

タイトル	Linuxと情報教育
著者	晴山，雅寛
引用	北海学園大学学園論集，137：1-24
発行日	2008-09-25

Linux と情報教育

晴 山 雅 寛

1 はじめに

高等学校で情報教育が必修化され、これを受けての大学での情報教育はどうあるべきかを考えてみる。現在、入学してくるほとんどの学生は高等学校で OS としての Windows の基礎的な学習を終えている。更に、アプリケーション ソフトウェアではワードプロセッサ(ワープロ)の基礎知識と表計算の一部を学んでいる。これらを学ぶためのソフトウェアは他にもいろいろあるにも係わらず、OS 同様にマイクロソフトの提供するソフトウェアである MS-Office に大きく依存しているようである。これに基づく教育を受けた学生にたいして、大学における情報教育が、高等学校で情報教育が必修化されていなかった以前のままだに、高等学校と同じソフトウェアを使い、その使用法を同じ様に繰り返し教えるだけでは不十分である。ワープロ教育では DTP(desk top publishing) の必要性や重要性は以前から言われていることである。今後、情報教育の場でこれを避けて通るわけにはいかない。一例として、数式の表示はこの種の多くのワープロでもそれ相当に表示可能ではあるが、実際に使ってみると結構手間暇のかかる作業になるため、学生は十分に使いこなせる段階に到達しているとはいえない。同時に教える側にも数式表示の規則性は無頓着なことも影響していて、数式表示に関して的確に教育していないのではなかろうかと推測する。自然科学や工学の分野では数式を日常的に使う必要があり、論文やプレゼンテーションの中で適切に表現することが第3者の理解を促す上で大切である。

表計算に関しては、MS-Excel はかなり良くできたソフトウェアであることはまちがいない。しかし自然科学の分野にはデータの精度、計算結果の信頼度が問題になる場合が多々ある。したがって、結果を得るための手続きをマニュアルにしたがって順に教育することだけでは、大学で行う情報教育とはいえない。用いたデータや得られた結果の吟味があってこそ初めて情報教育といえる。現在の学生は MS-Excel に関しても入出力の基礎知識を有している。それならばと、MS-Excel に組み込まれている関数の使いかたを教えることは高等学校で十分に教育されていないのだから、そのことを新しく教えるのが大学での情報教育である、とはならない。結果の吟味といっても、MS-Excel とその関数だけの範囲ではチェックすることすらおぼつかない。数値を自分で指定した精度で計算し、MS-Excel の計算結果やデータと比較検討するこ

ともまた、情報教育としては必要不可欠なことである。

MS-PowerPointはプレゼンテーションの道具としては、申し分のないソフトウェアである。現在、学生に対して最も必要とされる教育の一つが、自分の考えや意見を第3者に的確に、かつ、正確に伝えることのできるコミュニケーション能力を持たせることである。この補助手段として、MS-PowerPointは多くの可能性を持っている。しかし、これも的確な文章と吟味された計算結果があって、初めて成り立つことである。

プログラムを書いて、数値計算をしたり得られた計算結果を数値のみならず、グラフに描いて示すことは現象や結果を理解する上で極めて有効である。これらのことをWindows上で実行できるソフトウェアはMathematica他数多くある。しかし、フリーのソフトウェアであるMaximaを除くと、この種のソフトウェアは一般に高額であるため、なかなか教育の現場で活用されていない。このMaximaを使うと、数式処理のみならず、数式の2次元や3次元プロットを簡単に実行できる。例えば、スピンや角運動量の状態関数はルジャンドル同伴多項式を用いて表現されることが知られている。これらの状態関数は一般に正弦関数と余弦関数の組合せで与えられるが、実際の関数形は複雑で式から形を思い浮かべるのが難しい。これもグラフに表現すると極めて分かり易くなり教育的である。

これらの要求に対して、高額の予算を伴うことなく、すぐさま満たしてくれるのがLinuxを基本OSとして採り入れることであろう。もとより数多くの教育・研究上で有用なフリーのソフトウェアがLinuxの世界には蓄積されている。また、最近のLinuxやWindows上で動作するMS-Officeとの高い互換性を持つフリーのソフトウェアの登場によって、WindowsとLinuxの使用上での差は無くなりつつある。従って、ワープロが容易で、かつ、多くの人々が慣れ親しんでいるとの理由だけでは、今後もWindows上でMS-Officeを教え続けることで情報教育が十分に成り立つとは思わない。今まではLinuxを使いこなすには、UNIXの基本の知識を相当量必要とした。この高いハードルのため、一部の人々を除いてなかなか普及しなかったこともあろう。最近ではLinuxもWindowsとあまり変わりがなくインストールができて、各種の設定や更新が容易になってきている。この論文では、教育上で重要と思われるWindowsとLinuxにおける各種アプリケーションソフトウェアの使用上の各種の検討を試みる。

フリーのソフトウェアの中にもJavaやOpenOffice.org(Open Sweet Office)のようにWindows, MacintoshとLinuxの各OSでまったく同じに動作するものもある。OSを選ばないということは互換性の上から見ても、将来に対しての知的財産としても大切なことである。あるいは、文章閲覧用のソフトウェアであるPDF(Portable Document Format)も無償で提供されているが、一般には、複数のOSで動作しても有償の場合が多い。このような有償、無償を問わず各種のOS間で共通のソフトウェアの比較は意味がない。しかしながら、Javaを教育の場に積極的に導入するべきであるとの提案はここでしておかなければならない。今は、あくまでもLinuxを基本OSに選んだ上で、フリーソフトウェアがどの程度教育に役立つかに重点をおいて調べる。

次の2章では数式混じりの横書の日本語文書での句読点の使われ方、およびかな変換のソフトウェア間のローマ字入力の違いについて考察する。さらに数式表現の一般的な慣用法についても述べる。3章は表計算に関するソフトウェアとC++やJAVA等のプログラム言語を

用いる重要性について述べる。4 章はプレゼンテーションにおける表やグラフの重要性を例をもとに示す。5 章はまとめである。また、6 章と 7 章は付録である。ここで Vine Linux の導入法と各種の設定や、この論文で紹介したフリーのソフトウェアの使用法を簡潔にまとめた。

2 日本語入力

マイクロソフト社の最新の OS である Windows-Vist において、他の日本語辞書に先がけて漢字の略字表示をやめて正式の書式に改善したことは評価すべきことであろう。例えば、辻は 2 つ点の付いたしんにゅうの **辻** が (もまた?) 正当な漢字表示であることに気がつかない人が多い。これは全て今までのワープロ教育の弊害とは言えないまでも、ハードウェアの進歩にソフトウェアの進歩が追い付こうとしなかった一例かもしれない。日本語の言語ツール IME2007 と Anthy での入力の違いも含めて、教育上気になることを以下の小節で記すことにする。

2.1 句読点

横書の日本語表示において、句読点については 3 通りの使われ方が多くの書物に見受けられる。縦書きの文章では、(。) と (、) のみであるが、横書の文章においてはこの他に、(ピリオッド。) と (カンマ,) の組合せや (。) と (,) の組合せでも使われている。一般的な傾向として、半角の英数字や数式を用いた文章を多く含む文章においては、(。) と (,) または (。) と (,) が多く使われている。前者には、「数学辞典」や「物理学辞典」に代表される多くの辞典類や「理科年表」, 「日本物理学会誌」, そしていくつかの数物関連の雑誌がある。後者の代表的な使われ方として、高等学校用の横書の教科書やセンター試験に代表される大学入学試験の問題文がある。他にも一般科学の啓蒙雑誌のかなり多くが使っている。大学で用いる教科書には (。) と (,) の組合せと (。) と (,) の組合せの両方が見受けられる。一方、古来の代表的な句読点 (。) と (、) が横書の自然科学関連書物にまったく使われていないわけではない。「文部科学時報」(インターネット版) や「学術の動向」(インターネット版) といった国の出版物や一部の科学一般の雑誌に使われているが、それほど多くはない。

