれているのではないことを付け足しておきます。Wavelet については、筆者は 90 年代初頭の MAC の上の千葉さんの Sig-Mathematica の隠れメンバーであった時代から Mathematica でしか遊んだことはないのですが、時間が残っていれば 1 節をあげて取扱いたいと思います。

特別) あなたの書いたプログラムは、ループが多くて前方参照ばかりで、非効率で遅いプログラムになっていて、よくないと思うのですが？

たとえば、ループをシークエンスで代用してプログラムを高速にする方法は存在します。10 年くらい前までは、そういうこともあらゆるプログラミング言語で注意を受けた事項です。しかしながら、十分にマシンの性能が上がった現在では、たとえ今の GAUSS や MATLAB のマニュアルにそのような記述があっても徐々に過去のことになりつつありま

す。現在は、あるいはこれから将来は、読み手にわかりやすいプログラムを書くということが重要になっています。そういう文脈で、前方参照も肯定的に受け入れられるようになってきています。10 年も以上前に、FORTRAN や Pascal を学んだ時に、先生が言っていたことを今現在も真であると信じるのは、愚かなことです。前方参照をしてでも、また、くどいほどループをきっちりと記述してまでも読み手にわかるプログラムを書くのが現在の趨勢です。読み手にやさしいプログラミング、そして、コード上で思考するプログラミングを本書はめざします。

7) あなたの本書の目指すものはいったい何ですか？新しいソフトウェアですか？

いいえ。名刀正宗を作る刀鍛冶になるつもりは毛頭ありません。新しいソフトウェアを作ったりツールボックスを作ったりするとは、それなりの意味のあることですが、私はまったく興味がありません。本書を新しいソフトウェアを作り出すという視点からはとらえないでください。あくまで GAUSS の中で勝負できる素地を身につけるように、応用分野の経済学のみなさんに啓蒙を促そうというものです。計量理論の方々も参考になるのであればそうしてください。本書のめざすものは、アメリカにおいても日本においても、マニュアルやソースコードを学習することなしに、極めて自己流にゆがめられて教えられているプログラム教育および GAUSS のプログラム法に、かつを入れて、正すことにあります。過去に Fortran や Basic などプログラム集とかがありましたが、そういう羅列的な無駄なことはやりたくありません。正統的な読みやすい論理だったプログラムの仕方やテクニクそのものを伝授したいと思います。世の中にはお金だけで動く人ばかりだとは思わないでください。崇高な理念をもって闘う人がこの世界には数多くおられます。私もそのひとりではありますが、冒頭で述べたように、最終的には学会に発言権があるポジションを最低限得ることを目的にしています。

8) 本書は、少しフォーマルな読み物ではないのではないのでしょうか？

非常に provocative です。おそらく、既存の GAUSS 書きの学者さんたちの価値を低めたり人生を狂わせるかもしれません。しかしながら、GAUSS 文法通りにプログラムしていないことを喚起することは後学の人々によい影響と学問的効率をもたらすことと思います。読めないプログラムは所詮読めないし、頭の中もそうっておられるのだと思います。私が会ったことのない方には敵意は全くありません。非効率率は早く是正してってください。私が観察したかぎりでは、一般的な傾向として、Political Science 系の統計学と英語をよく理解する人たちは、GAUSS の文法に忠実にわかりやすいプログラムを書いています。一方経済学系の方は、特に韓国人や韓国人とジョイントをしている学者グループのコードは1つの独特なスタイルをもっていて、非常に読みづらいものになっています。私が見てもよくわからないようなコードは、おそらく、ほかの人にとっても解読するのに相当の根気と忍耐が必要になるでしょう。英語をろくにわからずにアジア系の T A や研究者が下支えしているアメリカの計量経済学教育の負の側面が如実に出ていたといえましょう。よく統計学や計量経済学の先生に、学期末には受講生が数人になるのが普通だという人がいます。

