

計算物理学II (第4回)

第4回の内容

- シェルスクリプトの基本
- LaTeXによる文書作成
- gnuplot によるグラフ作成

シェルスクリプトの基本

ファイルの詳細情報を表示(ls)

- ls -l オプションでファイルの詳細情報を表示

```
hinohara.nobuo.ga@physicslinux1:~$ ls -l
合計 96
drwx----- 3 hinohara.nobuo.ga 4096 5月 24 2017 AppData/
drwx----- 3 hinohara.nobuo.ga 4096 4月  2 2018 Contacts/
drwx----- 3 hinohara.nobuo.ga 4096 6月 14 19:48 Desktop/
```

ファイルのパーミッション

ファイルの所有者

ファイルのサイズ

最終更新日時

- t オプション 最終更新日時の順番で表示(新しいファイルから順番に表示)
- S オプション ファイルサイズの順番で表示(大きいファイルから順)
- h オプション ファイルのサイズをバイト、キロバイト、メガバイトなどで表示
- r オプション 並び方を逆順にする

ファイルとパーミッション

ファイルのパーミッション

```
drwxr-xr-x 2 hinohara.nobuo.ga 4096 6月 28 16:01 compphys2/  
-rw-r--r-- 1 hinohara.nobuo.ga 852 6月 28 16:36 compphys2.tgz  
-rw-r--r-- 1 hinohara.nobuo.ga 21 6月 28 16:01 report.txt  
drwxr-xr-t 2 hinohara.nobuo.ga 4096 6月 26 19:38 thinclient_drives/  
lrwxrwxrwx 1 root 26 3月 21 2017 www -> /www/hinohara.nobuo.ga/www/  
lrwxrwxrwx 1 root 27 3月 21 2017 wwws -> /www/hinohara.nobuo.ga/wwws/
```

1文字目はファイルタイプ：d ディレクトリ、l シンボリックリンク

次の9文字は3文字ずつに分けて、それぞれ 所有者、グループ、その他のユーザに対するパーミッション
r: 読み込み可能(readable)、w: 書き込み可能(writable)、x: 実行可能(executable)

ファイルを表示・ディレクトリの中を表示するためには読み込み可能である必要がある。
ファイルを編集・ディレクトリの中のファイルの操作には書き込み可能である必要がある。
ディレクトリに移動する、またプログラムは実行可能である必要がある。
通常パーミッションを気にする必要はないが、自分で作成したシェルスクリプトは
実行権限を持っていないためパーミッションの変更の必要となる。

シェルスクリプト

- シェルスクリプトとは
 - 複数のコマンドをファイルに書いておき、ファイルを実行するとそのコマンドを順番に実行する。
 - 同じコマンドを何度も入力する手間が省ける。
 - 複雑な分岐、ループなども可能。(詳細は省略)

emacsでシェルスクリプトを作成してみる。以下の例では list.sh というファイルを作成し、一行目に #!/bin/bash (これはbashのシェルスクリプトであるということを指定) 下に実行したいコマンドを書く

```
hinohara.nobuo.ga@physicslinux1:~$ emacs list.sh -nw
hinohara.nobuo.ga@physicslinux1:~$ cat list.sh
#!/bin/bash
ls -l
```

ファイルのパーミッション変更(chmod)