表 1 英字と数式が混じった横書文章中の句読点の例

。と、	・と,	。と,
文部科学時報 学術の動向	数学辞典 物理学辞典 理化学辞典 理科年表 日本物理学会誌 教科書 (大学)	日本学術会議ニュース 全ての教科書 (高校) 教科書 (大学) 大学入試問題 (センター, 大学)
雑誌の極一部	「数理科学」他雑誌の一部	「ニュートン」他雑誌の一部

横書の文章の句読点をどう選ぶのかの規則性は無いが、半角の英数字や数値、関数が多用される文章では句読点として(。)と(、)の組合せを使わないのが望ましい気がする。即ち、 A 、 B 、 C 、 $\sin \theta$ と書くよりは $A, B, C, \sin \theta$ とすると、日常的に英語の論文を読んでいる人は抵抗感無く落ち着いて読めるであろう。一方、日本語の文章であれば、句点の前は必ず動詞なので、読点の(,)を受けて(.)で終わってもよいし、和洋折衷になるが教科書スタイルの(。)で終わっても特に違和感はないであろう。

学生はこのような句読点の使われかたをしていることに気がついていない。また、高等学校の情報で教えていない気がする。マイクロソフトの提供する日本語IME2007では、default設定が(。)と(、)になっているが、(。)と(、)を含む他の表記を初期設定として選ぶことも可能である。一方、Linuxで使われているAnthyやCannaでは(。)と(、)の折衷タイプは初期設定ができないため、最初に(。)と(、)の組合せから(、)を(、)に代える手間がある。

2.2 ローマ字入力変換

ローマ字入力の方法に古くから訓令式とヘボン式がある。ヘボン式の標準入力の規則性を知らないけれど、IME2007とAnthyの間には次の表に示すかな変換への違いが見受けられる。

表2 IME2007とAnthyのローマ字変換の例

ローマ字入力	IME2007での変換	Anthyでの変換
chuus(h)inn	ちゅうしん	ちゅうしん
cyuus(h)inn	ちゅうしん	×(変換されない)
tyuus(h)inn	ちゅうしん	ちゅうしん
chokinn	ちょきん	ちょきん
cyokinn	ちょきん	×(変換されない)
tyokinn	ちょきん	ちょきん
dhisuku	ディスク	ヂスク
dyisuku	ヂユイスク	ディスク
dhisupurei	ディスプレイ	ヂスプレイ
dyisupurei	ディスプレイ	ディスプレイ

当然のことながら、訓令式でのローマ字入力を行えば、両者でかな変換の違いはない。しかし、ヘボン式のローマ字の入力では、かな変換に対して2つの間に少々違いがある。これらに対する対処法を考えてみると、規則性が無い現状では、それぞれの場合、場合に合わせるし

か方策はないようである。学生にはこのように日本語へのかな変換に曖昧性や多様性があることを理解させることもまた、必要なことであろう。こう見てくると、将来もしローマ字からのかなへの変換に際して、何らかの混乱が生じるのなら、数式や英単語の混じった日本語の文章表現やローマ字入力とそのかな変換に対してある程度の指針が必要になるかもしれない。

2.3 日本語の文章中の数式表現

数式を表現するには、3種類のフォントを使い分ける必要がある。 x や t をスカラー変数として扱う際には、細字のイタリック体を使うことで定義無しに暗黙に了解される。同じ変数であってもベクトル変数なら、 $\boldsymbol{F}$ や $\boldsymbol{x}$ の如く太字のイタリック体で表すのが慣習である。一方、 $\sin$ や $\log$ といった関数の多くはイタリック体で表記せずにローマン体(立体)で表記する。更に、数字と幅乗を与える添字やテンソルの成分を示す添字が数字の場合にはローマン体を使う。

これらの慣習に従って数式を英字やギリシア文字を用いて表記するには $\text{\LaTeX}$ では簡単であるが、MS-Office、Office 2007 や OpenOffice.org では少し面倒できれいな仕上りにならないのが実状である。きれいな印刷物を作ること、読む人に内容を分かりやすく印象づける上でも必要なことであろう。ワープロのソフトウェアでは、一般にメニューバーの I のアイコンをクリックすることで、イタリック体の活字を選ぶことができるが、英数字やギリシア文字において、このイタリック体は正確にはスラント体(斜体)である。例えば、ローマン体の英小文字 a のスラント体は a であり、数式に用いるイタリック体は a である。このため一般に、数式表示用のイタリック体は数式エディッタとして別に用意されている。Office 2007 では、数式の表記がかなり改善されてきれいな仕上りを期待できるようになってはいる。これもそれなりの手間暇をかけた上でのことである。一方、MS-Office XP や OpenOffice.org でも、数式エディッタが別に用意されているが、美しい数式表現にするのはある程度の技術がいる。ところが、 $\text{\LaTeX}$ では、 $\$$ と $\$$ の記号ではさむだけで数式表現が実現する。もし数式中に関数が含まれるのなら、US キーボードならバック スラッシュ $\backslash$ または、日本語のキーボードなら通貨の円記号 $\yen$ (実際は半角の記号) を関数の頭につけて、 $\backslash\sin$ または $\yen\sin$ と書くことで変数と区別される。またギリシア文字は例えば、 θ は $\backslash\theta$ ($\yen\theta$) と lower case で、 Θ は $\backslash\Theta$ ($\yen\Theta$) と 頭文字を upper case で、ギリシア文字そのものの呼び名を書くことで表記される。

学生はこの数式表現の慣習を理解していないことが多い。その理由の一つに高等学校の物理の教科書では、ニュートンの運動方程式があえて次の様に記述されている。

$$\boldsymbol{F}=\boldsymbol{ma} \quad (1)$$

これは、ニュートンの運動方程式は大切な公式であるとの観点から、注意を喚起するために等号を含む全てを太文字で書いてある。この公式も上の表記法に従うのなら、力と加速度はベクトルの物理量であり、数式の等号 $=$ と質量はスカラー量で書くことになるのだから、

$$\boldsymbol{F} = m\boldsymbol{a} \quad (2)$$

とするのが一般の理工系の専門文書に見られる表現である。

関数をローマン体で示す表記法に関しては、高等学校の物理や数学の教科書では慣習の通りに書かれている。しかし実際の教育の場では教師が板書を速やかに行う等の目的のためか $\sin\theta$ の如く書くことが多い。これを受けて生徒はノートに無意識にいわゆる、筆記体で写しとるようである。これが癖になっていて、関数の表記法に無頓着になってしまうのではないだろうか？

将来、論文や報告書を書く際に、慣習的な数式の表記法を知っていることも必要なことである。しかし MS-Office, Office 2007 や OpenOffice.org で数式混じりの文章を記述することは面倒である現状では、理工系の学生には $\text{\LaTeX}$ の使用法を通常のワープロ教育と並行させて教育する事がよいと考える。なおかつ、多くの Linux ではこの $\text{\LaTeX}$ が標準のソフトウェアとして付いている。書き出しのコマンドさえ理解すれば、 $\text{\LaTeX}$ で文章を書くことはそれほど難しいことでは無い。