それは、あなたの過去の教育と師匠が劣悪であっただけです。計量経済学はどんな内容でもおもしろく実践的に教えられるものです。少なくとも、私はそう学びました。読めないプログラムを書いている偉大な学者や、プログラムで考えるとつじつまが合わない理論を展開する学者には、内外を問わず容赦ない「学問的攻撃」を仕掛けます。

9) 統計ソフトはほとんどが無料なのですが。GAUSS は本当に必要ですか？

計量経済学系の研究者ではなくて、統計学系の研究者は無料のものを使う、あるいは、自分で作ってしまって、学者の数だけソフトが無数にある状態になっています。統計学はそれでいいかもしれませんが、計量経済学あるいは経済学は実践的な学問であって、資源を集中して行なわないと効率的ではありません。アメリカの経済学の大学院のほとんどが必須としている現在、また、アメリカの金融系の政府機関にこのソフトを使う人たちが数多く存在する現在、GAUSS に集中して人的資金的資源を集中しないと学問的発展はありません。むしろこのことは、現実的にその都度目標や判断を柔軟に変えてくる中国人勢力や、とにかくいかなる手段を使っても真似て真似て真似たおして自分のものにしてしまう韓国勢力が GAUSS に集中してとりくんでいることから見ても重要さはわかると思います。統計系の学者さんは、偉いのはじゅうじゅうわかりますが、あなたの気まぐれは日本の若手研究者に非効率と学問的停滞を招いています。

10) GAUSS に統計プログラム言語を統一する必要があるとお考えですか？

そうは思いません。そういうエスプレント語的発想というか、国連軍創設の発想というか、そういう発想自体が無意味です。各言語には考案者や中興の祖、そして支持者たちの分野ごとの歴史や背景があるわけで、それらを破壊して GAUSS に統一しようとする運動をするつもりは毛頭ありません。むしろ、それとは逆に、ある程度のレベルの研究者は複数のプログラミング言語を互換的思考をもって自由自在に読み解く能力が必要であると考えます。私の属している学会は SPSS だからとか、何某先生の門下はすべて TSP だからとか、そういう発想は学問を閉塞した方向にもっていきだけです。持論としては、1つのプログラム言語、もう1つのパッケージ言語を自分の手足となるくらい完全に扱えるようになると同時に、それ以外のメジャーな言語はほとんどすべて「読める」力を持つべきだと思います。幸い、GAUSS は、プログラム上の手続きは最低限しかなく、アルゴリズムのみを示す簡易言語というかユニバーサルアルゴリズム言語の原型として最も適していると私は考えます。GAUSS を理解していれば、Matlab であろうと S であろうと何でもいけるはずで

本文中の用語について

Accept という表現について

アメリカの計量経済学統計学では、主に西海岸のカリフォルニアでは、Not reject のことを Accept と積極的に言ってもよいとする傾向にあります。その他の土地では Not reject は Not reject であって、Accept と表現するとすべて零点という習慣があるようです。本書はこの点を適宜、文脈に応じて使い分けています。

F Test という表現について

何をもって F テストするのが不明で（おそらく F 分布にしたがう統計値を用いたテストのことを言っているものと推察する）F テストという表現は本書は厳しく排することになります。日米に限らず、経済学部全体の用語上の悪習でありましょう。

Likelihood という用語について

戦前の敵性用語のような訳語が統計学や数学にはあふれている。Likelihood は英語で読んで字のごとく Likelihood であって、他のなにものでもない。Maximum Likelihood も同様である。無理をして怪しい敵性用語には訳さないようにところがける。Sample もサンプルであって、英語そのままかカタカナにするのが望ましいと私は考える。「重回帰」という日本語訳はおかしい。それに GAUSS では、1 変数の説明変数のケースも、2 変数以上の説明変数のケースも基本的に扱いは同じなので、「重回帰」というなやましい日本語はやめましょう。

n と k について

プログラム上で複雑な計算をする場合には異なる変数の設定をしているところもありますが、特にことわりのないかぎり、n はそれぞれのデータの数（すなわち行列の行数）、k は定数項があるならそれも含んだ説明変数の個数（すなわち説明変数行列の列数）と定義します。行列を用いない教科書で勉強されてきた方々は、k は定数項を含まない説明変数の個数と丸暗記されているかもしれません。行列を扱って行列で計算してしまう場合にはそれでは都合が悪くなります。本書では、説明変数側の行列の列数が常に k とし、したがって、自由度は常に $n - k$ とします。さらにトレンド項がつけば、それを含めた説明変数の個数全体を k とします。

