読むと少しだけためになる十の記事 *

渡部 善隆 †

本稿は、九州大学大型計算機センターを利用するに際して、知っていると多少の得をする情報を集めたものです。これらの情報は、過去の広報記事や、利用の手引のどこかに何らかの形で掲載されています。しかし、情報を体系づけてまとめた（例えば利用者の質問に答える Q&A 的な）記事での援護射撃が少なかったため、センター利用者から忘れられる寸前のものばかりです。そこで、それらの中から、利用者が「困った、誰かに相談したいな」と思いそうな事柄を無作為に十項目選び、まとめてみました。実は、この原稿はもともとセンターの内部用資料として、受付などに置くつもりで書き始めました。そのため、困ったことが生じて受付に相談に突進される利用者の質問への即興的な対応の性格が残っている反面、その場しのぎの、深みのない内容になっています。

あげられた項目は、内容が多分一年後には全面的に刷新されそうなものばかりです。従って、ほとんど同じネタで一年毎に記事が出せるのではないかという、せこい思惑もあります。記事の表現は、これまでのセンター刊行物の文体の伝統に大きく逆らって、しかも、"計算機科学の分野でしか通用しない訳のわからない専門用語はできるだけ使わない"ことに努めました。センター側にとっては、計算機は大事な「食いぶち」ですが、利用者の立場からみれば、計算機ごときは単なる研究の道具の一つにすぎません。記事を適当に斜め読みでもされて、もし一つでも有益な情報を得て頂ければ、本稿の役割は本懐を遂げます。そうして、すぐに日々の忙しさに紛れ、皆さんから忘れられていくと思います。

目次

---

*1993 年 11 月 25 日受理

†九州大学大型計算機センター・研究開発部

1 Manual Guide

この記事を読まれているセンター利用者で、実はセンターがニュースと広報以外の刊行物として、『利用の手引』を発行していることを知らなかったという人は、多分たくさんいると思います。センター開設以来の伝統を誇る MSP の世界では、『利用の手引』など必要としない計算機の達人がたくさんいらっしゃいます。昔ある研究室で先輩が作ったプログラムが、その人が研究室からいなくなった後も重宝され、ついには誰も中身が理解できないまま、しかも使用方法も口移しで代々継承されていくとも、噂では聞きます。

しかし、ここでは僭越ながら、九州大学大型計算機センターで発行している利用の手引、広報記事を中心に、センター利用のためのマニュアル類を目的別に分類してみました。なお、刊行時期が古い記事は、レベルアップなどで内容が一部変更になっている場合もあります。機能の追加 / 削除などは随時センターニュースでお知らせしています。配布されたニュースをすぐゴミ箱に捨てずに、たまには目を通すことをお勧めします。項目にあげたマニュアルは、最低限のものですが、これらのマニュアルを読まれれば、関連の参考文献が「芋蔓（いもずる）式」にわかるものを選んでいます。

【Fortran 使用法】

センターの Fortran は富士通製の Fortran 77/EX という製品です。 Fortran 77/EX の特徴は、文法チェックが厳しいことです。他の計算機にのっている Fortran では問題なく動作していたプログラムが、 Fortran 77/EX で走らせようとするとうまく実行してくれない、という質問がプログラム相談によくあります。この場合、調べてみると利用者のプログラムの誤りだったというケースがほとんどです。 Fortran 77/EX の出力するエラーメッセージを参考に、プログラムをじっくり再検討されれば、必ずバグが見つかるはずです。

- 「MSP/FORTRAN 利用法」 畑 三千代，渡部 善隆，国宗 真，肥田木 直子
  九州大学大型計算機センター広報, Vol.26, No.3, pp232–293, 1993.
  : MSP 上で Fortran を実行するための概説です。

- 「MSP/FORTRAN を VP で実行するには」肥田木 直子，渡部 善隆
  九州大学大型計算機センター広報, Vol.26, No.4, pp397–451, 1993.
  : ベクトルプロセッサ (VP) とは、普段対話形式に処理を行なっている計算機の背後に隠れた、高速大容量の計算を得意とする計算機です。 VP に処理を依頼するには、 MSP も UXP も所定の手続きが必要です。この記事は、 MSP の Fortran プログラムを VP で処理するための方法を解説しています。大規模な数値計算が必要な人は、これを読まねばなりません。

- 「UXP/M for the Working Scientists」佐藤 周行, 1993.
  : UXP の総合入門書です。この中から、言語処理関係を参照下さい。

- 「Unix 環境でスーパーコンピューティングをする人のためのガイド」佐藤 周行
  九州大学大型計算機センター広報, Vol.26, No.4, pp452–484, 1993.
  : ベクトルプロセッサの UXP 上での Fortran, C プログラムの実行方法の解説です。

上記文献は、 Fortran 言語そのもののプログラミングについては一切解説していません。 Fortran 言語は、市販の参考書で優れたものが出回っていますので、それらの中から自分に適したものをお読み下さい。ここでは参考までに 2 つの文献をあげます。

- 「FORTRAN 77 入門（改訂版）」 浦 昭二，培風館， 1990.
  : Fortran 入門書として定評があります。

- 「FACOM ユーザのための TSS による FORTRAN77 入門」 山崎 光悦 他，
  サイエンス社， 1986.
  : 表題の通りです。 MSP の入門書としても使えます。

【C 使用法】

Fortran の項であげた UXP 関係の文献 2 つを参照下さい。 MSP の C は、サポートしてはいますが、色々な理由から、使われないほうが無難です。

【センター利用に関する基礎的な知識】

- 「利用の手引・基本編」 九州大学大型計算機センター， 1993.
  : 内容は本当に基本的なことです。これを読んだからといって、すぐに計算機が使えるようにはなりません。利用案内 / 規則が詳述されています。

【MSP 入門】

- 「利用の手引・ TSS 編」 九州大学大型計算機センター， 1993.
  : MSP に関しての実質的な基本編です。 MSP 利用者必読。

- 「利用の手引・ MSP コマンド編」 九州大学大型計算機センター， 1993.
  : MSP コマンドを全て網羅したリファレンス・マニュアルです。 MSP 利用者必携。

- 「利用の手引・バッチジョブ編」 九州大学大型計算機センター， 1992.
  : 大規模ジョブ、 MT(磁気テープ) の実行に必要なバッチ処理の解説です。

- 「利用の手引・ネットワーク編」九州大学大型計算機センター, 1993.
  : MSP のネットワーク機能についての解説です。

【UXP 入門】

- 「UXP/M for the Working Scientists」佐藤 周行, 1993.
  : UXP に関する総合ガイドです。 UXP 利用者必読ですが、実はまだ手引として印刷されていません。しばらくお待ち下さい。

待ち切れない人のために、上記マニュアルは UXP の`/usr/local/doc/uxpguide.dvi` に TEX の dvi ファイルとしておいてあります。従ってセンター 2 階の PostScript printer に出力可能です。しかしながら、総ページが 200 pages 近いガイドなので、 PostScript printer へのプリントアウトの際は、一人が長時間プリンターを占領することになりますので、何回かに分割した出力をお願いします。また、 dvi ファイルをコピーして自分の研究室のプリンターで出力されてももちろん結構です。 PostScript printer への出力は次のように行ないます。

```
kyu-cc% more /usr/local/doc/uxpguide.toc  ⏎     <--- まず目次の検索
```

必要なページを確認後、 PostScript printer にページを指定して出力する例です。

```
kyu-cc% dvi2ps -f 24 -t 34 /usr/local/doc/uxpguide | lp -dps -Tps   ↵
```

“-f 24 -t 34” は “Form page 24 to page 34” の意味です。従って、24 page から 34 page までが印刷されます。出力の際は、ページ数指定のオプションだけを変更下さい。

【ライブラリガイド】

現在センターでサポートしているライブラリのガイドとして次の手引が 1994 年の春に刊行予定です。この手引は、申し訳ありませんが /usr/local/doc 下においてありません。刊行までお待ち下さい。

- 「利用の手引・ライブラリ編」九州大学大型計算機センター, to appear.

【データベース】

- 「オンライン・データベース利用ガイド」全国共同利用大型計算機センター，ライブラリ・データベース連絡会, 1993.
  : 全国共同利用の大型計算機センターおよび学術情報センターのデータベース・サービスの概要と簡単な利用方法の解説です。
- 「利用の手引き・情報検索システム AIR —AIR を用いた文献情報の検索 —」
  九州大学大型計算機センター, 1993.

AIR 利用の手引は /usr/local/doc/airguide.dvi に TEX の dvi ファイルがあります。PostScript printer への出力方法は、「UXP/M for the Working Scientists」と同様です。

【電子メール】

- 「UXP における電子メール入門」 九州大学大型計算機センター, 1993.