3 表計算

表計算のソフトウェアは大変実用的である。しかし、大学教育の場において単に実用性のみを教えるだけで良いのであろうか？ かつて、パーソナルコンピュータの黎明期において、まともに見えるソフトウェアが何もないとき N88BASIC なるプログラミング用ソフトウェアが計算機にインストールされていた。電卓に比べると BASIC のプログラムが簡単に走るのだから多くの研究者を魅了したものである。だが、すぐに倍精度計算が桁落ちし、結果が信用できないということが分かり多くの人を慌てさせ、失望させたものである。この轍を再び踏まないためにも結果にいたる計算過程や公式、なかんづく統計処理法などは一度は基礎から教え、それを表計算のソフトウェアだけに頼らずに、他の手段で丹念な計算を実行して、その得られた結果とを比較検討させることも大切であると考え。

測定値などの処理に際して、表計算のソフトウェアを用いて処理すると、すぐさま何がしかの結果が得られるため、その結果を漫然と受け止め、結果の吟味さえ行わないといった事になりかねない。細心の注意払わないと、有効数字すら意識しなくなる。実際のプログラム言語でプログラムを書いて処理するなら結果の桁数もプログラム中で考慮される利点がある。

数多くのデータを最小二乗法によって 直線への当てはめ等で近似する処理が実際の仕事の場では多いことであろう。これも表計算の関数を使うとすぐに関数形が求まり、直線近似であるならグラフに直線がすぐさま描かれる。最小二乗法とはどのような処理方法なのかを何も知らずにグラフは描けるのである。教育の場では、統計の初歩的な意味を教え、それをプログラミング言語を用いて実際に計算して初めて理解できることでもある。

統計誤差が全ての測定点で同一なら表計算のソフトウェアからすぐに最適な曲線を引くことが可能である。ところが同じ測定をしても、別の実験グループが異なる機器等を用いて実験結果を出すと中心値も統計誤差も同じ測定点であっても大きく違っている場合がある。このような複数のデータの混在したデータ処理には現在の表計算のソフトウェアは無力である。

このようなデータから最適な曲線を得るための手法である統計誤差をウェイトにする処理法や、 $\chi^2/e.v.\chi^2$ ($e.v.\chi^2$ は *expectation value of χ^2*) による曲線との一致度についても触れる必要がある。これを実践するには、Linux 向けのフリーのソフトウェアであり、現在は Windows 用にも提供されている Gnuplot や Maxima を使うのがよい。これらのソフトウェアは計算や作図を簡単に行う事ができる。あるいは、少々理解が難しいが OS によらない JAVA や C, C++ を使った教育も必要になるであろう。表計算のソフトウェアが必ずしも万全ではないことを知らしめることは大切な事である。

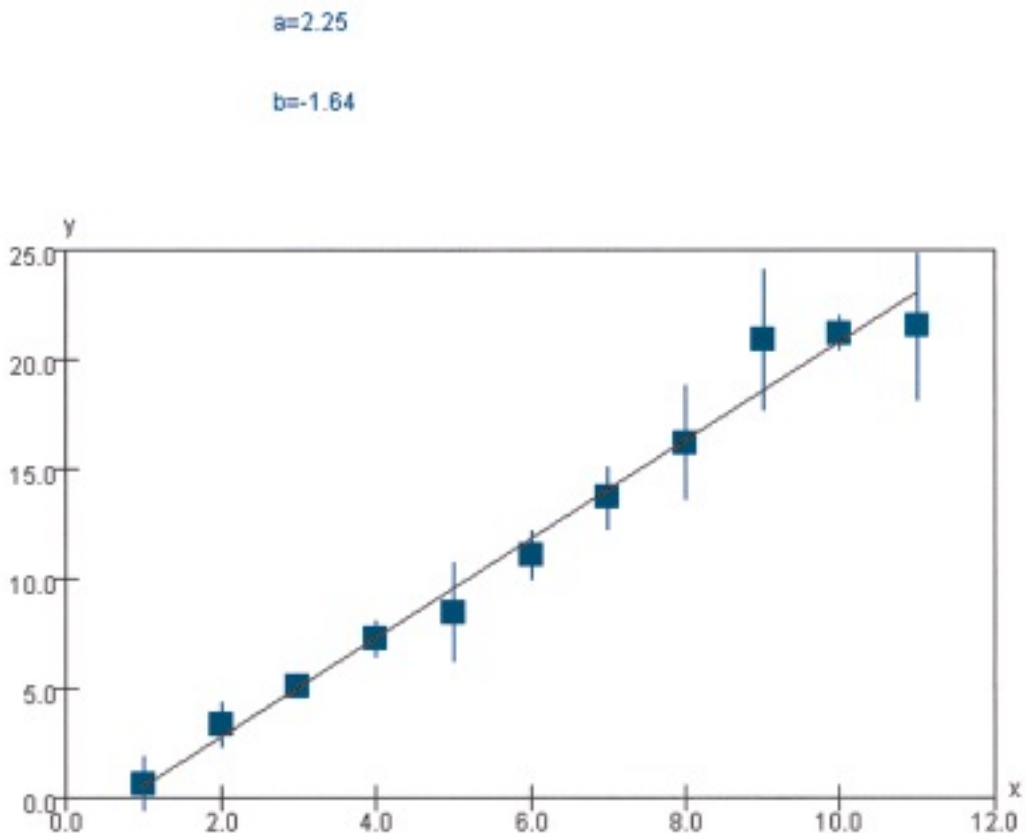


図 1 誤差を含んだデータの最小二乗法による解析の例

具体的に示すと、図 1 のデータは測定点ごとに統計誤差の値が異なっている。このときの直線近似 $y = ax + b$ を与える最小二乗法の計算は、それぞれの測定値の統計誤差 ε_i の 2 乗を重

みとして

$$a = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sigma^2} \quad (3)$$

$$b = \frac{\langle x^2 \rangle \langle y \rangle - \langle x \rangle \langle xy \rangle}{\sigma^2} \quad (4)$$

で与えられる。ここで $\sigma^2 = \langle x^2 \rangle - \langle x \rangle^2$ は分散である。また、 $\langle \dots \rangle$ は $n = 11$ 個のデータの重み付きの平均値

$$\begin{aligned} \langle x \rangle &= \frac{\sum_{i=1}^n \frac{x_i}{\varepsilon_i^2}}{\sum_{i=1}^n \frac{1}{\varepsilon_i^2}} \\ \langle y \rangle &= \frac{\sum_{i=1}^n \frac{y_i}{\varepsilon_i^2}}{\sum_{i=1}^n \frac{1}{\varepsilon_i^2}} \\ \langle x^2 \rangle &= \frac{\sum_{i=1}^n \frac{x_i^2}{\varepsilon_i^2}}{\sum_{i=1}^n \frac{1}{\varepsilon_i^2}} \\ \langle xy \rangle &= \frac{\sum_{i=1}^n \frac{x_i y_i}{\varepsilon_i^2}}{\sum_{i=1}^n \frac{1}{\varepsilon_i^2}} \end{aligned} \quad (5)$$

である。この直線近似のデータとの再現性を見る $\chi^2/e.v.\chi^2$ は、(3) と (4) 式で求めた a と b を用いて

$$\chi^2 = \sum_{i=1}^n \frac{[y_i - (ax_i + b)]^2}{\varepsilon_i^2} \quad (6)$$

を $e.v.\chi^2 = n - 1$ で割った値になる⁽¹⁾。ここで用いたデータでは $\chi^2/e.v.\chi^2 = 0.21$ と小さな値であるが、図1を見ると直線近似が実験データと極めて良い一致をしていることから理解できる。測定点の実験値の分布がガウス分布をしていれば、 n 個のデータの χ^2 の期待値は $n - 1$ である。従って、 $\chi^2/(n - 1)$ が1以下であれば、統計処理の結果は受け入れることができる (acceptable であるという)。