私自身は“excellent”な人物ではありませんが、自分も含めたすべての人や状況のタイムパスをまったく変更してしまうような pretty darn “good”なものを書こうと思っています。偉い先生方が束になって莫大な研究費でやるようなことを、かろうじて Windows が動くような環境で、資本なし、まったくの素手でぶちのめしていきたいと思います。私がやっていることは、例えて言うならば Kurt Russel 主演の映画 Tombstone で Val Kilmer 演ずるガン使いの名手 Holiday が悪党どもの同じくガンの名手 Johnny Ringo にラテン語で話しかけて Educated man であることを確認してから、悪党 Johnny Ringo がぐるぐると早業でガンを振りまわすに対して、ガンの代わりに金属製のカップでぐるぐると回して見せる。ち

ようど、このようなことを GAUSS で素手でやって他の人を挑発しているものと解釈してください。これは計量経済学の分野の学者さんたちへの大いなる Challenge です。なお、間違いや勘違いの記述は気にしないで書き進めますので、その点はお許し願います。メインのアルゴリズム自体の根本的な誤りはないように努力していますので、細かい間違いは著作権事項にしたがって御自分で修正してお使い下さい。ご質問は E メールでご自由に。

ソフトウェアの興亡

日本のゆがめられた現実

日本では、なぜか計量経済学は基礎からやらずに、すぐにモデルビルディングにジャンプしてしまって、ソフトウェアも TSP オンリーになってしまっている。官僚も民間もそして学者もみんな無用な当たらない予測に興じている。また、労働経済学を TSP で無理やりやろうとする。これは、やるのは少なくとも誤りというか相手にされないことをしてる。かつて Welch 大先生に TSP はどうだと尋ねた時、彼は肩をすばませておられました。また日本では、学部は EXCEL で大学院は TSP という世にも不思議な構図が固まってしまっている。少しばかり先を見通したところでは RATS が扱われてきた。マクロや数学系がかかったところでは Matlab でもって、アメリカなどの大学者が組んだプログラムをわけのわからないまま、そのまま動かしている。その流れに、Wavelet を扱う時系列系の学者が合流している。違った流れとしてミクロ経済系は Mathematica を扱う。新しい流れとしてマクロ系および官庁エコノミスト系に Eviews に乗換えるグループがある。社会科学系では、SPSS を信奉する方々も根強い。また、日本だけ Greene 教授を信奉するグループがいて、ソフトウェアも日本では教科書と抱き合わせで訳書として販売されていることもあって、彼の本だけでなくソフトウェアも信奉されている特殊な状況ができあがりつつある。

しかしながら、少なくともアメリカでは 2000 年をすぎるとあたりから、計量経済学やシミュレーションは GAUSS、そして予測やマクロビルディングは Eviews、労働経済学および社会科学は STATA という構図に固まりつつある。一方、数学系工学系では Matlab、統計学系では SAS と S-Plus とその仲間がメインだ。農業経済学系では、SHAZAM と GAMS ということになっている。TSP は経済学の大学院では教えられることは欧米では少数派だ。

インターネットが与えた影響

今までソースコードが師弟関係という限られたサークルでしか回覧されていなかったものが、インターネットにより情報が世界を駆け巡るようになったのが大きい。その恩恵を受けて 90 年代半ばころから EasyReg をはじめとするフリーのソフトウェアが一時的にバブリーになった。しかしながら、それは同時に、空間を超えてソースコードを回覧してその次のものを生み出すという点で、開発者が同じ地方や州にいる必要がなくなったために、開発もアメリカ全土で行ないその送受信からソフトのアップデートまで Bill Gates ソフト社的にインターネットで自動で行なうようになった STATA (ステイタ) が急速に広ま