この利用の手引も、広報記事などの形で表に出ていませんが、UXP 上での電子メールの利用方法を詳しく解説した記事です。1993 年度の講習会資料として作成されました。
/usr/local/doc/mailguide.dvi に TEX の dvi ファイルがあります。PostScript printer への出力方法は、「UXP/M for the Working Scientists」と同様です。繰り返しになりますが、もう一度出力方法を示します。出力される場所は、センター 2 階の PostScript printer です。出力結果は自分で取りに来る必要がありますので、必ずセンター内で出力命令を与えて下さい。「UXP における電子メール入門」は約 70 pages です。できるだけ分割して出力することをお願いします。印刷状況の確認は lpstat コマンドです。

```
kyu-cc% dvi2ps -f 1 -t 20 /usr/local/doc/mailguide | lp -dps -Tps   ↵
        :
要求ＩＤは ps-1957（標準入力）です
kyu-cc% lpstat  ↵                           <--- 印刷状況の確認
ps-1957                 a79999a          37238   Nov 30 20:37 ps 上で印刷中
```

これにより「UXP における電子メール入門」の 1 ページから 20 ページまでが印刷されます。

【PFD, PFDE】

- 「PFD と PFDE の使用法」平野 広幸，九州大学大型計算機センター広報, Vol.25, No.2, p119–165, 1992.
  : PFD, PFDE は MSP のデータセット管理の上で欠かすことが出来ません。 MSP 利用者は必ずお読み下さい。

【オンライン・マニュアル】

以下に、端末の画面から検索可能なものを紹介します。なお、 UXP と MSP とは全く別物です。 UXP で `PLUM` と打ち込んでも、 MSP で `man` と打っても無駄です。ご注意下さい。

- 富士通株式会社提供の MSP コマンドは `HELP` コマンドで機能を検索できます。`HELP` とだけ入力すれば、参照可能なコマンド一覧が表示されます。

- UXP のコマンドも `man` コマンドで使用方法を検索できます。`man` コマンド自体の使用方法は `man man` でお調べ下さい。

- MSP では `JWD`, `JWE` で始まる Fortran 発行のメッセージおよびシステムメッセージを `LM` コマンドで調べることができます。使用方法は先ほどの `HELP` コマンドを早速使って `HELP LM` でお調べ下さい。

- 過去のセンターニュース、広報記事の表題の検索を MSP, UXP ともに `koho` コマンドで行なえます。使用方法はとりあえず `koho` と打って下さい。すぐにわかります。

- MSP では富士通製マニュアル「OS IV/MSP FORTRAN 77 EX 使用手引書 V12 用」を端末画面から検索可能です。検索は `PLUM` コマンドで行ないます。使用方法は、`PLUM` と打てば、使い方自身が画面から調べられます。ただし、`PLUM` は、センターへのアクセス方法によっては利用出来ない場合がありますので、ご注意下さい。

その他、センターで開催される講習会の資料には、その時点での最新情報が満載されています。講習会を忙しくて受講できない場合でも、資料のみの請求も可能です。また、講習会資料で残部のあるものは、センター 5 階受付で配布しています。

『利用の手引』はどうすれば入手できるか

九州大学大型計算機センターに近い利用者は、直接センターに来訪下さい。建物 5 階の受付で申し込めば入手可能です。センターから遠い利用者の方は、連絡所経由で希望の手引を申し込み下さい。なぜ連絡所経由にするかといえば、実は、 1 対 1 の申し込みに応じて個別対応で郵送を行なうと、郵便代が膨大になるからです。何を隠そう、世界有数の計算機を抱えるくせに、センターは貧乏なのです。そのため、連絡所単位でまとめて申し込んで頂ければ、まとめて発送でき、郵便代が節約できる訳です。ご協力をお願い致します。

くどいですが、再度のお願いです。

$\boxed{\text{\TeX で利用の手引を印刷するときは、分割出力をお願いします。}}$

2　パスワードについて

センターの受付やプログラム相談室には、パスワードに関する質問がかなり多く寄せられます。パスワードは利用者が管理すべき基本的かつ最も重要なものです。まず、次のあたり前の注意をさせていただきます。

パスワードは自分でしっかり管理して下さい。

さて、パスワードの問題を話すと、すぐに“セキュリティ”の話になって、文章がかたくなります。そこで、ここではQ&Aの形式で、よくあるパスワードの質問についてまとめてみます。内容は極めて大事なことですから、しっかりお読み下さい。

なお“答”の部分は、プログラム相談員の皆さんの口調の最大値（一番ぶっきらぼう）にあわせてみました。

計算機利用申請を完了したばかりです。MSPにLOGONする時のパスワードがわからないんですが。

【答】新規に登録をした時にセンターから送られる「計算機利用承認書」をよく読んでいませんね。申請時に、あなたには初期パスワードが設定されています。その申請書から、あなたに割り当てられた初期パスワードがわかるはずです。ただし、そのパスワードは「計算機利用承認書」を見れば、誰でもすぐにパスワードが分かる形式になっていますので、あなたと反りの合わない研究者や、単なる暇人が、承認書を読んで、あなたの課題に入って荒し回る危険があります。すぐに自分にしか分からない固有のパスワードに変更して下さい。

パスワードの変更方法がわからないんですが。

【答】『利用の手引・基本編』をよく読んでいませんね。MSPのパスワード変更は`PASSWORD`コマンド、UXPは`passwd`コマンドです。使用方法は、コマンドを叩いてみればわかります。パスワードは、8文字以内の英数字の文字列にして下さい。ただし、最初は英字です。短いものや、類推のしやすいものは避けて下さい。安直なパスワードを入れておくと、あなたと反りの合わない研究者や、単なる暇人が、あなたのパスワードを見破り、課題に入って荒し回る危険があります。

MSPから`SINSEI`コマンドでUXPの利用申請をしました。もう一度`SINSEI`コマンドを使用して、UXPの登録手続きの完了を確認しました。しかし、パスワードを入れてもUXPに入れません。どうしたらいいでしょう。

【答】UXPへ入る時に、初期パスワードから変更したMSP側のパスワードを入力しているのではないでしょうか。UXPの初期パスワードは、MSPの初期パスワード、即ち「計算機利用承認書」に記載された形式のパスワードと同じです。あなたは当然、MSP側のパスワードを初期値から変更している筈ですね。しかし、変更後のMSPのパスワードはUXPの申請

時の初期パスワードと一致しません。UXP の初期パスワードは、最初あなたに「計算機利用承認書」で割り当てられた初期パスワードです。

「計算機利用承認書」を再度読みかえし、初期パスワードを確認下さい。また、UXP のパスワードもすぐに passwd コマンドを用いて固有のパスワードに変更下さい。さもないと、あなたと反りの合わない研究者や、単なる暇人が、あなたの課題に入って荒し回る危険があります。

ここで注意をさせていただきます。

『計算機利用承認書』は捨てないで大切に保管下さい。

質問の続きです。

先ほどの忠告を守り、難しいパスワードを考えたのですが、しばらく計算機を使っていなかったので、自分で設定したパスワードを忘れてしまいました。申し訳ありませんが、どうしたらいいでしょう。

【答】う~ん、困った人ですね。思い出せないとすれば、利用者レベルではどうしようもありません。登録番号と本人、つまり貴方が本当に課題を承認された人であることを証明するものを持って、センター 5 階の共同利用掛、またはセンター 2 階の受付に相談下さい。たいへん失礼ですが、パスワードを知ろうとする、又は変更を要求されるあなたと、あなたのお持ちの課題とが一致することの確認をとっておかないといけません。さもないと、あなたと反りの合わない研究者や、単なる暇人が、あなたの名前をかたってパスワードを変更した上、課題に侵入し、荒し回る危険があります。

どうもありがとうございました。

【答】いえいえ。

なお、質問 / 答の会話は、プログラム相談員が男性で、相談に来られた方も男性の場合を想定しています。もちろん、プログラム相談員が男性で、相談に訪れた人が女性の場合は、"答"の口調は大きく変化します。逆も真です。

3　セッションのログを取る

端末の前に座って計算機とやりとりをしていると、端末に表示される情報をファイルに落としたい場合があります。UXP, MSP とも、打ち込んだコマンドや、それに対するシステムからの応答メッセージ等の入出力データを、ログ情報として保存することが出来ます。コマンドは UXP が `script`、MSP が `TRACE` です。

TRACE コマンド

MSP の端末情報の保存は、`TRACE` コマンドで行ないます。まず、大切な注意点です。

MSP の端末情報の保存は、必ず `TRACE OFF` で終了して下さい。

それでは、具体的な使用例に沿って解説します。Fortran プログラムを対話的に処理していますが、翻訳メッセージや実行結果が、どうも納得いきません。そこで、端末に表示される Fortran プログラムの翻訳 / 実行結果を、データセット `TRACE.DATA` に保存して、じっくり検討することにしました。データセット `TRACE.DATA` が存在しない場合は新規に作成します。`TRACE.DATA` が存在する場合、`MOD` オプションを指定しない限り、上書されます。

```
  READY
 TRACE ON DA(TRACE.DATA)   ↵       <--- データ採取開始
  READY
 FORT TEST.FORT   ↵                <--- Fortran プログラムの実行
   【FORTRAN77 EX 翻訳開始】
FORTRAN77 EX 診断メッセージ：プログラム名(MAIN),FLAG(I),OPTIMIZE(B)
JWD2006I-I                    C は宣言だけされていて実行文で引用されていません
JWD2006I-I                    E は宣言だけされていて実行文で引用されていません
                         :
          ( この間の端末入出力情報が記録 )
                        :
  READY
 TRACE OFF                           <--- データ収集を終了
  READY
```

同じデータセットに追加書きしたい場合は、次のように`MOD` オプションを 指定します。

```
 READY
TRACE ON DA(TRACE.DATA) MOD   ↵
```

次のように指定すると、その間のログ情報がセンターのラインプリンターへ出力されます。

```
 READY
TRACE ON SY(U)   ↵        <--- NLP, CLP への出力要求
   :
TRACE OFF   ↵             <--- 終了時は TRACE OFF を必ず指定
```

`TRACE` コマンドは必ず `TRACE OFF` を入力して終了して下さい。指定がない場合、データの収集は行われません。更に、`TRACE OFF` を指定して終了しないと、次セッションでは `TRACE` コマンドが正常に起動できなくなることがあります。また、ディスプレイ端末のフルスクリーンデータ（PFD の画面など）の情報は収集されません。

script コマンド

UXP の端末入出力情報は script コマンドで取ります。script コマンドは、画面に現れたすべての情報をファイルに格納します。データ収集の終了は exit、または CTRL-D（Ctrl キーを押しながら大文字の D を入力）です。このとき、script コマンドを起動していないにも関わらず exit を入力すると、exit コマンドは logoff コマンドと同じですので、UXP のセッションが終了しますので、ご注意下さい。

例として、script コマンドによって、端末に表示される情報をファイル trace.dat に保存します。ファイルが存在しない場合は新規に trace.dat を作成します。ファイル名が指定されていない場合、typescript というファイルに勝手に保存されます。

```
 kyu-cc% script trace.dat   ↵               <--- script コマンド
 Script started, file is trace.dat
 kyu-cc% dvi2ps -f 166 -t 166 soft > soft.ps  ↵
[/usr/local/lib/dvi2ps/tex.ps]
Prescanning ..........................................................
Reading font info ...............................
            :
   ( この間の端末入出力情報が記録 )
            :
 kyu-cc% exit  ↵                             <--- exit または CTRL-D
 kyu-cc% Script done, file is trace.dat     <--- script コマンドの終了
 kyu-cc%
```