シェルスクリプトファイルは登録されたコマンドではないのでファイル名だけでは実行できず、パスを明示的に指定する必要がある。

カレントディレクトリにあるので ./ をつけることで カレントディレクトリにあるlist.shファイルを実行する、となる。

シェルスクリプトを実行するにはあらかじめ実行権限を与える必要がある(chmodコマンド)

```
hinohara.nobuo.ga@physicslinux1:~$ list.sh
list.sh: コマンドが見つかりません
hinohara.nobuo.ga@physicslinux1:~$ ./list.sh
-bash: ./list.sh: 許可がありません
hinohara.nobuo.ga@physicslinux1:~$ ls -l list.sh
-rw-r--r-- 1 hinohara.nobuo.ga 20 6月 28 18:52 list.sh
hinohara.nobuo.ga@physicslinux1:~$ chmod u+x list.sh
hinohara.nobuo.ga@physicslinux1:~$ ls -l list.sh
-rwxr--r-- 1 hinohara.nobuo.ga 20 6月 28 18:52 list.sh*
hinohara.nobuo.ga@physicslinux1:~$ ./list.sh
合計 96
drwx----- 3 hinohara.nobuo.ga staff 4096 5月 24 2017 AppData
drwx----- 3 hinohara.nobuo.ga staff 4096 4月 2 2018 Contacts
drwx----- 3 hinohara.nobuo.ga staff 4096 6月 14 19:48 Desktop
```

これだと登録されているコマンドのリストを探しに行ってしまう

カレントディレクトリにあるlist.shを実行する、という意味になる

ファイルに実行権限がないので実行出来ない

所有者に実行権限を付与

実行可能ファイルはlsで見ると*がつく
(ファイル名には含まれない)

シェルスクリプトに書かれた
ls -l が実行される

chmod コマンド

- ファイルのパーミッションを変更する

chmod u+x ファイル名 : ユーザ(u)に実行権限(x)を付与

chmod ug+rw ファイル名 : ユーザとグループ(g)に読み込み(r)、書き込み(w)権限を付与

chmod o-x ファイル名 : その他(o)から実行権限(x)を剥奪

または r=4, w=2, x=1 の数字を使って一発で指定することも可能

chmod 777 ファイル名 : 777 は rwxrwxrwx (r+w+x=7)

chmod 755 ファイル名 : 755 は rwxr-xr-x (r+x=5)

chmod 644 ファイル名 : 644 は rw-r--r-- (r+w=6)

必要がない限りグループやその他には実行権限や書き込み権限は与えない方がよい。
(セキュリティ上の問題が起こりうるため)

シェルスクリプトを実行するには所有者に実行権限が必要であるため

chmod u+xとするか、もともと644で作られているため 744または755に変更する

ログイン時の動作(発展内容)

- bashではログイン時に初期設定が書かれたシェルスクリプトが実行される。
- ログイン時： `~/.bash_profile` が実行される。(ドットから始まるファイルは隠しファイル)
- `~/.bash_profile`には通常 `~/.bashrc`の内容を実行するように書かれているため続いて`.bashrc`の中身が実行される。
- ユーザ側で設定を変えたい場合は慣習的に`~/.bashrc`に記述する
- `.bash_profile`と `.bashrc`の内容を`cat`で見よう。
 - `.bashrc`で通常設定するのは環境変数とエイリアスの設定

環境変数の例(発展内容)

- PATH: コマンドが格納されている絶対パスを登録

```
hinohara.nobuo.ga@physicslinux1:~$ echo $PATH
/usr/local/anaconda3/bin:/home/hinohara.nobuo.ga/.rbenv/shims:/home/hinohara.nobuo.ga/.rbenv/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/opt/Adobe/Reader9/bin:/usr/local/src/trunk/bin:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/local/root/bin
```

- PS1: シェルのプロンプトの表示形式を指定

```
hinohara.nobuo.ga@physicslinux1:~$ echo $PS1
\[ \e]0;\u@\h: \w\a] ${debian_chroot:+($debian_chroot)}\u@\h:\w\$
```

- SHELL: ログイン時に起動するシェルを指定

```
hinohara.nobuo.ga@physicslinux1:~$ echo $SHELL
/bin/bash
```

- HOME: ホームディレクトリの絶対パスを格納

```
hinohara.nobuo.ga@physicslinux1:~$ echo $HOME
/home/hinohara.nobuo.ga
```

echoコマンド: シェル変数の値を表示。(シェル変数は\$から始まる)文字列を渡した場合は文字列をそのまま表示

```
hinohara.nobuo.ga@physicslinux1:~$ echo Hello
Hello
```