った。カリフォルニアという限られた地方や区域で限られた人たちによって開発が行なわれてきた RATS や Eviews は急速にその人的資源という問題から相対的に地位を落としている。一方、Finis Welch 大先生（彼は現在はプログラムはしていない。黒のカーボーイハットをかぶって車椅子から 牧場兼せり場を監督するように、にらみをきかせているだけである）とともにカリフォルニアから故郷テキサスに里帰りした STATA は完全に地域限定の人的制約をインターネットによって打破し、より強固になっている。（ただし、不幸にも STATA が存在する大学は韓国人勢力や日本人を嫌う南米系ユダヤ人およびユダヤ人勢力がいるために、近年日本人の学者はそこから輩出されないばかりかビジティングでさえも近づけないために、STATA の本当の使い方は日本には伝わってはいない。Finis Welch 大先生が UCLA にいた時代の資源が間接的に UCLA の Web 上に公開されていて、日本の学者は、それを頼りに手探りで行なっている状況であろう。）

今後の展望

Greene 教授の教科書は、1990 年代までに SAS と SHAZAM によってそのほとんどのデータと項目についてプログラムされてしまったので、Greene 先生のソフトウェアが今後日本以外の地で広まることは考えにくい。また、新しいバージョンで始めてリニアの制限を最大化に取り入れたように技術的には他を模倣する状態にある。TSP も最近になってやっと Procedure の機能を取り入れた。これらは、残念ながら他を追従する勢力に地位を落としている。遅れをとり相手の真似をすることに専念する言語たちとそうではなくて金融工学などに打って出て勝負する言語たちに大きく分化してきた。基本的には、低レベル言語としての戦いは、GAUSS と S-Plus（または R）と Matlab の戦いは今後も続くであろう。ここで Matlab は日本の高校高専でも教えられている工学のソフトであるので、発展することはあっても、早晚なくなるとは考えにくい。また、S-Plus は優秀なソフトウェア体系であるばかりか、その公式なクローン R はコードをすべて公開している上に成り立っている。GAUSS の勢力を削ぐことはあっても、衰退するとはしないであろう。S-Plus が手続的に遅いという指摘は確かなのだが、これが問題となって将来衰退するとは私はないと見る。むしろ、将来的にはマシンの性能があがって、遅い速いは基本的に問題とされない環境が数年のうちにやってくるものと思われるからである。金融工学の分野が S-Plus に投資しているのは、誤りではない。私の個人的な予想としては、これら 3 者は多少のオーバーラップはあるとしても三者三様に異なる分野で住み分けをしていくように思われる。ただし、S-Plus のクローンの R、そして GAUSS のコードを勝手にコンパイルした OX といったクローンの影響は大きいであろう。インターネットに続くエピソード 2 は「クローンの攻撃」であるかもしれない。他方、高級言語レベルの戦いとしては、現在 STATA と Eviews が壮絶な戦いを繰り広げている。SAS と同様にいくつかの分野に分かれて別売りがなされている SPSS は、既に単一売りパッケージの STATA の軍門に下ってしまった。Eviews は TSP の発展的な代用としてマクロモデル屋のツールとして残っていくだろう。STATA は、経済学という小さなマーケットではなくて、どうも統計関連のすべての分野の外堀を埋め