また、既存のファイルに上書きしたい場合は、次のように-a オプションを指定します。

```
 kyu-cc% script -a trace.dat   ↵
 kyu-cc% Script done, file is trace.dat
     :
  ( 端末入出力情報 )
     :
 kyu-cc% exit  ↵
 kyu-cc% Script done, file is trace.dat
 kyu-cc%
```

4 プロセスのキャンセル方法

世の中には恐ろしいことがたくさんあります。多くの恐ろしいことは、対人間との関係において生じます。しかし、暴走するプログラムに対し、その暴走に“待った”をかけられない状況は、暴走によって生じる (おそらく) 良くない結果と、たかが計算機のプログラムひとつ止められない己の無力さに対する自己嫌悪が襲ってくることから、恐ろしいことのひとつに数えていいでしょう。どうしようもない不可抗力から、多分に意図されたものまで、プロセスの暴走や端末のハングアップする状況は、計算機を利用者する人に一回は確実に襲います。ここでは、どうにも対処の方法がない場合の強制的プロセスのキャンセル方法をお教えします。

その前に、注意をひとつ。

パニックになっても、センター内の端末のスイッチは切らないで！

計算機システム自体が死んでしまった場合や、端末のハードウェアそのものが壊れたのでない限り、プログラムの暴走は利用者レベルにおけるプロセス異常にほかなりません。センターの専用端末はパーソナルコンピュータ（パソコン）ではありません。パソコンから電話回線を経由してセンターの計算機に入っている場合、モデムの回線を叩き切ったり、ひどい人はパソコン端末となっているパソコンのスイッチを切ることで窮状を回復しようとします。確かにこのようにやれば、プログラムの暴走は止められますし、場合によっては最も効果的な方法でもあります。しかし、センター内に並んでいる端末の電源を切っても、開設中のセッションはキャンセルされません。また、これもセンター内にある X 端末やワークステーションの電源をいきなり切りにいくのも、絶対に止めて下さい。ファミコンで熱くなると、「頭にきて」リセットをかけてやり直すことがよくありますが、センターの計算機に対しては、気持ちはわかりますが、そのような単純な対処はご遠慮下さい。

とりあえずやってみること

ありがちなこととして、間違ったコマンドを入力した結果、見たこともないプロセスを起動してしまい、何を入力しても入力前の状態に復帰できない状況が考えられます。何かのシステムに入り込んでしまった場合は、そのプログラムに対し、「終了」を意味するコマンドを打ち込めば出られる筈です。従ってそういう場合は、何でもいいですから「終了」を意味するコマンドを打ち込んでみましょう。数打ちゃ当たるです。

```
SGPP-X>                       <--- 変なシステムに迷い込んだぞ困ったな...
SGPP-X> end    ↵              <--- end で脱出だ！
SGPP-X> quit   ↵              <--- quit ではどうだ！
SGPP-X> bye    ↵              <--- bye では！
SGPP-X> finish ↵              <--- これはどうだ！
SGPP-X> exit   ↵              <--- えっと、残るはこれだ！
kyu-cc%                       <--- おお、やっと脱出成功 やれやれ
```

MSP の場合、大文字で入力すると脱出できる場合があります。

それでも駄目な場合は、起動されたプログラムに割り込みをかけて実行を中止させることを考えます。次のようにやってみて下さい。

UXP	MSP
CTRL-c を入力	Attn キーを押す
CTRL-D を入力	Reset キーを押す
CTRL-c を連打	PF3 キーを押す

MSP の場合は UXP と比べ、円熟した（枯れた）ソフトが多いためか、プログラムが暴走することは少ないようです。センター内の MSP 専用端末ならば、Attn キーの数回の入力で、大体プログラムから抜け、READY モードに戻れるはずです。一方、パソコン端末の場合は、Attn キーは普通存在しません。その場合、通信に用いているソフトによって、Attn キーが設定されています。また、他のキーに関しても設定が異なる場合があります。通信ソフトの手引で確認下さい。MSP の場合、READY モードになれば脱出成功です。また、心配性の人のために疑似割り込み（Attn キーもどき）を次のように設定することもできます。例として、文字列"ATTN"に疑似アテンションを設定します。

```
 READY
TERMINAL INPUT(ATTN)   ↵      <--- 疑似割り込みの設定．そのセッションのみ機能
 READY
ATTN   ↵                      <--- ! が出ればOK
 !
 READY
TERMINAL NOINPUT   ↵          <--- 設定を解除する場合に入力
```

UXP の"CTRL-c"、"CTRL-D" は、Ctrl キーを押しながら c または D を入力することを意味します。実際やってみればわかりますが、"CNTL-D"を入力するためには、Ctrl キーを押しながら、さらに Shift キーを押しながら D のキーを押す必要があります。

UXP でのプロセス強制キャンセル – kill コマンド –

UXP でプロセスが思う通りに走らず、"CTRL-D"、"CTRL-c"でもプロンプト（通常は kyu-cc%）が返ってこない場合、最後の手段として、強制的にプロセスのキャンセルを行ないます。コマンドは kill です。

まず、セッションを新たに開設します。一つの端末で複数のセッションが開ける場合は、同じ端末で結構ですが、一つの端末で単一のセッションしか開設できない環境では、現在唯一のセッションが死んでいるのですから、当然別の端末から login します。

殺し屋専用セッションを開設したら、ps コマンドでプロセスの状況を調べ、殺しにいくプロセス ID を確認します。課題番号が a79999a の場合、次のようにします。

```
kyu-cc% ps -u a79999a   ↵
   PID TTY      TIME COMD
 18679 pts/3    0:08 a.out
  3305 pts/3    0:00 xclock
  3304 pts/3    0:01 xbiff
 18482 pts/2    0:00 csh
  9031 pts/3    0:01 csh
 18680 pts/2    0:00 ps
```

“-u a79999a” によって a79999a に関係するプロセス一覧が表示されます。プロセス ps は ps コマンド自身です。殺したいプロセスは、一番上の a.out とします。さて ps コマンドによって、殺したいプロセスが確認できました。ここで、殺したいプロセスの PID をメモします。 a.out のプロセス ID は最初の PID に対応した 18679 です。

次に、kill コマンドでこのプロセスを強制キャンセルします。殺し屋用セッションから次のように入力します。

```
kyu-cc% kill -9 18679   ↵        <--- プロセスの強制キャンセル
kyu-cc%
```

-9 オプションは、強制的なキャンセルの指定です。これでプロセスの強制キャンセルが終了しました。

MSP でのセッション強制キャンセル – CANCEL コマンド –

MSP の場合、 UXP の様にプロセス単位のキャンセルを行なうという洒落たことは出来ません。 MSP ではセッション単位、要するに LOGON していたセッション丸ごとのキャンセルを行います。以前は LOGON TSS CANCEL しか手段がなかったのですが、 MSP も同一の課題で複数のセッションを開設できる“マルチセッション”が出来るようになったことで、機能強化された CANCEL コマンドで対応可能です。

まず UXP と同様に、別のセッションを開きます。センター内の端末ならば、ハングアップした端末を見捨てて、隣の端末から自分の課題に LOGON します。続いて、 UXP の ps コマンドに相当する ST コマンドでセッション ID を確認します。

```
 READY
ST   ↵
 KEQ56197I JOB A79999AX(JOB08541) IS WAITING FOR EXECUTION
 KEQ56211I JOB A79999A#(TSU07771) IS EXECUTING
 KEQ56211I JOB A79999A#(TSU07772) IS EXECUTING ON THIS TERMINAL
 READY
```

最初の A79999AX(JOB08541) は、実行待ちのバッチジョブ (バックグラウンド処理のジョブ) です。もちろん、A79999AX(JOB08541) は、A79999A さんが実行を指示したジョブです。これを殺すことも出来ますが、ここでは目的が違います。

一番下の A79999A#(TSU07772) は“Executing on this terminal”とある通り、ハングアップしたセッションを殺すために開設した現在の殺し屋セッションを意味します。
真中の A79999A#(TSU07771) が殺したいセッションです。殺し屋のあなたは、 UXP の PID と同じく、この A79999A#(TSU07771) をしっかりメモします。

次に、CANCEL コマンドによってキャンセルを行ないます。

```
 READY
CANCEL A79999A#(TSU07771)   ↵            <--- セッションの強制終了
 READY
```

これでセッション・キャンセルが終了しました。隣の端末では、セッションがキャンセルされ、LOGON 可能な状態に戻っているはずです。

殺し屋の任務は完了です。では、殺し屋のセッションから、自身のセッションである A79999A#(TSU07772) を殺しにいくとどうなるでしょう。

```
 READY
CANCEL A79999A#(TSU07772)  ↵
 KEQ56206I JOB A79999A#(TSU07772) NOT CANCELED BECAUSE EXECUTING ON THIS
 TERMINAL
READY
```

自殺は失敗しました。

最後に、

kill、CANCEL は最後の手段です。

例えば、エディタを使ってファイルを編集していた段階で、強制的なキャンセルを行なうと、場合によっては、編集中のデータが失われたりする危険があります。あくまでも kill、CANCEL でのキャンセルは、どうしようもない場合の最後の手段としてお使い下さい。

5 できるだけ楽をして Fortran の実行速度をあげる

センター利用の目的を大きく分けると、

1. Fortran, C を用いた数値計算
2. inspec などのデータベース検索
3. 電子メール、電子ニュースなどの利用
4. その他アプリケーションの利用

となります。この中で計算機の使用時間の消費という観点に立てば、文句なしに Fortran による数値計算の利用が主役です。繁忙期（12 月 ~ 2 月）になると、MSP にはたくさんの Fortran ジョブの依頼が集まります。これら各々は大規模な数値計算を行なう場合が多いため、この時期は依頼したジョブの結果がなかなか得られないこともあります。