Office 2007 と OpenOffice.org の間では含まれている関数も大差なく、表計算に関しての特段の大きな違いは無い。ただ、背後で実際に計算を実行している数値処理用のプログラムが走っていることを考慮すべきであろう。従って、できるなら得られた数値をクロスチェックするにはこれらとは別の数値処理のプログラムを実行するのがよい。最低でもこれらを Office を用いてステップごとに計算し、得た結果を整理して比較検討することを一度は教育することが望ましい。

こう考えてくると、やはり Linux を基本 OS に選んでおくとう便利である。Linux は本来 C で書かれているため、C や C++ のソフトウェアを新規にインストールしなくてもすぐに使うことができる。これもまた Linux を基本 OS に選ぶ際の大きな利点の1つといえる。

Office 系では、表計算の表や図をワープロに反映できる利点がある。これを使うと表計算側で誤りを修正するだけで、ワープロの文章中に引用されている表や図が自動的に修正される。しかしながら、実際の場合では真に修正が反映されたかを比較検討することが必要であって、便

利かどうかは一概には言えない気がする。一方、 $\text{\LaTeX}$ を用いて、文章中に表やグラフを書き込むのは、これらの表やグラフを別名の通常は EPS(Encapsulated PostScript) ファイルにして保存し、ファイル名を通してこのファイルを $\text{\LaTeX}$ の文章中に取り込むのが一般的であるから、図表の修正が反映されているのは明白である。マイクロソフト社以外の多くのソフトウェアメーカーの提供するグラフィックス関連のソフトウェアでは標準として EPS ファイルへのエクスポート機能が備わっていることを注意しておく。

4 数式処理とプレゼンテーション

統計処理や数値計算の結果はグラフにするとわかりやすい。この点前節の Office には MS-Office, Office 2007 のみならず OpenOffice.org でも同じようにグラフにして表示できる。しかし 2 次元や 3 次元関数の特定の変数領域での関数値を 2 次元や 3 次元グラフで表現することは Office 系では難しい。このようなグラフィックスを可能にする代表的なフリーのソフトウェアとして、Maxima がある⁽²⁾。Maxima は Linux 版同様に、Windows 版も提供されており、グラフィックのみならず微積分を伴う数値計算や行列、行列式の値の計算においても、この種の代表的なソフトウェアである Mathematica に匹敵するかなり強力な数式処理ツールである。例えば、行列式の値の計算や逆行列を求めることは手計算では時間がかかり計算間違いも多い。数学での定義を学んで後はこの種のソフトウェアを使うのも良いことである。

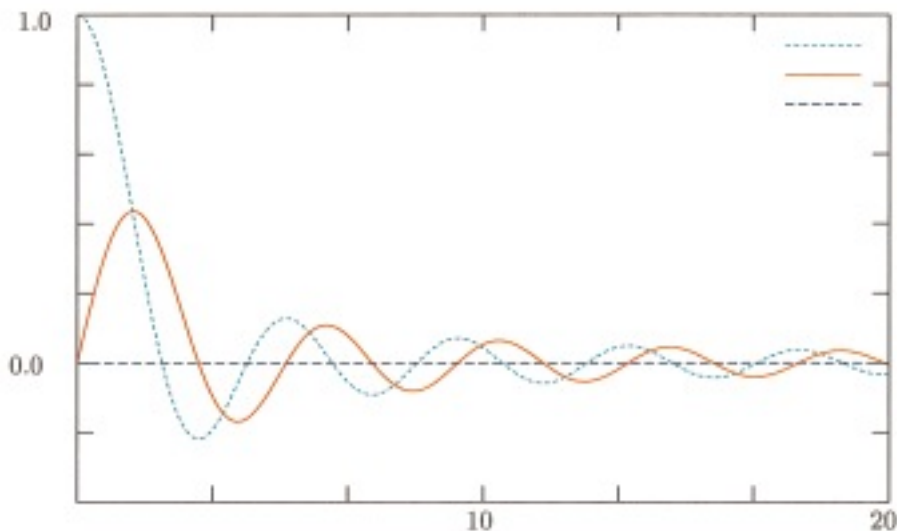


図 2 $j_0(x) = \frac{\sin(x)}{x}$, $j_1(x) = \frac{\sin(x)}{x^2} - \frac{\cos(x)}{x}$

Maxima による 2 次元グラフ表示の例として、球ベッセル関数 $j_n(x)$ の中の $n = 0$ (緑の点

線) と $n=1$ (赤の線) を $0 < x < 20$ の範囲について図2に示す。これは数学の $x \rightarrow 0$ への極限の振舞を理解する上でも役に立つ。

$$j_0(x) = \frac{\sin(x)}{x} \quad (7)$$

$$j_1(x) = \frac{\sin(x)}{x^2} - \frac{\cos(x)}{x} \quad (8)$$

統計力学における2次元空間での気体分子のある温度における運動エネルギーの分布はマックスウエル分布になる。これはガウス分布関数を用いて与えられ、

$$f(x, y) = (x^2 + y^2)e^{-(x^2+y^2)} \quad (9)$$

の図3に図示する3次元グラフの振舞と同じである。

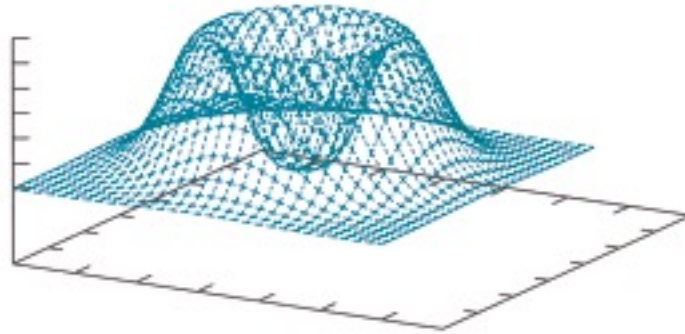


図3 $(x^2 + y^2)e^{-(x^2+y^2)}$ v.s. $x-y$

量子力学において、角運動量の演算子 L_z と L^2 の固有値と固有関数は、次の固有値方程式で与えられる⁽³⁾。

$$L_z Y_{\ell m} = m \hbar Y_{\ell m} \quad (10)$$

$$L^2 Y_{\ell m} = \ell(\ell+1) \hbar^2 Y_{\ell m} \quad (11)$$

ここで、 $Y_{\ell m}(\theta, \phi)$ はルジャンドル同伴多項式 $P_\ell^m(\theta)$ で展開される。

$$Y_{\ell m}(\theta, \phi) = (-1)^m \left[\frac{(2\ell+1)(\ell-m)!}{4\pi(\ell+m)!} \right]^{1/2} P_\ell^m(\theta) e^{im\phi} \quad (12)$$

ルジャンドル同伴多項式は $\sin \theta$ や $\cos \theta$ の多項式で表される。

$$P_\ell^m(\theta) = (-1)^{\ell+m} \frac{(\ell+m)!}{(\ell-m)!} \frac{(1-\cos^2\theta)^{-m/2}}{2^\ell \ell!} \left(\frac{d}{d\cos\theta} \right)^{\ell-m} (1-\cos^2\theta)^\ell \quad (13)$$