てから本丸の経済学を押さえようとしている戦略であるようだ。Stata は社会科学関係をすでにほぼ制圧しつつあり、現在の関心事は、医学および生物学の分野のようである。特に、STATA の開発環境のまわりには、そのプロトタイプを構築したという Welch 大先生が労働経済学の大御所であって Logit や Probit や Tobit 関係を完全な信頼性のもとに制圧しようとしているばかりでなく、同じ大学にパネルデータの大御所中の大御所、それにノンパラメトリックの新進気鋭の学者がいるので、少なくともそれらの分野は今後 2、3 年でかなりの発展を見るであろう。ただし、そこにはモデルビルディング関連の人材がいないように思われるので、Eviews が本来ターゲットとすべきマーケットへの進出は大幅に遅れるか、そもそも、開発はなされないように思われる。S-Plus(および R)は、低レベルの細かいプログラムをするためだけではなくて、高級言語としての性格も帯びている優秀な言語体系である。SAS のようにプログラムというよりは手続きに終始するようなプログラム体系を打ち破った優秀なツールと言えるだろう。統計学を知った本物の勢力が S-Plus(および R)をもって STATA と戦っていく状況が中長期的には続くように思われる。残念ながら、RATS にはもともと偉大な学者たちがついているが、ビジネスとして成り立っていないばかりでなく、インターネット社会や表計算ワークシート化の流れに完全に乗り遅れてしまっている。時系列系のプログラムはもとは RATS に書かれたものを他に移植したものが多くだけに、時代の移り変わりは非情である。Eviews は、信頼性の点で確かな人的資源に欠けている一方で、あまりに選択方式で手続きを自動化しているために玄人に嫌われる傾向にある。

日本では、Greene 教授を信奉する勢力や環境関係の人々が若干恣意的な動きを見せているので、日本においてどのような勢力地図になるかは予断を許さない。ただし、少なくとも長期的な趨勢（それが何十年になることも考えられるけれども）としては、TPS で時系列やモデルビルディングをやっている研究者のかなりの部分は Eviews に乗換え、労働経済学をやっている研究者は Stata に大挙して移行し、それで操作で結果が出せずにプログラムを必要とするものは、アメリカのトレイン教授の流れともあいまって、GAUSS で行なうようになるであろう。ただし、私が本書を書くことによって、機先を制された優秀な学者さんたちは、無料でアプリケーション的性格もある GAUSS のクローンの OX に行ってしまうであろうとは十分に予測できるし、その兆候が出始めている。また、今後発売されるであろう日本人学者による Structural Economics という英文の書籍が、GAUSS の言語的な使い方ではなく、自分たちが作ったパッケージの使い方を推奨しているので、GAUSS とはそういうものだとか合点してしまう方々を日本でたくさん創出するかもしれない。あわせて、OX のパッケージとしての関数の充実とあいまって、日本ではイギリスに対する思い入れがかなりつよい国なので GAUSS の使い方が相当にゆがめられてしまう恐れもかなりある。技術的には、信号や波動といった工学系の応用をそのまま持ってこななければならない分野では Matlab がローカルに優勢な状況が今後しばらく続くであろう。この流れには、マクロの理論の研究者も加わり、1 つの流れとして一時的に経済学部で勃興する兆しがある。金融工学系の学者や研究機関が今後どのソフトをメインに据えるかも、Economics と

Finance の学部が未分化で資源を共有する日本においては、外性的な要因として大いに影響してくるところであろう。EXCEL は、ジョブトレーニングの立場から学部教育として今後もしばらくは生き続けるであろうが、研究の分野では顧みられもされない分野になるであろう。ただし、90 年代半ばに STATA や Eviews、それに SHAZAM Pro に表計算ワークシートを導入させた功績は大きい。EXCEL は、計量のソフトにおいて、表計算ワークシートを用いる高級言語と、それを用いない細かい計算をする低レベル言語との、2 つの大きな流れを完全に分離する役割を果たしたように思われる。Bill Gates ソフト社の製品をはじめとする自動インターネットアップデートは、いち早く STATA にとり入れられている。こうした技術をもっているかどうか、急速に変化し、バグをリアルタイムで修正し、新しいコマンドをリアルタイムで出してくることで、計量ソフトの覇権の勢力地図拡大に大きな要因になっているし、今後ますますそうになっていくであろうことは間違いない。