センターで Fortran プログラムの処理を行なう利用者の持つ秘かな欲望は、手持ちの計算機環境では手に余る計算の実行です。つまり、

記憶容量をできるだけたくさん取り、しかも高速に実行したい

ことを考えるわけです。次のプログラムを見てみましょう。

```
      program test
      parameter(maxn=101)
      real*8 matrix(maxn,maxn,maxn),vector(maxn)
      integer count,result
         :
```

大規模数値計算で記憶容量の鍵を握るのは、確保する配列の大きさです。上のプログラムでは matrix(maxn,maxn,maxn) の大きさがプログラムの規模を実質決定します。maxn の値を変え、matrix(maxn,maxn,maxn) が確保する記憶容量を表にします。

maxn	memory
64	2MB
128	16MB
256	128MB
512	1024MB

【計算方法】
倍精度変数は１つ 8 byte です。
matrix(maxn,maxn,maxn) は、変数の総数 × 8 byte 領域が必要です。1KB=1024byte, 1MB=1024KB を使えば左の値が求まります。

matrix は３次元倍精度配列なので、maxn が倍になれば matrix(maxn,maxn,maxn) が必要とする記憶容量は 8 倍になります。実際は maxn 自体、あるパラメータの order に依存するでしょう。例えば、差分をとるときの分割数をパラメータとして、この分割数の何乗かに maxn が比例している場合を考えれば、配列の大きさは元のパラメータを少し変えるだけで、すぐに凄い勢いで増加することがわかります。この辺りの苦労は、メモリーの壁につき当たった利用者の方なら「うんうん」と同意して頂ける筈です。

大規模計算はバッチ処理で

MSP で端末から対話的にプログラムの処理を行なう TSS 処理形式では、確保できる領域として 50MB が最大です。それ以上の領域をプログラムが必要とする場合、バッチ処理と呼ばれるバックグラウンドでの処理を行ないます。 UXP の場合はベクトルプロセッサ VP2600/10 上で Fortran, C プログラムを実行させる場合、同じくバッチ処理の手続きが必要になります。

が、バッチ処理はコマンド一発で簡単にできる訳ではありません。それなりの段取りが要ります。 UXP ではバッチリクエストと呼ばれる一連の処理を記したファイルを作成する必要があります。 MSP もジョブ制御文と呼ばれる、これも一連のプログラム処理を記したデータセットを作成します。バッチ処理は、完全に利用者の手を離れバックグラウンドで実行されますので、「入力データと出力データはこれですよ」とか、「このライブラリを参照しますよ」とかの情報をあらかじめ与えておく必要があります。つまり、プログラムの処理をプログラムすることになります。しかし、 Fortran のバッチ処理に関しては、処理環境が十分に整備されていますので、それほど苦労はいりません。具体的な Fortran プログラムのバッチ処理での実行方法は、次の広報記事をお読み下さい。

> 「スゥィート・ベクトル・プロセッサ」九州大学大型計算機センター広報, Vol.26, No.4, pp397–484, 1993.

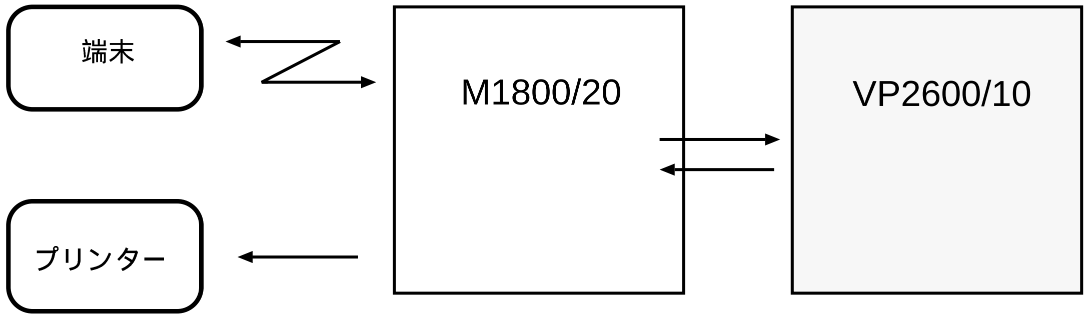
高速な実行は VP で

九州大学大型計算機センターでは、現在 Fujitsu VP2600/10 を導入しています。 VP2600/10 は、通称 VP と呼ばれています。 VP の特徴をあげます。

- 大規模演算を高速に処理するため (だけ) に生まれてきた計算機
- 利用者との対話的な処理が行なえない“はにかみ屋”

VP 上で動作するアプリケーションは数えるほどで、しかも Fortran で書かれたものがほとんどです。 Fortran 言語そのものが、もともと IBM の当時 (1950 年代後半) のスーパーコンピュータ専用の言語として開発されたという経緯があります。 VP2600/10 はその伝統を引き継ぎ、実質 Fortran 処理のために存在する計算機です。逆に VP に魅力的なアプリケーションがあまりのってないということは、 Fortran プログラムの融通のなさの証明でもあります。

2 番目の“はにかみ屋”の性格は、 Fortran 処理に集中できるようにセンター側で VP を利用者から「隔離」したのが原因です。その代わり、 VP の窓口として M1800/20 という汎用機（= なんでも屋）が受付を行なっています。利用者は M1800/20 を通して VP2600/10 に Fortran プログラムの処理を依頼し、直接 VP と口をきくことはできません。 VP2600/10 は M1800/20 から受けとった Fortran プログラムを、高い電気料金を使いながらも黙々と処理し、処理結果を再び M1800/20 に返します。利用者は M1800/20 を介して処理結果を受けとります。



科学技術用プログラムは、配列データを対象とした DO 文の集合が多数見られます。 VP はこうした DO 文に対する、同一作業の繰り返し処理を高速化する機能を持つ計算機です。従って，プログラムが高速処理を可能とする DO 文で大部分が占められていなければ、高速化につながりません．VP には Fortran 77 EX/VP コンパイラというライブラリがのっています。このコンパイラが、利用者の Fortran ソースプログラムを解釈して、出来るだけ DO 文の集合を高速化するように努力します。そのため、利用者は別段 VP に合わせたプログラミングを考えなくてもよいのですが、より高速化を狙うためには、自分で工夫することも必要です。 VP を用いた Fortran プログラムの実行方法は、再掲になりますが、次の記事をお読み下さい。

「スゥィート・ベクトル・プロセッサ」九州大学大型計算機センター広報, Vol.26, No.4, pp397–484, 1993.

VP 用サブルーチンライブラリを利用する

VP 上で Fortran の実行を高速化するには、 Fortran 77 EX/VP コンパイラが、繰り返し処理を高速化する解釈をしてくれることが必要です。経験的に、プログラムの実行時にかかるコストはある限られた部分、例えば、連立一次方程式を解くサブルーチンなどに集中することが知られています。即ち、コストがかかる部分を集中的に高速化すれば、最終的に実行速度が速くなります。

安直、かつ効果的な方法は、 VP 用にチューニングされたサブルーチンライブラリを利用することです。センターでは次のライブラリが利用できます。

- 富士通株式会社提供 ： SSL II/VP
- 名古屋大学大型計算機センター提供 ： NUMPAC/VP

ご自分で書かれたサブルーチンで、上記サブルーチン群と機能が一致するものは、試しに利用されて、性能を比べて下さい。サブルーチン一覧は『利用の手引・ライブラリ編』に掲載予定です。また、マニュアルはセンターのプログラム相談室／図書室で閲覧できます。

最適化レベルを調節する

続いての簡単な高速化方法は、 Fortran プログラムの翻訳時に、最適化のレベルを上げることです。実は、最適化オプションの機能／注意点／指定方法に関しては、かなりしつこくセンターニュースや広報の記事で書きましたので、ご存知の方も多いと思いますので、ここでは繰り返しません。次の文献に機能が詳述してあります。参照下さい。

「MSP/FORTRAN 利用法」 畑 三千代，渡部 善隆，国宗 真，肥田木 直子
九州大学大型計算機センター広報, Vol.26, No.3, pp232–293, 1993.

「Unix 環境でスーパーコンピューティングをする人のためのガイド」佐藤 周行
九州大学大型計算機センター広報, Vol.26, No.4, pp452–484, 1993.

作業用ファイルに SSU を割り当てる

Fortran プログラムの実行時に、一次的な外部ディスクを作成してそこに読み書きを行なっている場合は、外部ディスクの代りに SSU (System Storage Unit) を割り当てると、読み書きが高速化されます。 SSU は、ディスクと主記憶の中間的なものです。しかし、残念なことに SSU の利用は現在 MSP のみに制限されており、 UXP では利用出来ません。また、割り当てた SSU は一時ファイルであり、実行が終了すると強制的に消去されます。あくまで作業用ファイルとしてお使い下さい。使用方法は、以下をお読み下さい。

「MSP/FORTRAN を VP で実行するには」肥田木 直子，渡部 善隆
九州大学大型計算機センター広報, Vol.26, No.4, pp397–451, 1993.

「VP2600 のシステム記憶とその使用法」 島崎 眞昭
九州大学大型計算機センター広報, Vol.25, No.3, pp203–209, 1992.

自分でチューニングする

最終的に VP の性能を最大限に引き出すには、どうしても利用者の Fortran ソースプログラムを VP 向けに改良（チューニング）することが大事です。次の参考文献をお読み下さい。閲覧は、センターのプログラム相談室、オープン室で可能です。

「OS IV/MSP FORTRAN77 EX/VP 使用手引書 V12 用」(79SP-5041)，富士通株式会社.