ここに、例として $\ell=2, m=0$ と $\ell=2, m=\pm 1$ の 2 次元と 3 次元プロットを、それぞれ、図 4 と図 5 に示す。これらから角運動量状態を図形で見ることができて、わかりやすいであろう。なお、これらの図では縦軸と横軸の縮尺を変えている。

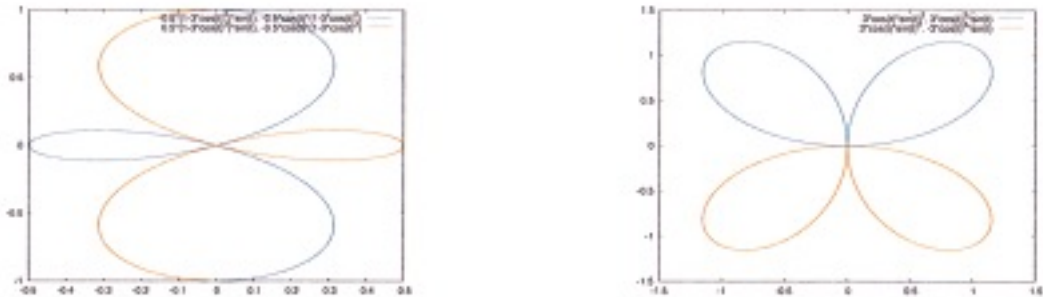


図 4 $P_2^0(\theta)$ と $P_2^{\pm 1}(\theta)$ の 2 次元プロット, 横軸は $x-y$ 面への射影

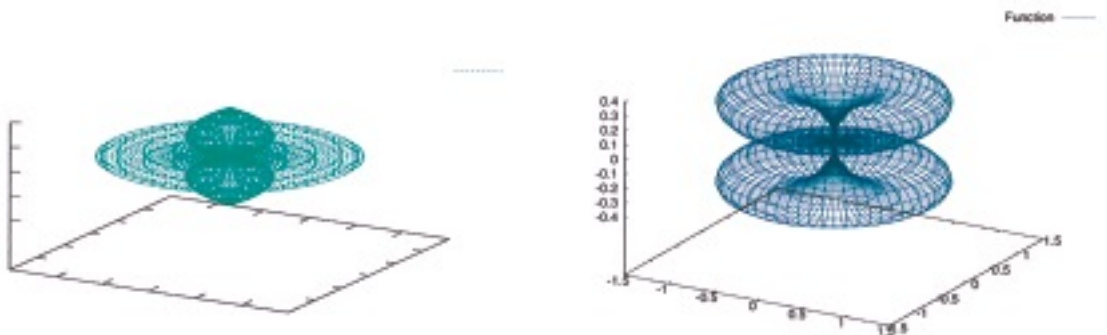


図 5 $P_2^0(\theta)$ と $P_2^{\pm 1}(\theta)$ の 3 次元プロット図

このようにして得た図やグラフをプレゼンテーションのツールである Windows なら MS-Office Powerpoint や Office 2007 Powerpoint で Linux なら OpenOffice.org Impress に組み入れて、第 3 者に対して分かりやすく説明できる。分かりやすくするには、図やグラフを用い

るだけではなく、簡潔に説明可能な作文の能力を最も必要とすることであり、この点は今の学生に欠けている点でもある。

OpenOffice.org Impress は Office 2007 Powerpoint に比べて、背景やアニメーション機能は少ないけれど、通常の使いかたをする上では大きな違いは無いと言える。

5 まとめ

ここまで見てきたように、OS に Linux を選ぶことにより、多くのフリーのソフトウェアが使用できることが分かる。なおかつ、これらのソフトウェアはどれも理工系の教育や研究の場で日常的に必要とされ、研究発表や論文を書く上でまた欠かせないツールにも成りうるのである。

一方、数多くの人が利用しているソフトウェアの使用法を教育することの方が、将来的に見てさほど互換性のないソフトウェアを教育するよりも有効であるとの意見も多くあるであろう。しかし、Office 2007 に固執して使用法を教育することが将来に渡って大きなメリットをもたらすとも思えない。多くの OS やソフトウェアの使用法を理解した後でなら、誰であっても Office 2007 を理解することはそれほど困難な事ではない。特に、Linux の OS の環境にあっても OpenOffice.org がインストールされてさえいれば、例え Office 2007 で書かれた添付文書があっても互換性に基づき読むことは可能である。

現在のインターネットを論文や研究連絡の重要な手段として発展させてきた、一番の功労者であるともいえるアメリカの物理学会や数学会は、MS-Office や OpenOffice で書かれた論文や公式文書を添付ファイルとして付けたことはない。全ての論文等の文書は OS に依らずに購読が可能な PS ファイル や pdf ファイルに変換して配布している。

数式を含む文書は $\text{\LaTeX}$ で書くのが美しい仕上りになり、いまのところこれに取って代わるソフトウェアは見当たらない。また数値計算のプログラムに関しても C++ や Java が標準的であり、将来性もあるであろう。

6 付録1. Vine Linux のインストールと設定

現在、数多くの Linux が提供されている。ここでは、導入とシステム構築や更新が比較的やさしい、Project Vine のメンバーによって日本語で提供されている Vine Linux (現時点の最新バージョンは 4.2 である。) のインストールと設定方法を紹介する。インストール作業は Windows の OS を新しくインストールする作業とほとんど同程度と考えてよいであろう。もしインストール途中でうまくいかなくなり、その原因が特定できないときは、NIC(Network Interface Card) や SCSI などの古いカード類を 1 枚ずつ抜き取り、CD からの再インストールを試みてみるとよいであろう。インストール後のメンテナンスも簡単ではあるが、念のため UNIX のコマンドをコンパクトにまとめた参考書を手元に用意することで、いろいろと役に立

つに違いない⁽⁴⁾。

6.1 入手方法

1. 北陸先端技術大学院等のミラーサーバからインターネット経由でダウンロードする。(参照: <http://vinelinux.org/>)

Windows や Macintosh のパソコンでダウンロードのみ行い (実行しない), ファイルとして保存する。この保存ファイルを CD-R 等にコピーする。700MB の CD1 枚に収まる。(上級者向?)

2. Amazon.co.jp から本または CD を購入する。Version 4.2 がなければ, 4.0 または 4.1 でもよい。インストール後にバージョンアップする。(中級者向?)

3. 書店に出向き, 付録に Vine Linux Ver.4.0 または 4.1 の CD のついた解説つきの書籍本を購入する⁽⁵⁾。(初級者向, 付録の CD が Vine Linux Ver. 3.* は買わないのがよいであろう。)

これらのどれかによって, 入手する。しかし, インストール作業に移ったとき 1. の場合にコンピュータの立ち上げの Bios の設定を CD に変更しても, コンピュータに組み込まれている CD や DVD ドライブの機種によっては, CD を読めない場合もあるので注意する。

6.2 インストール環境

基本的には, あまり古くなく, かつ最新の周辺機器の組み込まれた機種でなければ, Vine Linux をインストールできる。Pentium3(450MHz) で RAM サイズが 512 MB あれば快適に動作する。ハードディスクの容量は 40GB 程度あればよい。また, 使っている NIC が 10-Base 対応なら, 今後に備えてギガビットにも対応可能なものに変更し, 更に複数の USB2 と IEEE1394 ポートのついたカードがあると何かと便利である。SCSI のカードを認識しない場合がある。このときは内蔵の SCSI ドライブは使用ができないが, 外付けの SCSI 機器は USB2 への変換コネクタをつけることで USB2 接続の機器として認識される。