計量経済学およびコンピュータ教育について

コンピュータを用いた計量経済学および労働経済学などの応用経済学の実習は、基本的には大人数でやることは不可能に近い。本家アメリカにおいても、コンピュータ実習は行なわれているが成功はほとんどしていない。その成績に占める割合は3割とか4割とか比重は大きいのだが、基本的にコンピュータ実習で出される課題はほとんどの人が満点に近いことを要求される。すなわち、100%に近い人がわからなくても他人のものをコピーしてコメント行や変数名などを付け替えて提出することが慣例化している。あとの割合を、中間と期末の試験で暗記の戦いになって、コードを書く技術の優劣は測定されない。TA制度が極度に発達して、宿題や課題は結果しか先生が見ない状況では、逆に試験の暗記力が劣る生徒でコードが書ける生徒は疑いの対象になる。これは、なにもプログラムだけにかぎったことでないが、宿題においても、そのような不当な疑いをかけられることは往々にしてありうる。この私も、南米系ユダヤ人の国際マクロの先生に「いったい君の宿題は誰が書いているのか？」と公衆の面前で詰問されたことがある。また、いくら計量経済学でいい GAUSS のコードを書いても全く評価されず、ほとんどすべての人がコピーして満点そして暗記のテストで優劣がつくという状況が続いていた。無論、過去 10 年分のすべての全問題解答付の資料をアメリカに渡る前から持っている韓国人と、affirmative が効いていて大学教授の problem solver がついている南米人やアジア系の女性には、天才でないかぎりかなうものではない。そこで、わたしは次のような提案をしたい。

- 1) コンピュータ教習の課題は基本的に、一人または1グループにつき「異なる問題」を1つずつ与える方法を採用する。

これにより、インストラクターの負担は人数またはグループの数の倍数だけ大きくなるけれども、インストラクターも自分の研究に関連するプログラムをより多く作成できるメリットがある。一人一課題はプログラム学習には特に有効である。また、超簡単な次に続く

ようなプログラムを回数をかせいで出題することが重要である。乱数設定なしで結果が全く同じであればどちらかがカンニングだという論理は GAUSS では通用しません。起動からの時間でシードは決まるので、同じタイミングならば、同じ結果もあり得ます。

2) コンピュータ課題は基本的にインストラクターが直接目を通し評価する。

これをしないと、だれが本当にプログラムを書く能力があるのか、だれがコピー屋さんなのか分からない。また、本当にプログラムに労力をさいて書いている人が勘違いをされて罪人にされることもありうる。プログラムの世界では、同じことを書いた場合、常に後から書かれたものの方が間違いもなく正確で、洗練され、読みやすいという法則がある。また、暗記に優秀な人はプログラムの能力が一般的にないというのもよくお目にかかる事実です。宿題と違って、TAにまかせきりではいけない。関与することが大切である。

3) エクストラクレジットの課題を採用する。

就職期や期末の忙しい時にではなくて、ほかのコースワーク上忙しそうにないときにエクストラクレジットを与える課題を募集することも大切であろう。インストラクターの研究に直接関連したものでもよいであろう。どんなマイナーな大学にもとんでもないハッカーやクラッカー、ゲームの達人とかは存在する。そういう人物にインセンティブを与えるためにも、その課題プログラムを作成して提出したら、期末のテストがたとえ平均点あたりであっても、自動的にAやA+を与える制度にしたい。プログラムの世界には、1万人の凡人よりも1人の天才（または偉大な苦勞人）がいれば事は足りる。たとえ、そのエクストラクレジットの課題を提出する人物が5年に一度か10年に一度であっても、それはそれで価値があるものである。

4) 計量経済学のTAや補助インストラクターにはプロを内外から採用してつける。

英語学習の場合、高校ではイギリス人やアメリカ人を英語の先生がともなって授業をやっていたり、ラジオテレビの講座でも常に横に英米人が控えていたりする。コンピュータ補助員には、特別に数年契約くらいの優秀な人物が必要である。そのような人物を奨学金対象に学内の大学院生から選ぶのではなくて、近隣の大学のオタクのような講師や民間の研究所の人物、または、夏休みに帰ってきている英米留学中のその大学出身の大学院生などにあたらせるとインストラクター自身の技能向上にも役立つであろうし、授業などが効率化する。ただし、計量経済学を知らない理系の人物を呼んできても、私は無駄であると思う。私が学んだ大学の経済学部には、3人から5人の理系のコンピュータテックが主に理系の大学生大学院生から選ばれて常勤していたが、それらの人の評判は先生方にはあまりよくはなかった。理系では理系なりの統計学の教え方があって、経済学部の計量経済学とやっていることが違うからである。また、UNIXのシステム設定やコンピュータのケーブルの配線などには通じていても、計量で使うソフト自体の専門家ではなかった。日本にも、理系ではなくて、経済学出身のEveiwsを使わせたらあの人しかいないという人は数人から十数人いるだろうし、RATSでもそうである。GAUSSに関しては、アメリカでも相当に教育制度は劣悪なものであるので、私以外に適当な人がいるかどうかはわからないが、とに