また、チューニングを支援するツールとして、 Analyzer というライブラリが用意されています。使用方法は UXP, MSP とも「スゥィート・ベクトル・プロセッサ」を御覧下さい。

まとめ

大規模な科学技術計算を高速に実行するための手段です。

1. VP を利用する。実行はバッチ処理であり、所定の手続きが必要。
2. VP 向けにチューニングされた既成のサブルーチンライブラリを使用する。
3. 最適化レベルを変更する。
4. 一時ファイルを使用する場合、 SSU に割り当てる。
5. 最終的には利用者でチューニングを行なう。

6　MSP のバッチ処理結果をデータセットに保存する

意外と知られていないのが、MSP のバッチジョブの処理結果、即ち、MSO, SORP などで検索する出力情報をデータセットに保存する方法です。バックグラウンドで処理された実行結果をプリンターへ出力するだけではもの足らず、データセットに落し、データを加工して使いたいと思われる方のために OUT コマンドによるデータセットへの出力をお教えします。

その前に、次の覚悟をして下さい。

OUT コマンドで出力したジョブはシステムから消去されます

つまり、一度 OUT コマンドを使ってしまえば、利用者のデータセット以外には処理結果は存在しなくなります。従って MSO, SORP, PFD コマンドを用いた検索はできなくなります。また、センターのプリンター前のコンソールからの出力要求リストからも除かれます。ただし、落したデータセットをプリンターに出力すれば同じことができます。

さらに、次の注意も必要です。

出力リストが膨大な場合、出力が失敗する場合があります

出力リストが膨大になる代表的なものが、有限要素法を用いた構造解析ソフト MARC です。MARC の出力リストは、多い場合は MB 単位のリストを出します。これをそのまま OUT コマンドでデータセットに出力しようとすると、データセットの省略値の大きさを越えてしまい、結果が途中で切れてしまう場合があります。何も考えずにどんどんファイルを取りにいく UXP と違って、MSP は資源の管理に厳しいのです。回避方法としては、あらかじめ出力用のデータセットを大きめに作成しておき、そこに OUT コマンドで書き込みをさせれば、うまくいきます。しかし、とんでもなくリストが多くならない限り、心配は要りません。MSP のデータセットの作り方は「利用の手引・TSS 編」を参照下さい。

では、実際にやってみましょう。まず、ST コマンドでバッチジョブの処理状況を調べます。操作を行なっている人は A79999A さんとします。

```
 READY
ST   ↵
 KEQ56211I JOB A79999AW(JOB01084) IS EXECUTING
 KEQ56211I JOB A79999A#(TSU05318) IS EXECUTING ON THIS TERMINAL
 KEQ56192I JOB A79999AA(JOB01099) IS WAITING FOR OUTPUT
 KEQ56192I JOB A79999AB(JOB01100) IS WAITING FOR OUTPUT
 READY
```

A79999AA と A79999AB は処理を終了して、出力待ち状態です。A79999AW は現在実行中です。この登録番号に続く‘A’とか‘W’などは、利用者がバッチジョブ投入の際に自分で割り振る識別番号です。番号は A ~ Z, 1 ~ 9 までが自由につけれます。それを知らずに全部 1 などとつけていると次のように、整理がつかなくなります。

```
 READY
ST   ↵
 KEQ56211I JOB A79999A1(JOB01084) IS EXECUTING
 KEQ56211I JOB A79999A#(TSU05318) IS EXECUTING ON THIS TERMINAL
 KEQ56192I JOB A79999A1(JOB01099) IS WAITING FOR OUTPUT
 KEQ56192I JOB A79999A1(JOB01100) IS WAITING FOR OUTPUT
 READY
```

この場合は仕方がないので、例えば A79999A1(JOB01100) までを指定してジョブを区別します。そこで、注意を

ジョブ識別番号をきちんとつけましょう

A79999AB をデータセット TEST.OUTLIST に出力します。

```
 READY
OUT A79999AB PR(TEST)   ↵
 READY
```

これで TEST.OUTLIST への出力が完了しました。例のように、データセット名として PR(TEST) と指定すると、勝手に .OUTLIST が最後に付加されます．これを抑止して TEST.DATA と名前を付けたい場合は、'A79999A.TEST.DATA' と完全データセット名を指定して下さい。

先ほどの識別番号に 1 ばかりをつけた困った利用者が A79999A1(JOB01100) を出力させたい場合、面倒ですが、次のようにします。

```
 READY
OUT A79999A1(JOB01100) PR(TEST)   ↵
 READY
```

最後に、最も大事な注意を

OUT コマンドの PR オペランドを忘れないで！

PR オペランドを忘れて、OUT A79999AB とすると、端末にジョブ処理リストが出力されて、それきりシステムから消去されます。そのため、ジョブの再検索はもちろん、データセットへの出力も出来ません。消えても構わないようなジョブならばいいですが、苦労して得られたジョブだった場合、取り返しのつかないことになりますので、くれぐれもご注意下さい。

7　高速プリンターで大量のリストを印刷する

計算機にプログラムの処理を依頼するには、何らかの編集用エディタを用いてプログラムをファイルとして作成します。しかし、プログラミングに煮詰まったり、デバックがうまくいかない時など、プログラムのリストを印刷し、ペンを片手に推敲することも大事です。

UXP のファイルはセンター 2 階の PostScript printer から出力できます。 PostScript printer で出力すると、印字は確かに綺麗なのですが、処理性能は最大で 1 分あたり 5 枚と遅いので、データが大きい場合、やたらと時間がかかります。そこで、長大な Fortran プログラムやデータなど、テキスト形式のデータ出力にセンターの高速なラインプリンターを利用しては如何でしょう。 MSP データの場合の出力も、もちろんです。

次のラインナップです。

B4 版 NLP

NLP は Nihongo Line Printer の略です。連続紙を用いた出力です。日本語の印刷も可能です。処理能力は 1 分あたり 2,000 行を印刷します。ページ数では、 1 分あたり約 30 枚です。オープン室に 2 台設置されています。オープン室にある分は、センターが開館している間使用可能ですが、トナーの交換や連続用紙が切れた時の交換などは、利用者でやっていただく必要があります。

A4 版 NLP

こちらは B4 版よりさらに高速です。 A4 版なので、コンパクトな印刷が可能です。処理能力は 1 分あたり 10,000 行、ページ数では 1 分あたり約 150 枚です。ただしクローズ室に設置されていますので、利用者は直接操作出来ません。

CLP

CLP は Cutsheet Line Printer の略です。連続紙ではなくて A4, B4 用紙を用いた出力を行ないます。また、両面印刷も可能です。処理能力は A4 版で 1 分あたり約 50 枚を印刷します。両面印刷の場合は 1 分あたり約 20 枚 (40 ページ) です。オープン室に 1 台設置されています。

NLP, CLP 出力は基本負担金 40 円に加え、 1 枚あたり 4 円の入出力負担金がかかります。

UXP ファイルの出力 – utoprint コマンド –

UXP ファイルの出力は `utoprint` コマンドで行ないます。例として、 Fortran プログラム `test.f` をプリンターに出力させます。

```
kyu-cc% utoprint test.f  ↵
utoprint : output to NLP
kyu-cc%
```

`utoprint` コマンドは、 UXP から MSP へファイルを転送してバッチ処理形式で出力を行ないます。利用者は `utoprint` コマンド投入後、センター 2 階のプリンター横のコンソールから出力要求を行ない、印刷結果を受けとります。ジョブ認識番号は ‘`0`’ です。出力要求すべきジョブ名は `a79999a` さんならば、`A79999A0` が UXP ファイル `test.f` の印刷結果のジョブ名になります。

MSP データセットの出力 – LIST コマンド –

MSP データセットの出力方法は幾つかの方法があります。

command	機能
%LIST	CLP への出力形式を制御. NLP へも出力可能
SPSTINGY	2 段組の形式で NLP に出力
COMPACT	NLP, CLP に様々な形式で縮刷印字を行なう

上記コマンド群に関しては、『利用の手引・MSP コマンド編』に機能 / 使用方法が詳述してありますので、そちらを参照下さい。ここでは、LIST コマンドを用いた最も簡単な例を示します。

データセット TEST.FORT を印刷します。肝心な点は 'SY(U)' の指定です。

```
 READY
LIST TEST.FORT SY(U)
KEQ52801I  DATA SET A79999A.TEST.FORT BEING PROCESSED
 READY
```

実は UXP も MSP も、これだけの手続きでは印刷結果を得られません。利用者は、次にプリンター・コンソールから出力要求を出します。プリンター・コンソールが何処にあるかといえば、オープン室に設置された CLP, NLP ではプリンターのすぐ横に設置された端末です。クローズ室の NLP では、利用者控え室の横のターンテーブルの傍らにある端末です。双方次の画面が出ています。

```
LL          PPPPPPPPP     OOOOOOOO     UU       UU   TTTTTTTTTT
LL          PP       PP  OO        OO  UU       UU       TT
LL          PP       PP  OO        OO  UU       UU       TT
LL          PP       PP  OO        OO  UU       UU       TT
LL          PPPPPPPPP    OO        OO  UU       UU       TT
LL          PP           OO        OO  UU       UU       TT
LL          PP           OO        OO  UU       UU       TT
LLLLLLLLL   PP            OOOOOOOO      UUUUUUUU         TT




                  ENTER USERID   ===> a79999a
                  ENTER PASSWORD ===>
```

端末に向かい、 ID と password を打ち込むと、次の画面に変わります。画面はターンテーブルの傍らにある端末の場合です。

```
       NO.  JOBNAME(JOB NO.) STATUS
------------------------------------------------------------------------------
  ==>H  1  A79999A#(TSU0262) OUTPUT
  ==>   2  A79999AO(JOB0299) OUTPUT
  ==>   3  A79999AA(JOB0401) OUTPUT



                                                         O: JOB OUTPUT(B4)
                                                         H: JOB OUTPUT(A4)
                                                         K: JOB OUTPUT(KANA)
                                                         D: JOB DELETE


                                                         REQUEST ==> PF1
                                                         EXIT    ==> PF3








------------------------------------------------------------------------------
```

A79999A# が、MSP の LIST コマンドで印刷要求した分、A79999AO が、UXP の utoprint コマンドで要求した分です。クローズ室の A4 版高速 NLP での出力を行なう場合、メニューに従って、==> の横にカーソルを移動させ 'H' を入力して Enter キーを押します (return キーではありません)。B4 版 NLP の場合は ==> の横にカーソルを移動させ 'O' または 'K' を指定します。'ACCEPT' が画面に表示されれば、印刷が開始された合図です。しばらくターンテーブルの前で「ぼけっ」としていると、出力リストがベルトコンベアに運ばれてやってきます。