6.3 インストール

CD を挿入すると, Vine の名前から予想される葡萄のつるや房の下に, /boot で止まり入力が促されるので, Enter を押す。ほとんどの場合にモニターを自動認識する。モニターの解像度をモニターに付いている説明書に従って選択する。例えば, 21 インチの正方形に近いモ

ニターなら 1280×1024 を選ぶとよいかもしれない。この選択だけで済んでしまうことは、Linux のインストール経験のある人にとっては、画期的なことであろう。キーボードや言語、地域そしてインターネット接続を DHCP 接続とするのか Static IP 接続にするのかを選ぶあたりは Windows と変わらない。

ハードディスク hda にインストールするとして、この分割も自動化されている。手動で試みるのなら、マウントポイント /boot に 200MB、/ には /swap に 2GB 程度とった残りの全ての容量を割り振るとよい。こう割り振ると 40GB のハードディスクなら hda1=200MB(boot として)、hda2=37GB(Vine Linux 本体のインストール先として)、hda3=2GB(Swap として) 程度のメモリーが割り振られる¹。Swap はハードディスクの容量の一部を RAM に代用する仕組みである。ファイアウォールの設定では、HTTP と HTTPS を指定から除外しておく。Root の pass word(pw) とユーザー名とユーザーの pw を入力する。ユーザーが複数のときには残りのユーザーはインストール後に追加する²。デスクトップのシェル等はデフォルトのままでよい。こう選ぶと、シェルが bash で Window Manager に GNOME が選ばれる。最後にフォーマットとインストールが自動的に行われる。インストールが終わると、バージョンが 4.0 と 4.1

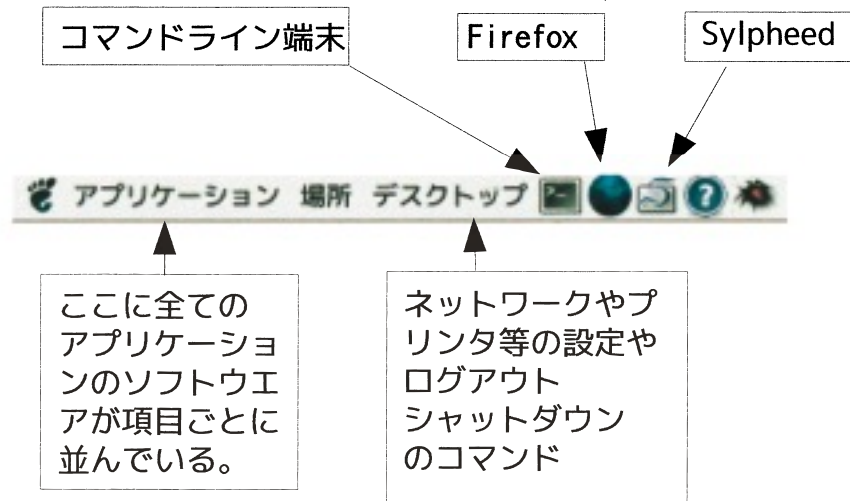


図6 Vine Linux のタイトルバー

では、雲の浮かんだ青空を背景に Vine Linux と表示されるであろう。メニューバーは図4に示す GNOME のシンボルの「足跡」から始まり右に「アプリケーション」「場所」「デスクトップ」ディスプレイ、地球以下各種のアイコン... が並んでいる。右端の時刻やスピーカーのアイコンの中にキーボードのアイコンがある。この状態では半角の英数字が有効になっている。漢字

¹Linux のディレクトリツリーは一番下に / があり、その上に /bin, /boot, /root, /usr, /home... がある。ここで、/usr には共通のアプリケーションのソフト等が格納される。一般ユーザーが各自のファイルを格納するのは、/home の中にある /haruyama のごとくユーザー名の付いたディレクトリである。これがマウントポイント / をハードディスクのどこにとるのかを指定する理由である。

²メニューバーの「デスクトップ」→「システム管理」→「ユーザーとグループ」から追加する。

入力は Windows XP 以前の使いかたと同じで、**Alt** + **半角/全角 漢字** (US キーボードなら **~** 記号のキー) で王冠のアイコンに変わり漢字モードになる³。日本語は Linux ユーザーに馴染みの Canna から Anthy に変更されている。一方, emacs や mule を起動したときは, **Ctrl** + **o** で漢字と英数字が交互に入れ替わることに変わりがない。このときの終了のさせかたも **ファイル** → **保存** からでもよいが, **Ctrl** + **x**, **Ctrl** + **c** で **y/n** から **y** or **n** を選んでもよい。また, 画面の上にコンピュータ, 誰それ (ユーザー登録をしたユーザー名) のホーム, ゴミ箱のアイコンが表示されている。これらは Windows のアイコンと同じ機能を有する。

アプリケーション 場所 除ができる。



Windows と同じ感覚でファイルのコピーや移動・削除ができる。

外部接続機器を取り付けたときに表示される。今はUSB-メモリースティック

図 7 Vine Linux のホーム画面

Vine Linux のバージョンアップをするには, root とユーザーの 2 者共にインターネット接続を有効にしておく必要がある。両者で別々に内容的には同一のネットワーク接続を最初に行う。まず, メニューバーの **デスクトップ** → **設定** → **ネットワークのプロキシ** を選んで, 適切に設定する。次に同じく **デスクトップ** → **システム管理** → **ネットワークの管理** も設定する。これらの設定がユーザー名で login したときには, root の pw が必要になる。ここでの設定では, 更にネットワークに関する DNS 等のアドレスを入力する必要がある。

最後にメニューバーの青い地球のアイコン (インターネット ブラウザの Firefox コミュニティーエディション) をクリックし, **編集** → **設定** → **詳細** → **ネットワークの接続設定** からプロキシの

³Windows に使われている MS IME2007 では **半角/全角 漢字** キーを押すだけで漢字と半角英数字の切替えが可能である。

設定をする。このとき Firefox コミュニティー エディションの最初の起動では Vine Linux のホームページに接続されない。プロキシ設定後に、再起動する。Firefox コミュニティー エディションを起動しても、大学内であれば接続されない(接続に失敗する)が、ファイル→URL を開く から `https://**.***.*.**` (**の数字は管理者に問い合わせる) のアドレスで認証を受ける。認証後ホームページが表示されるに違いない。

再起動と終了はメニューバーにあるディスプレイのアイコン GNOME 端末 (コマンド・ライン端末) から

```
[root@localhost root]#shutdown -r now (再起動のとき)
```

と入力する。また、

```
[root@localhost root]#shutdown -h now (終了のとき)
```

と入力すると電源が切れる。または、メニューバーのデスクトップ→シャットダウン からでもよい。

ここで root と su(super user) の入力における使いわけを示すと、まず最初に、全権限を持った root を設定しなければならない(形式的には root の pw を決めて登録する)。その後 root が haruyama のような登録されたユーザー名を持つユーザーの login を許可する。したがって、たとえユーザーが一人であっても Linux に login できるのは、root とユーザー (haruyama) の 2 者になる。Vine Linux ではメニューバーの GNOME 端末 (コマンド・ライン端末) を開く。この端末の画面上でコマンド入力をする。もし、root で login したなら、コマンド入力は # に続く。一方、ユーザー名での login では # ではなく \$ に変わる。これがコマンド入力待の状態である。

```
[root@localhost root]#
```

```
[haruyama@localhost haruyama]$
```

ユーザーで login していて、設定を変更したり、消去しなければならなくなったときには

```
[haruyama@localhost haruyama]$ su
```

と入力すると、root の pw の入力が促される。

パスワード (P):