かくシステムではなくてその「ソフト」に通じた人物を補助員につけることが計量経済学という技能を経済学において活性化することになるであろう。

5) ハードではなくて、ソフトウェアを買おう。クローンでなく正式ソフトを。

日本の官公庁でも大学でも、モニターは液晶であるとか、コンピューター自身がラップトップであったり、しなくてもいい投資をしているが、ソフトウェアに金をかけているようには思われない。もちろん、Bill Gates ソフト社には多大な買い物をしているであろうが、計量や数学のソフトのライセンスを買うというソフトの面で投資をしないと、コンピューターはただの箱であって（日本では液晶であるから「面」であるかもしれない）人材も育成されない。私はマック派だと自慢して、公金で研究時間の過半をインターネットについやす人物をかつぼさせてしまうことになる。その反対に、計量や経済の演習は、OX や Matlab のクローンを大学として使用させるような馬鹿なことをやっている。クローンで教育された人は、ソフトウェアはただなものという観念は一生抜けないであろうし、ソフトウェアなんてそんなものだと考えてしまうであろう。GAUSS の製造元 Aptech 社はクローンの攻勢に対して、無料で FTP 経由で Light 版を公開するとともに、個人売りもするように最近変わった。日本の代理店を通すとまだまだ中間マージンが入り高いが、Student 価格でアメリカから（おそらくアメリカカナダ等の居住者に限られる？）直接購入すると数百ドルで手に入る。クローンではなくて、少なくとも、個人で Light 版で試した後は、正式版を大学や研究所単位でサイトライセンス契約をしていくことをお勧めする。それが、ソフト面の投資である。液晶やフラッシュメモリだけ買っても何も用をなさない。いくら百万もする小さなプラズマテレビで面白くない番組をみさせられる日本社会よりも、千数百ドルで買える 50 インチ以上の箱型のプロジェクションテレビで選びきれない面白い番組をふんぞり返って見ているアメリカ人の方が偉いし、指数乗おもしろい。ご存じのようにハッカークラッカーたちのそのほとんどがとてつもない遅い中古のコンピュータを使っていることがそのほとんどだ。マシンの性能で彼らの優劣はつかない。ソフトに投資しないかぎり人は育たないし、GAUSS は永遠に「プログラムが必要な」「行列操作の」言語であるといううわさが永遠に漂い続けるだけである。

6) 使うソフトウェアの強制はしてはいけない。

私が学んだアメリカの複数の大学院複数の学部では、2つの原則があった。1つは EXCEL を研究や宿題の本体には使わないこと。もう1つは、けっして1つのソフトウェアを使うと授業では決めてはいけないこと。Finance の専攻の人や、農業経済の人、時にはコンピューターサイエンスの人など大学院のコースにも他学部が制度的に必修単位でやってきたり自らの意思でやってこれる制度のもとでは、当然使い慣れたソフトも違う。また、工学部単位を取りにいくとみんな Matlab で宿題がだされたりする。閉鎖的な日本の学問社会ではしょうがないかもしれないが、できるかぎり使うソフトウェアを強制しないことが重要である。公共の教育機関が、ある会社のソフトに利することをするのは問題があることは言うまでもない。利することはあっても、あるグループからの自由選択に少なくとも

すべきである。もちろん、多数決で決めてしまうのも、かえって強制になる。特に、自ら既得権益守るために TSP に固執するのは醜いばかりでなく、学問を 10 年も 20 年も遅らせることになっている。私も、GAUSS がすたれて S-Plus や Matlab が優勢になれば、いさぎよくそちらに移る。ソフトウェアとはそういうものである。手続きではなくてどのような言語にも通用するアルゴリズムで勝負すべきである。