オープン室にある NLP, CLP も同様ですが、メニューに 'H', 'K' がありません。

最後にコンソール操作のマナーを

最後は必ず PF3 キーを押して ID 入力の画面に戻して下さい。

印刷指示を与えて、プリンターのコンソールを離れる時は、必ず PF3 を押し、前ページの LPOUT 画面に戻し、次の人が ID と PASSWORD を入力出来る状態にしてから、立ち去って下さい。

8　chfn コマンドで電子メールに人間味を出す

まず、電子メールの宣伝をします。文章は、センターで 1993 年に行なわれた電子メール講習会の資料からの抜粋です。

> 1993 年 1 月現在，電子メールが到達可能な国は約 120 カ国である．紙の文書は，封筒に入れて宛先を書き，郵便局に渡すと，宛先をもとに郵便局が配送を行ない，相手に届けることができる．これと同じように，コンピュータのファイルとして作成した文書も，宛先を指定し，電子メール配送システムに渡せば，宛先をもとに電子メール配送システムが配送を行ない，相手に届けることができる．電子メールには，手紙にはない特徴がある．
>
> 座ったままでよい
> 手紙の場合，机の前に座り，文字を書き，手紙を封筒に入れ，糊付けし，切手を貼り，ポストのあるところまで歩き，投函する．外が台風の場合や見知らぬ土地でポストの在処がわからない場合があることなどを考慮すると，相当な努力が必要になる場合がある．これに対し，電子メールは，これと同じことをするのに，コンピュータの前に座って指を動かすだけでよい．最初はどう指を動かすかわからずに苦労するかもしれないが，慣れてしまうと非常に楽である．
>
> 宛先が複数書ける
> 手紙に複数の宛先を書くとはできないが，電子メールの場合，宛先を複数書くことができ，同時に複数の人にメールを送信することができる．
>
> コピーが容易
> 機械可読な文書は，紙の文書と比べて，コピーが非常に容易であるという特徴を持つ．したがって，他人の文章を引用することも容易である．しばしば文脈を無視し，一部の文章だけを引用した挙げ足取りが横行するのも電子メールの特徴である．ご用心．
>
> 文書以外を送受信できる
> 電子メールでバイナリーデータを送受信すれば，プログラム，画像，音声など，テキストに拘らない「データ」のやりとりが可能である．
>
> その他
> セキュリティ，配送の速さ，配送の確実さなども手紙とは異なる．

「電子メール講習会・資料」より

UXP で電子メールを利用するために、次の資料が用意されています。

「UXP における電子メール入門」 九州大学大型計算機センター, 1993.

入手方法は、本稿の「Manual Guide」を御覧下さい。

さて、九州大学大型計算機センターを利用するためには、登録番号の交付を受けなければなりません。登録番号は、同一の登録番号で一人の利用者が全国 7 つの大型計算機センター及び学術情報センターを利用できるように共通利用番号制をとっています。このため利用者に交付される登録番号は、例えば `b73432a` や `f73531a` などと、簡潔ですがそっけないものになります。

UXP で電子メールを利用する場合は、この機械的な番号付けが気にいらないという利用者も多いと思います。次の電子メールを御覧下さい。

```
Date: Thu, 25 Nov 93 09:43:57 JST
From: a70040a <a70040a>  <--- そっけない登録番号
To: a70033d
Subject: Doctor

九州大学理学部の岡部です。
来週の金曜に、岸さんの博士論文公聴会が行なわれます。
既に柴田さんと、田中勝さんにはお知らせしていますが、
武さんと田原先生にはそちらから連絡して頂けないでしょうか。
よろしくお願いします。

また、公聴会の後でうち上げを予定しています。場所は波多江です。
岸さんは「菊姫大吟醸」が飲みたいとおっしゃっていますが、高いので
そうは行きません。「天狗舞」が手頃なところでしょう。

週末で真理さんと「きまぐれポニーテール」にでも行きたい気分でしょうが、
参加をお願いします。なんなら連れてきても結構です。

では。
```

このメールは a70040a さんから a70033d さんあてに配達されたものです。文章の冒頭で a70040a さんが自己紹介をしているので、a70033d さんはこのメールが誰から来たものか分かりますが、“From: a70040a” ではあまりに無味乾燥です。一般のワークステーションなどから届く電子メールは、登録課題 = 個人名で行なう場合が多いので、誰からのメールかはっきり分かります。次の例は UXP 以外から a70033d さんに届いたメールです。

```
Date: Thu, 25 Nov 93 11:05:06 JST
From: watanabe yoshitaka <watanabe@doronpa.cc.kyushu-u.ac.jp>  <--- わかりやすい
Return-Path: <watanabe@doronpa.cc.kyushu-u.ac.jp>
To: a70033d@kyu-cc.cc.kyushu-u.ac.jp
Subject: I agree

九州大学大型計算機センターの渡部です．
岡部さんからのメールで既にご存知と思いますが，
岸さんの博士論文公聴会は金曜です．
タイトルは
‘Verified Computation of Solutions for Elliptic Equations Related to JRA’
という題です．

金曜は真理さんと「で～と」したいでしょうが，土曜にして下さい．
岡部さんは彼女も一緒に飲みましょうと言っていましたが，わたしも大歓迎です．
また例の店で「いもほり」をやって勝負しましょう．
それでは，週末を元気に頑張って下さい．
```

chfn コマンド

UXP 利用者には finger 情報と呼ばれる情報があります。この情報は、電子メールの発信時に発信者名として受信者に通知される情報です。登録番号としての a70040a などは変更できませんが、この finger 情報を変更することで、メールのやりとりに少し人間味を持たせることが出来ます。例として、 finger 情報の省略値“a70040a”を“Okabe Michiko”に変えてみます。 finger 情報の変更は chfn コマンドで行ないます。

```
kyu-cc% chfn ⏎
Changing Finger Information for a70040a.

Name [a70040a]: Okabe Michiko ⏎
kyu-cc%
```

これで finger 情報が変更されました。確認は `finger` コマンドで行ないます。

```
kyu-cc% finger a70040a ⏎
Login name: a70040a                     In real life: Okabe Michiko
Directory: /home/usr0/a70040a           Shell: /usr/bin/csh
On since Nov 25 11:19:21 on pts/8 from doronpa.cc.kyushu-u.ac.jp
1 minute 21 seconds Idle Time
No unread mail
No Plan.
kyu-cc%
```

`In real life:` の項が `Okabe Michiko` になっています。`finger` コマンドは、自分の登録番号だけではなく、現在 login している利用者の情報や、UXP 内ならば、他の登録番号の人が最後にいつ login したかなどの情報も知ることができます。
上の例では `a70040a` さんの finger 情報を変更しました。変更後に `a70040a` さんから `a70033d` さんに出したメールは、`From:` の部分を見れば、誰から来たメールなのかすぐに分かるようになりました。

```
Date: Mon, 8 Dec 93 15:23:57 JST
From: Okabe Michiko <a70040a>      <----- 多少の改善
To: a70033d
Subject: Doctor

九州大学理学部の岡部です。
先週は岸さんの博士論文公聴会に出席下さり、ありがとうございます。
あの時、話に出ました、並列計算機 CM5 の利用については、
追って連絡があると思いますので、もう暫くお待ち下さい。
それから、明日から出張します。10 日のセミナーには参加できませんが、
できましたら資料を頂きたいと思います。よろしくお願いします。

波多江でのうち上げには、真理さんを連れて見えられ、本当にびっくりしました。
まあ、しばらく見ない間に随分大人っぽくなられて..... この幸せもの~。ポカポカ。
しかし、大きな声では言えませんが、彼女の歌は何とかした方がいいですよ。
その、カラオケで熱唱するのは結構なのですが、小指を立てて、しかも虚ろな目つき
                    (以下あぶないので省略)
```

なおメールの内容は、一部フィクションです。
それでは、最後にやっかいな問題を

“Okabe Michiko”にしますか。それとも“Michiko Okabe”にしますか。

9 UXP から MSP へのファイル転送の注意点

現在、 50MB 以上の領域を確保する Fortran プログラムは MSP 上で実行する必要があります。そのためには、 MSP 上にファイルを作成しなければなりません。 UXP や他のワークステーションにプログラムがある場合は、もちろん手で同じものを打ち込むのは面倒なので、MSP にファイルを転送したいところです。ここでは UXP から MSP に FTP コマンドを用いてファイルを転送する方法と、注意点を述べます。例としてあげたプログラムは、大容量長時間ジョブとは程遠いもので、大規模ジョブの必要性が切実に伝わってきませんが、ご勘弁下さい。なお、 FTP(File Transfer Protocol) に関しては次の記事をお読み下さい。

「FTP サーバ / クライアントの公開について」 九州大学大型計算機センター広報, Vol.26, No.1, pp73–80, 1993.

a79999a さんが、ディレクトリ fort の下に、test.f というプログラムを持っています。

```
      program test
      write(6,*) 'Hello Honey !'
      end
```

これを MSP に転送します。転送方法は MSP 側から FTP コマンドを用いて行ないます。MSP の A79999A から、FTP コマンドを入力します。

```
 READY
ftp   ↵                       <--- FTP コマンドの入力
 FTP
host kyu-cc   ↵               <--- UXP を指定
 KGT10104A ENTER USER NAME.
a79999a   ↵                   <--- ID と password
 KGT10101A ENTER PASSWORD.
          ↵
 FTP
```

続いて、ディレクトリ fort から test.f を転送します。 MSP 側のデータセット名は TEST.FORT として新規に作成します。

```
 FTP
cd fort   ↵                              <--- directory fort に移る
 KGT10500I 250 CWD command successful.
 FTP
recv in(test.f) out(test.fort)   ↵       <--- ファイルの転送
 KGT10013I NEW FILE ALLOCATED. DS NAME:A79999A.TEST.FORT.
 KGT10002I FILE TRANSFER ACCEPTED. DS NAME:A79999A.TEST.FORT,ID:24.
 KGT10032I FILE TRANSFER STARTED.
 KGT10003I FILE TRANSFER COMPLETED. DS NAME:A79999A.TEST.FORT,ID:24. 72  BYTES
 SENT IN 1 SECONDS (72 BYTES/S).
 FTP
end   ↵                                  <--- FTP コマンドの終了
READY
```