これで一時的に root 権限を持つことができる。

```
[root@localhost haruyama]#
```

ここで必要な作業を行ってから、再度ユーザーに戻るには、

```
[root@localhost haruyama]# exit
```

と入力する。

6.4 バージョンアップ

コンピュータが外部に接続できることを確認してから、メニューバーの GNOME 端末 (コマンド・ライン端末) を開く。Root 権限のもとに、ユーザーの home directory /haruyama から

```
[root@localhost haruyama]# cd /tmp
```

に移り、作業をする。

```
[root@localhost tmp]# apt-get update
```

と入力すると、インターネットを通じ Vinelinux.org に接続され、自動的にアップデートの有無が確認される。一連の確認作業の後、アップデートがあるときには、コマンドの入力待ちになってから

```
[root@localhost tmp]# apt-get upgrade
```

と入力する。ダウンロードとインストールに要するファイル数と必要メモリー数が示され確認が求められる。入力待ちの において、大文字の を入力する。あとは数分から数 10 分の作業が自動的に始まる。バージョン 4.0 からのバージョンアップなら 200 個程度のファイルのダウンロードが行われるであろう。途中で全体の何パーセントの取得がされたかが示される。ダウンロードが済むと自動的にファイルの 作業に移る。これも途中の経過が ##### のマークと % で示される。終了終盤での適用を終えると、背景が青空からセピア調の黄色の多くの花びらに突然変わるであろう。適用が済むと、Vine Linux 4.2 にアップデートされている。作業が終了後はユーザー名に戻れば、再起動の必要もなく更新されたソフトウェアが使えるようになっている。

ここでのアップデートとアップグレードは月に一度程度行い、Vine Linux を常に最新の状態に保っておくのがよい。

6.5 各種アプリケーション ソフトウェアのインストール

メーラーとして Sylpheed がデフォルトで使えるにも係わらず、電子メール設定について何も述べてこなかった。メーラーの Thunderbird は大変使い易いのでこれをインストールすることにする。その他にも、メニューバーの「アプリケーション」をクリックしてもインストールされているソフトウェアが少ないことに気がつくであろう。Thunderbird を初めとして、各種のアプリケーション ソフトウェアもインターネットを介してインストールできる。その入手方法を示す。

メニューバーの「デスクトップ」→「システム管理」→「Synaptic パッケージマネージャ」を開く、root の pw を入力する。インターネットに繋がっていることを確認してから Synaptic メニューバーの「再読込」をクリックする。再読込を終えると、項目の見出しが増えていくことに気がつく。そこで例えば、インターネットの項目から Thunderbird コミュニティー エディションに関する項目を選び、項目の最初に記されている★印が小さい四角形の右上に囲まれた箇所をクリックすると、「インストール指定する」の見出しが出るのでこれをクリックしてインストール指定する。他のアプリケーションソフトウェアに関してもこの種の作業を繰り返し、Open Office, Maxima, Xmaxima, ggv 等をインストール指定する。Maxima と Xmaxima の指定をすると gnuplot 他が関連指定される。必要なソフトウェアをインストール指定(項目の全体が緑色に変わる)してから、Synaptic メニューの「適用」をクリックする。これで指定したソフトウェアのダウンロードが実行される。ダウンロードが終わるとインストール 適用作業が連続して行われる。最後に、いくつかのファイルを適用する際にエラーがあった旨表示される。これには再ダウンロード、再インストール以外の対処法はないし、それでうまくいくとは限らないので、時にはそのまま我慢するしかない?

全ての作業を終えてから、Synaptic を終了させて、メニューバーの「アプリケーション」を開いてみると、インターネットの項目に Thunderbird コミュニティー エディション がオフィスの項目にデータベース開発、表計算、図形描画、プレゼンテーション、数式エディタ、ワープロが加わっているであろう。Xmaxima はコマンド・ライン端末で開くのでアプリケーションには表示されていない。

メーラーの Thunderbird コミュニティー エディションを開き、Thunderbird コミュニティー エディションのメニューバーから「編集」→「アカウントの設定」に必要事項を書き込む。Firefox コミュニティー エディションは Netscape Navigator と Thunderbird コミュニティー エディションは Mozilla mail と密接に関連していることに気がつくであろう。

6.6 プリンターの設定

プリンターの設定も以前に比べて簡単に行えるようになっている。プリンターが USB ケーブル等でコンピュータに直接に単独に接続されているときは、メニューバーの「デスクトップ」→「システム管理」→「CUPS(Common Unix Printing System) 印刷マネージャ」を開く。ここから Windows のプリンター設定画面と同じ様に設定できる。

PS-プリンターがネットワーク接続されている場合には、インターネットブラウザ Firefox コミュニティー エディションから **ファイル** → **URL を開く** で

`http://localhost:631/`

と場所を指定する。

管理 **プリンタクラスの管理** **オンラインヘルプ** 等々の画面が表示される。ここの **管理** をクリックし、ユーザ名: root と、パスワード:root の pw を入れる。**OK** をクリックしてから **Admin** → **プリンタ** → **プリンタ追加** に画面が変わっていく。ここの名前の欄に各自が名前をつけたプリンタ名を書く (下の例では mycolorprn)。プリンターの登録作業が終了したなら、すぐにこのプリンタ名とその IP-address を root で login し、/etc の中にあるファイルの hosts に emacs や vi で書き加え登録する。hosts の変更例をここに記す。

```
127.0.0.1 localhost localhost.localdomain
```

```
10.15*.???.++ mycolorprn    (*, ?, +には各自のプリンターのアドレスが入る)
```

ここではプリンタ名を mycolorprn と上と同じに仮りの名前にしてある。

話をプリンターの設定に戻す。名前の欄の下にある **場所** と **説明** の欄に書く、書かないは任意である。**次へ** をクリックして

デバイス: LPD/LPR host or printer

lpd://mycolorprn/queue (mycolorprn は仮のプリンター名)

メーカー:postscript (ここに具体的なメーカー名は入れない)

モデル:generic postscript printer (ここにも具体的なモデル名は入れない)

を入力の後、**次へ** をクリックし

メーカー:???????/Postscript (具体的なメーカーとモデル名を探す)

を入力する。この設定でプリンターが使えるようになる。

最後に、印刷作業の途中に、プリンターの電源が入っていなかったとかの理由でプリンターに送った Job を取り消す必要性が生じたときには、su になって

```
[root@localhost haruyama]#lpq (出力待の job number の数字???を記憶する。)
```

```
[root@localhost haruyama]#lprm ??? (???は取り消す job number)
```

とすることで取り消すことができる。

7 付録2. 各種ソフトウェアの使いかた

7.1 L^AT_EX の使いかた

メニューバーの GNOME 端末 (コマンド・ライン端末) を開いて

```
[haruyama@localhost haruyama]$ emacs example1.tex &
```

と拡張子 (tex) をつけて入力すると, 自動的に野鳥 (yatex) が⁵ エディタの emacs の中に組み込まれる⁽⁶⁾。なお, ここでは L^AT_EX のソース ファイルを, example1.tex とした。また, 行末の & は編集作業をバックグラウンドで処理することを意味する。さて, L^AT_EX や T_EX では, コマンドや各種の括弧を入れ子の状態で定義しなければならない。数式は通貨記号 \$ で挟む。数式の中で左 (にはこの括弧を閉じる右) が入れ子で直近に対で入る。あるいは, 数式番号つきの数式は

```
begin{equation}
```