7) プログラムソースが公開されないモデルによる予測やモデルビルディングそのものは、そもそも無効である。

インターネットは何のためにあるのか？それはチャットや掲示板で面白おかしく戯れるためのものではなくて、出版されたもの、あるいは公表されたもののより詳細な資料を見たい人だけに十分に見てもらうことが本来の役割であろうと思われる。TSP をプログラムする人に真のプログラマーはいない中で、モデルのプログラムソースを公開しないのは、最初からプログラムは間違っていてそれを指摘されるのを恐れるか、もしくは大部分が何かの丸写しで著作権上公開できないかのどちらかであろう。プログラムソースが公開されない予測は、今後無視することが肝要である。初級までの計量経済学のコマンドしかない和製ソフトで、パッケージとしてモデル予測をするのは、それはまたそれで技術的に眉唾ものである。そのような技術しかないものにモデルビルディングのパッケージとしての計算をゆだねるのは無謀である。だいたい含まれているコマンドの分野の量や質によって、そのソフトを作っている人のだいたいのバックグラウンドや実力は測れる。

8) 計量経済学や経済学の学習とプログラム学習を分離するのはおかしい。

経済学部で理系出身の先生が、その他の教科のカリキュラムとは全く独立に、CGI や Java などのプログラミングを教えているのは、まったくの無駄というか、給料のただ払いとしか言えない。また、計量経済学の分野でも当講座はプログラミングは教えないので各自学習するようにと指示したり、T A に任せきりにするのは非常におかしいと言わざるを得ない。直接教科と関係のあるプログラム学習に指導者はコミットメントしよう。私が金融工学関係の教科書を完全プログラミング化をめざして取り組んである程度の成果をおさめているように、数値計算ができるもの、パターンがあるもの、逐次計算を必要とするものはすべてプログラム可能である。教科に直接関係のあるプログラム学習をすることが、教科の理論自身の深い理解やさらなる発展を考える力を養うために重要であると考えている。下手に理系の研究者をコンピュータの管理者にたてていたりすると、逆に自分たちの通信記録やすべての情報を秘密裏につかまれてしまっている恐れもある。暗号理論や Perl 関係に通じている研究者は、相当の倫理観と良心がなければ、おそらくその方は大なり小なり「Peeping Tom」である確率が相当高いことを指摘しておきます。研究者関係のないものは教えない。関係のないものは指導者にしない。直接に教科と関係ある題材を教えよう。

9) Discussion Data や Discussion Program を研究者は公開しよう。

失敗は成功の鍵である。うまくいかなかった計量データやそのプログラムを公開して、他の人たちにってもらうことは非常に重要であるとは私は考える。私の本書も一種、敗者の産

物なのであるが、多くの場合というかほとんどの場合、失敗したりこと自体や、失敗したプロジェクトは永遠に葬られる。敗者や失敗の弁は永遠に口をつぐまれる。成功は成功でそれでいいのだが、そこからは人々は学ぶことは何もない。100%成功すれば、みんな拍手をして何か賞が授与されるだけのことである。何も異を唱えることもなければ、何も学ぶことはない。世の中には、8割か9割成功したものが登場して、それを議論する。しかしながら、100%失敗したばかげた事象やプロジェクトからも学ぶべきことは多いはずであるし、考え方をかえれば、100%失敗したものは100%何かを学べることでもある。不完全なお蔵入りの論文、書きかけのプログラム、Reject されて日の目をみなかった論文やプログラムをインターネット上にのせることこそが、インターネットの本来の役割ではないだろうか？そういう意味もこめて、私はここでプログラムに特化させて（ほかの専門的なことを叫んでも誰も相手にしないでであろうから故意に他に媚びるようにこうしている）敗者の弁を大いに天下に知らしめているのである。失敗を大いに評価する文化や社会が世に訪れることを望む。また失敗したものが実力に応じて相応に評価される社会であってほしい。