さて、 MSP で早速転送したプログラムを実行してみましょう。

```
 READY
run test.fort   ↵
 FORTRAN77 EX COMPILER ENTERED
 FORTRAN77 EX DIAGNOSTIC MESSAGES: PROGRAM NAME(MAIN),FLAG(I),OPTIMIZE(B)
 JWD1026I-S                NO END STATEMENT.
 JWD1043I-I                LINE NUMBER NOT DIGITS.
 JWD1044I-I                LINE NUMBER ORDER ERROR.
 JWD1302I-S      00000000  INVALID STATEMENT.
 END OF COMPILATION,HIGHEST SEVERITY CODE=12
 READY
```

何だか変なメッセージが出ました。“INVALID STATEMENT” とあるので、 Fortran の文章自体がおかしいと Fortran 77/EX コンパイラから文句を言われたようです。

データセットの中身を LIST コマンドで調べてみます。

```
 READY
LIST TEST.FORT   ↵
 KEQ52800I  A79999A.TEST.FORT
    pr ogram test                    <--- ボロボロ
    wr ite(6,*) 'Hello Honey !'
    en d
 KEQ52827I RECORD LENGTH IS TOO SHORT FOR LINE NUMBER FIELD
 READY
```

プログラムがちぎれてボロボロになっています。これは、 FTP で MSP 側に新規の Fortran データセットを作成する場合、自由形式のデータセットを作りにいってしまうためです。 MSP の Fortran 形式は自由形式と固定形式の 2 つがあります。対して、 UXP の Fortran 形式は固定形式です。このため、 UXP と MSP との Fortran 形式の差が原因で、結論からいえば、 8 文字分だけ左にずれてしまっているのです。 MSP の自由形式 / 固定形式に関しては、『利用の手引・ TSS 編』に詳しく解説されています。

対処方法は次の 2 つです。

- 新規にデータセットを作成するのを諦め、既存の固定形式のデータセットを MSP 上に作成した上で FTP で転送するとうまくいきます。
- MSP の PFD/EDIT でボロボロのプログラムを修正します。ただし、 Fortran の形式は自由形式です。

まず、第一の方法です、 FTP でファイルを転送する前に、固定長ブロック化形式のデータセットを作成します。作成方法は MSP の ALLOC コマンド、または PFD の機能を使います。詳細は再び『利用の手引・ TSS 編』を御覧頂くとして、ここでは例として MSP の ALLOC コマンドで、固定形式の順編成データセット TEST.FORT を新規に作成してみます。長いですが、めげずに打ち込んで下さい。

```
 READY
ALLOC DA(TEST.FORT) NEW CA SP(10 5) TRA REC(F B) BLK(23440) LR(80) DSO(PS)  ↵
FREE DA(TEST.FORT)  ↵       <--- FTP の場合、これも必要
 READY
```

この既成のデータセットTEST.FORT に対して、先ほどの FTP コマンドを用いて転送をおこないます。

第二番めの、ボロボロになっているデータセットの PFD/EDIT による修正方法ですが、これは、転送に失敗したように見えるデータセットを、強制的に 8 文字だけ右にずらすという力技です。実はこっちの方が簡単だったりします。実際は、プログラムは自由形式ですので、本当は 2 文字文だけの移動で実行可能ですが、 UXP との見た目の整合性を考えてみました。なお、自由形式と固定形式では、コメントや継続などで異なる部分がありますので、注意が必要です。

まず、ボロボロの TEST.FORT を編集画面に登場させます。

```
EDIT --- A79999A.TEST.FORT ---------------------------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> HALF
****** ***************************** TOP OF DATA ****************V10L30********
000001       program test
000002       write(6,*) 'Hello Honey !'
000003       end
****** **************************** BOTTOM OF DATA ****************************
```

ここで、'>>8' をプログラムの先頭と末尾に挟み込んで入力して、全体を 8 文字分右に押しやります。

```
EDIT --- A79999A.TEST.FORT ---------------------------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> HALF
****** ***************************** TOP OF DATA ****************V10L30********
>>8          program test
000002       write(6,*) 'Hello Honey !'
>>8003       end
****** **************************** BOTTOM OF DATA ****************************
```

'>>8' は、「集団で右に 8 文字動け」というサブコマンドです。 '>>2' とすれば、 2 文字右に移動します。 PFD/EDIT に関する使用方法は、次の解説記事をお読み下さい。

> 「PFD と PFDE の使用法」 平野 広幸，九州大学大型計算機センター広報, Vol.25, No.2, p119–165, 1992.

'>>8' で挟み込まれた部分は 8 文字右に移動します。

```
EDIT --- A79999A.TEST.FORT ---------------------------------- COLUMNS 001 072
COMMAND ===>                                                  SCROLL ===> HALF
****** ***************************** TOP OF DATA ****************V10L30********
000001               program test
000002               write(6,*) 'Hello Honey !'
000003               end
****** **************************** BOTTOM OF DATA ****************************
```

次に、行番号をつけます。 'REN' で行番号を表示させます。

```
EDIT --- A79999A.TEST.FORT ---------------------------------- COLUMNS 001 072
COMMAND ===> REN   ↵                                          SCROLL ===> HALF
****** ***************************** TOP OF DATA ***************V10L30********
000001              program test
000002              write(6,*) 'Hello Honey !'
000003              end
****** **************************** BOTTOM OF DATA ****************************
```

画面に行番号が表示されます。

```
EDIT --- A79999A.TEST.FORT ---------------------------------- COLUMNS 001 072
COMMAND ===> REN   ↵                                          SCROLL ===> HAL F
****** ***************************** TOP OF DATA ***************V10L30********
000100 00000100     program test
000200 00000200     write(6,*) 'Hello Honey !'
000300 00000300     end
****** **************************** BOTTOM OF DATA ****************************
```

表示が変ですが、このままデータセットをセーブして、一旦編集を終了した後、再度編集画面を呼び出すと、普段の Fortran 編集ができます。

ためしに、 RUN サブコマンドでプログラムを実行してみます。

```
EDIT --- A79999A.TEST.FORT ---------------------------------- COLUMNS 009 080
COMMAND ===> R   ↵                                            SCROLL ===> HALF
****** ***************************** TOP OF DATA ***************V10L30********
000100       program test
000200       write(6,*) 'Hello Honey !'
000300       end
****** **************************** BOTTOM OF DATA ****************************
```

下のようにうまくいきました。

```
FORTRAN77 EX COMPILER ENTERED
END OF COMPILATION,HIGHEST SEVERITY CODE=00
Hello Honey !
END OF GO,HIGHEST SEVERITY CODE=00
***
```

なお、さきほども述べましたが、データセットが自由形式なので、プログラムによっては、実行出来ない場合がありますので注意下さい。

結局、もともと転送前の UXP のプログラムが固定形式ですから、 MSP 側で自由形式に意識して変えたいという方を除けば、 FTP の時に MSP システムが新規にデータセットを自由形式で作りに行くことは、好ましいことではありません。しかし、申し訳ありませんが、FTP の際の新規に作成する Fortran の形式を固定長に変更することは出来ません。従って、結論として UXP から MSP への Fortran プログラムの転送の際は、次の注意をお守り下さい。

MSP 側に予め固定形式のデータセットを作ってから転送して下さい。

10　PostScript Printer 活用法

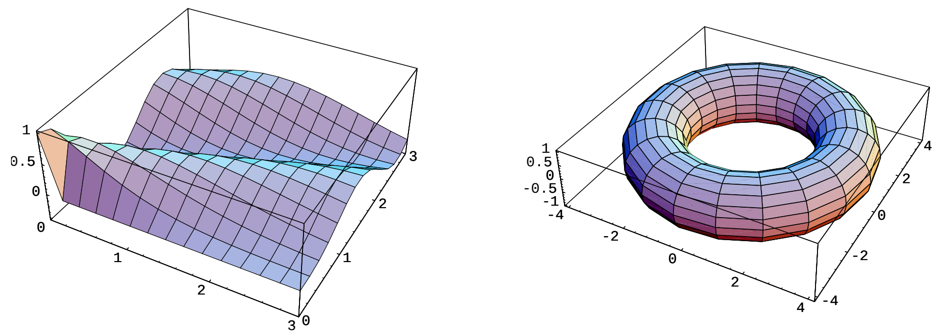
九州大学大型計算機センター 2 階には PostScript Printer が設置されています。解像度は 400dpi（１インチあたり 400 ドット）です。最近、世の中には PostScript 形式のファイルというものが出回っています。例えば、おなじみの Mathematica（おなじみでない方はおなじみそうな人に聞いて下さい）や S、 SAS などのソフトでは、図形データを PostScript ファイルで出力します。また、 TIFF, GIF, JPEG, BITMAP などのイメージデータを PostScript ファイルに変換するツールもたくさん出回っています。

ここでは、極めて具体的な例として、 PostScript ファイルを TEX に取り込んだ論文を UXP 上で作成してみます。なお、 MSP で同様の作業は出来ません。 PostScript ファイルの処理に関しては UXP のみが対応します。

PostScript ファイルの用意

近年のグラフィックス・ツールは、 PostScript データを出力するものがほぼ標準になってきました。 PostScript ファイルはテキスト形式なので、中を覗いて構造を調べることができます。また、 PostScript 言語を勉強すれば、データの修正や作成も可能です。

まず最初に用意する PostScript ファイルは、汎用数学処理システム Mathematica で作ったグラフィックスです。



次に用意する PostScript ファイルは、手書きのイラストをイメージスキャナーで読み込んだあと、 PostScript ファイルに変換したものです。

LaTeX への取り込み

以下は、LaTeX を多少使ったことのある人へ向けての解説です。何を書いてあるのかさっぱり分からない場合、重要なことは何も書いてありませんので、さっさと読み飛ばして結構です。