で始め, この行の下に数式を書き,

```
end{equation}
```

で終わる。数式の番号は自動的に付与される。このとき野鳥は対の end 文の入力を自動化してくれる。

次に, 文字や数式の入力を終えて保存してから, このソース ファイルをコンパイルして拡張子 DVI のついた デバイス ファイルに変換する。これには新たにメニューバーの GNOME 端末 (コマンド・ライン端末) を開いて, この端末から

```
[haruyama@localhost haruyama]$ platex example1.tex
```

と入力するだけですむ。通常はコンパイル中にエラーが生じ, エラーがなくなるまで, ソース ファイルを修正しなければならない。2つの端末が使えるからエラーの説明を見ながら修正作業ができることになる。エラーがなくなれば, デバイス ファイルを Postscript(PS) file や PDF(portable document format) file に変換してディスプレイ上に表示する。2つめに開いたコンパイル作業をした端末で

```
[haruyama@localhost haruyama]$ dvips -o example1.ps example1.dvi (Postscript file へ)
```

```
[haruyama@localhost haruyama]$ dvi2pdf -q example1.dvi example1.pdf (PDF file へ)
```

と入力する (background 処理ができないので行の末尾に & をつけない)。このときのファイルの書く順序が PS と PDF で異なることに注意する。あとは, これらを, それぞれの読み取り専用のソフトウェアで読む。例えば, PS file なら同じ端末から

```
[haruyama@localhost haruyama]$ ghostview example1.ps &
```

とする。PDF file の読み取り専用のソフトウェアは Acrobat Reader 他たくさんあるが、ここでは Synaptic からダウンロードしておいた ggv ファイルを使って

```
[haruyama@localhost haruyama]$ ggv example1.pdf &
```

と入力する (これらは & で終わる)。エラーが無いけれども思いどおりの出力になっていなければ、最初の編集作業用の端末を表示して、2 つを比較しながら修正が可能になる。

7.2 C++ のコンパイルと出力

GNOME 端末を開いて、エディタの emacs で C や C++ のソースプログラムを書き、セーブする。このとき拡張子を .c とするか .cpp とするかでコンパイラが C か C++ かが決まる⁽⁷⁾。例えば、プログラム名を testc-1.cpp とつけて /haruyama のディレクトリにセーブする。GNOME 端末からコンパイルは

```
[haruyama@localhost haruyama]$ g++ -o testc-1 testc-1.cpp
```

を入力する。エラーがなければ

```
[haruyama@localhost haruyama]$ ./testc-1
```

と出力ファイル testc-1 を実行すれば、結果が標準出力のディスプレイに書かれる。

7.3 Java のコンパイルと出力

GNOME の端末を開いて、エディタの emacs で Java のソースプログラムを書き、セーブする。拡張子は .java とする⁽⁸⁾。例えば、プログラム名を Testjava-1.java とつけて /haruyama のディレクトリにセーブする。Java は大文字と小文字が厳密に区別される。GNOME 端末からコンパイルは

```
[haruyama@localhost haruyama]$ javac Testjava-1.java
```

を入力する。エラーがなければ、結果は専用のビューワー appletviewer を使って見る方法

と HTML に変換して、ブラウザ (Internet Explorer や Firefox) で見る方法の 2 通りがある。Appletviewer を使うときは

```
[haruyama@localhost haruyama]$ appletviewer Testjava-1.java
```

とすれば、専用の画面に結果が表示される。

7.4 Maxima と描画のエクスポート

メニューバーの GNOME 端末 (コマンド・ライン端末) から

```
[haruyama@localhost haruyama]$ xmaxima &
```

と入力すると、Maxima が開始される⁽²⁾。通常は画面の下半分に実際の計算例が示されているので、これを参考に自学自習ができる。例えば、

```
(%i1) float(%pi);
```

と上半面に入力 (; をつけてから Enter) すれば、

```
(%o1) 3.141592653589793
```

とすぐに答えがでる。

図形描画の場合には、デフォルトではディスプレイ上に描かれるだけで、 $\text{\LaTeX}$ の文章中に取り込んだり印刷したりできない。例えば、4 章の 球ベッセル関数の $j_0(x)$ と $j_1(x)$ を同一のグラフ面に表示するには

```
(%i2) j0 : sin(x)/x$
```

```
(%i3) j1 : sin(x)/x/x - cos(x)/x$
```

```
(%i4) y : 0$
```

```
(%i5) plot3d([j0, j1, y], [x, 0, 20], [y, -2, 2]);
```

を入力すると、図 3 が描かれる⁴。この図形を EPS ファイルにして保存するには、2 つの方法がある。最初の方法は Xmaxima のメニューバーの options → plot windows → embedded

⁴Windows 版の Maxima(xMaxima や wxMaxima) では 3 次元グラフの表示では、view point の位置をマウス操作だけで自由に変更できる。Maxima は一般に Windows 版が優れている。

にチェックマークをつけてから 3 次元 plot を再実行する。こうするとユーザーのホーム・ディレクトリー (この設定では /home/haruyama) に maxout.gnuplot のファイルが作られる。

新らしく, GNOME 端末 (コマンド・ライン端末) を開き, ここから

```
[haruyama@localhost haruyama]$ gnuplot
```

を入力して, gnuplot を起動する。このとき gnuplot は background 処理ができないので, 行末に & をつけない。Gnuplot の起動後

```
gnuplot>load "maxout.gnuplot"
```

と入力すれば, gnuplot に maxout.gnuplot 名のファイルが取り込まれる。ここでも terminal は通常はディスプレイのままである。このファイルを eps-file に変換して, L^AT_EX の文章中に取り込んだり ghostview で表示できるように変えてみる。

```
gnuplot>set terminal epslatex color      (カラー表示もできるようにして)
gnuplot>set output "test-bessel.eps"    (ファイル名を test-bessel.eps とした)
gnuplot>load "maxout.gnuplot"
```

3 つめの GNOME 端末 (コマンド・ライン端末) を開き

```
[haruyama@localhost haruyama]$ghostview test-bessel.eps &
```

で確認する。Gnuplot の開いている, 2 つめの GNOME 端末 (コマンド・ライン端末) にもどり, terminal を標準のディスプレイにもどしてから gnuplot を終了する。これには

```
gnuplot>set terminal x11
gnuplot>quit
```

とする。ここで, x11 は標準コンソールであるディスプレイを指す。

もう 1 つは, Xmaxima の 入力文の中に eps ファイルへの変換コマンドをつけてから実行する。上の球ベッセル関数の例では (%i5) を

```
(%i5) plot3d([j0,j1,y],[x,0,20],[y,-2,2],[gnuplot_term,ps],[gnuplot_out_file,"test-bessel.eps"]);
```

と入力すれば, /haruyama に同じく eps ファイルができる。

参考文献

ここで引用した商品名は各社の登録商標および商標です。なお、論文中には登録商標・商標のマークは使っていません。

- (1) Particle data group, *Review of Particle Physics*, **33**(2006).
- (2) 横田 博史, はじめての Maxima, 工学社 (2006).
- (3) ガシヨロウイツ, 量子力学 I, 丸善 (株), (1998).
- (4) 中西 隆, UNIX 活用ハンドブック, 技術評論社 (1994).
- (5) 羽山 博, Vine Linux 4.0 スタータキット, 毎日コミュニケーションズ (2006).
- (6) 生田 誠三, $\LaTeX 2_{\epsilon}$ 文典, 朝倉書店 (2000).
- (7) トム・スワン, GNU C++ プログラミング, 翔泳社 (2000).
- (8) アーサー・バン・ホフ, サミ・シャイオ, オルカ・スターバック, かんたん JAVA, アジソン・ウエスレイ・パブリッシャーズ・ジャパン (株)(1996).