最初の Mathematica の PostScript ファイルを `ps1.ps`、次の、スキャナーで読み込んだものを `ps2.ps` とします。PostScript ファイルの LaTeX への取り込みは、`epsf.sty` という TeX の Style file を使います。以下 LaTeX のソースです。

`\documentstyle` の部分に `epsf` を、PostScript ファイルを挿入したい部分で`\epsfile` を指定します。`\epsfile` には大きさを調節するオプションなどがあります。
詳しくは `/usr/local/lib/tex/inputs/epsf.sty` を眺めて下さい。眺め方は次のように行ないます。

```
kyu-cc% more /usr/local/lib/tex/inputs/epsf.sty ↵
```

あとは普通の LaTeX ソースと変わりありません。
ここで、おことわりしておきます。

本文のドイツ語 (一部英語) はでたらめです

内容は極めて適当に書いています。LaTeX に取り込んだ図形も、何の関係もなく登場します。TeX の恐ろしいところは、論文の内容に関係なく"もっともらしく"見えるところです。
そういうわけで、あくまで"T EX のソース例"として見て下さい。

```
\documentstyle[11pt,epsf]{article} % epsf file
\oddsidemargin  -0.54cm
\evensidemargin -0.54cm
\topmargin -1.44cm
\textheight 24.6cm
\textwidth  16.4cm
\columnsep     2pc
\columnseprule 0pt
\begin{document}
\section*{Gau{\ss}--Seidel Verfahren}
Es sei $A = (a_{ij})$ eine $n \times n$ Matrix mit reellen Eintr\"agen
$a_{ij}$ und Diagonalelementen
$a_{ii} \neq 0$; $A^K = (a^K_{ij})$ sei die aus $A$
entstehende Matrix mit
\[
  a^K_{ij} = \left\{
   \begin{array}{lll}
     a_{ij} & \mbox{f\"ur} & j \le i \\
    -a_{ij} & \mbox{f\"ur} & j > i \; .
   \end{array}\right.
\]
Es sei $(\cdot ,\cdot )$ das euklidische innere Produkt im $n$-dimensionalen
linearen Raum $R^n$   der Vektoren $z = (z_1,z_2,\ldots,z_n )$ mit
komplexen Komponenten und f\"ur beliebige komplexe Zahlen
$z_k = u_k +iv_k,\; u_k,v_k \in U,\; k=1,2$, sei $[z_1,z_2] = u_1u_2+v_1v_2$.

Das {\em Gau{\ss}--Seidel Verfahren} f\"ur das lineare
```

```
Gleichungssystem $(b\in R^n )$ \[ Ax = b \]
ist gegeben durch die Iterationsvorschrift
\[ x_{l+1}=-M^{-1}Nx_l+M^{-1}b, \qquad l=0,1,2,\ldots, \qquad x^0\in R^n\; .\]

\begin{center}
\epsfile{file=ps1.ps,height=5.5cm}
\end{center}

Die Matrizen $M = (m_{ij})$ und $N = (n_{ij})$ sind dabei von der Form
\[
m_{ij}=\left\{
  \begin{array}{lll}
    a_{ij} & \mbox{f\"ur} & j\le i \\
      0    & \mbox{f\"ur} & j>i\;.
\end{array}\right.
\;\;\;\mbox{und}\;\;\;
n_{ij}=\left\{
   \begin{array}{lll}
     a_{ij}& \mbox{f\"ur} & j>i \\
         0 & \mbox{f\"ur} & j \le i \; .
   \end{array}\right. \;.
\]
Bekanntlich ([1]) konvergiert das Gau{\ss}--Seidel Verfahren
genau dann f\"ur ein beliebiges Ausgangselement $x_0$,
wenn alle charakteristischen Zahlen der Matrix
$-M^{-1} N$  Betr\"age kleiner als Eins haben.\\[0.3cm]

\newlength{\minitwocolumn}
\setlength{\minitwocolumn}{0.5\textwidth}
\addtolength{\minitwocolumn}{-0.5\columnsep}
\begin{minipage}[b]{\minitwocolumn}
  \begin{itemize}
    \item Verallgemeinerte Diagonaldominanz bei Matrizen
    \item Die Abweichungen
    \item Fundamental Technological Research
    \item Acknowledgements
    \item Stoffgesetzen f\"ur gro{\ss}e
  \end{itemize}
\end{minipage}%
\hspace{\columnsep}%
\begin{minipage}[t]{\minitwocolumn}
\begin{center}
\epsfile{file=ps2.ps,height=4cm}
\end{center}
\begin{center}
\vspace*{-0.3cm}
Es wurde ihr Nacht vor den Augen
\end{center}
\end{minipage}%
\end{document}
```

PostScript printer への出力

先ほどの LaTeX のソースファイルを test.tex に作成し、jlatex コマンドでコンパイルを行ない。さらに dvi2ps コマンドによって dvi ファイルを PostScript ファイルに変換し、センター 2 階の PostScript printer に印刷します。

```
kyu-cc% jlatex test.tex   ↵
kyu-cc% dvi2ps test.dvi | lp -dps -Tps   ↵
```

あとは、 PostScript printer の前で印刷結果が出てくるのを待ちます。通常、イメージデータを PostScript 形式に変換したファイルは、かなり大きめで、 1MB を越えるものも珍しくありません。そのようなファイルを LaTeX に取り込んで何枚も印刷する場合は、お手数ですが、他の利用者の迷惑になりますので、出来るだけ分割した印刷をお願いします。dvi2ps コマンドに印刷ページを '-f', '-t' オプションで指定して下さい。

```
kyu-cc% dvi2ps -f 1 -t 2 test.dvi | lp -dps -Tps   ↵
```

このようにすると、 1 ~ 2 ページが出力されます。

また、例題で用意した PostScript ファイル ps1.ps と ps2.ps は、 LaTeX に取り込むことなく、直接 PostScript Printer に出力できます。コマンドは lp コマンドです。その際次のことを厳守下さい。

PostScript ファイル出力時は -Tps オプションを絶対忘れないで下さい。

PostScript ファイル ps1.ps を印刷します。

```
kyu-cc% lp -dps -Tps ps1.ps   ↵
```

最初の '-dps' は、 2 階の PostScript printer の名前を表します。これは無条件で指定して下さい。次の '-T' に続く 'ps' オプションで、出力ファイルが PostScript 形式であることをプリンターに認識させます。もし、 lp -dps -Tps を lp -dps と指定すると、 PostScript ファイルではなく、通常のテキストファイルだと認識され、 PostScript ファイルの中身が「ずらずら」とプリンターから印刷されます。 PostScript 言語に興味のある方以外は、このような印刷結果は何の役にも立ちませんし、普通 PostScript ファイルは大きいファイルですので、紙の無駄にもなります。十分注意をお願いします。それを防ぐためには、 lp -dps -Tps を何か別のコマンドとして登録しておけば安全です。登録方法は、少しは自分で勉強して調べてみて下さい。

PostScript printer、 LaTeX に関する詳細情報は、次のガイドブックをお読み下さい。

「UXP/M for the Working Scientists」佐藤 周行, 1993.

次ページは印刷結果です。本記事全体も 2 階の PostScript printer から出力しました。

Gauß–Seidel Verfahren

Es sei $A = (a_{ij})$ eine $n \times n$ Matrix mit reellen Einträgen a_{ij} und Diagonalelementen $a_{ii} \neq 0$; $A^K = (a_{ij}^K)$ sei die aus A entstehende Matrix mit

$$a_{ij}^K = \begin{cases} a_{ij} & \text{für} \quad j \leq i \\ -a_{ij} & \text{für} \quad j > i \; . \end{cases}$$

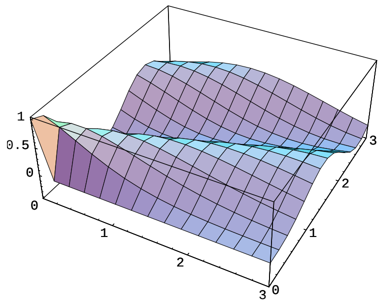
Es sei $(\cdot, \cdot)$ das euklidische innere Produkt im n-dimensionalen linearen Raum R^n der Vektoren $z = (z_1, z_2, \ldots, z_n)$ mit komplexen Komponenten und für beliebige komplexe Zahlen $z_k = u_k + iv_k,\ u_k, v_k \in U,\ k = 1, 2$, sei $[z_1, z_2] = u_1 u_2 + v_1 v_2$.

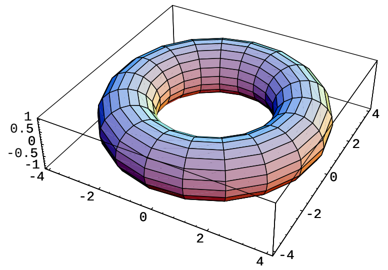
Das *Gauß–Seidel Verfahren* für das lineare Gleichungssystem ($b \in R^n$)

$$Ax = b$$

ist gegeben durch die Iterationsvorschrift

$$x_{l+1} = -M^{-1} N x_l + M^{-1} b, \quad l = 0, 1, 2, \ldots, \quad x^0 \in R^n \; .$$





Die Matrizen $M = (m_{ij})$ und $N = (n_{ij})$ sind dabei von der Form

$$m_{ij} = \begin{cases} a_{ij} & \text{für} \quad j \leq i \\ 0 & \text{für} \quad j > i \; . \end{cases} \quad \text{und} \quad n_{ij} = \begin{cases} a_{ij} & \text{für} \quad j > i \\ 0 & \text{für} \quad j \leq i \; . \end{cases} \; .$$

Bekanntlich ([1]) konvergiert das Gauß–Seidel Verfahren genau dann für ein beliebiges Ausgangselement x_0, wenn alle charakteristischen Zahlen der Matrix $-M^{-1}N$ Beträge kleiner als Eins haben.

- Verallgemeinerte Diagonaldominanz bei Matrizen
- Die Abweichungen
- Fundamental Technological Research
- Acknowledgements / Stoffgesetzen für große
- Illustrated by HIDAKI, Naoko. She designs to be a interior/exterior desinger.

Es wurde ihr Nacht vor den Augen