←文字列Helloの場合

シェルでのコマンド処理の流れ(発展内容)

- シェルで文字列を入力(例：cat test.txt など)
- シェルは空白で区切られた1単語目(例の場合はcat)をコマンドと認識
- 1. エイリアスが登録されているかチェック
- 2. ビルトインコマンド(history, echoなど)かチェック
- 3. 環境変数PATHに登録されているディレクトリにコマンドを探しに行く
ディレクトリを順番に探し、はじめに見つかったコマンドを実行
- コマンドがなければエラーメッセージ「コマンドが見つかりません」(Command Not Found)を標準出力に出力
- コマンドを実行したときにどこのコマンドが実行されるかは**which**コマンドで調べられる

```
[hinohara.nobuo.ga@physicslinux1:~$ which cat  
/bin/cat
```

cat と打つと PATHに登録されているディレクトリの中から/**bin/cat**が呼び出されることがわかる

エイリアス(発展内容)

alias エイリアス名='コマンド'

コマンドに別名をつけることができる。 .bashrcに設定しておくとも自動で読み込まれるため常に使用可能

よく使われる設定例

```
alias ls='ls -FG'    lsと打ったときに -FGオプションが自動でつくように設定
alias sl='ls -FG'    間違えてslと打ったときもlsが実行されるように設定
alias ll='ls -laG'   ls -lをllとして設定
alias la='ls -aG'    ls -aをlaとして設定
```

これらのエイリアスは既定で設定されている場合もあり、
これまでに使ったlsはエイリアスだった可能性もある。

lsのプログラムの本体はwhichで調べると/bin/lsにあるので、これを直接実行したときと挙動を比べてみよう。

```
hinohara.nobuo.ga@physicslinux1:~$ ls
AppData/  Downloads/  Pictures/  Videos/  list.sh*  www@  ドキュメント/  公開/
Contacts/  Favorites/  Saved Games/  WINDOWS/  report.txt  wwws@  ビデオ/
Desktop/  Links/  Searches/  compphys2/  tex/  ダウンロード/  ピクチャ/
Documents/  Music/  Start Menu/  compphys2.tgz  thinclient_drives/  テンプレート/  ミュージック/
hinohara.nobuo.ga@physicslinux1:~$ /bin/ls
AppData  Downloads  Pictures  Videos  list.sh  www  ドキュメント  公開
Contacts  Favorites  Saved Games  WINDOWS  report.txt  wwws  ビデオ
Desktop  Links  Searches  compphys2  tex  ダウンロード  ピクチャ
Documents  Music  Start Menu  compphys2.tgz  thinclient_drives  テンプレート  ミュージック
```

上のエイリアスが設定されている場合はls -FGが実行される

絶対パスでコマンドを指定。ディレクトリに/が表示されない

エイリアス(発展内容)

よく使われる設定例

```
alias rm='rm -i'  
alias cp='cp -i'  
alias mv='mv -i'
```

すでに設定されている可能性がある。

-iオプションは削除、上書きするときに確認メッセージを出力

```
hinohara.nobuo.ga@physicslinux1:~/compphys2$ cp list.txt report.txt  
cp: 'report.txt' を上書きしますか? ^C  
hinohara.nobuo.ga@physicslinux1:~/compphys2$ rm report.txt  
rm: 通常ファイル 'report.txt' を削除しますか? ^C  
hinohara.nobuo.ga@physicslinux1:~/compphys2$ /bin/cp list.txt report.txt  
hinohara.nobuo.ga@physicslinux1:~/compphys2$ /bin/rm report.txt
```

エイリアスが設定されている場合は-iオプション

直接実行すると確認なしで上書き・削除が実行される
(注意してください)

type と which (発展内容)

type コマンド名 : コマンドがエイリアスか、ビルトインか、PATHに登録されているかを示す
which コマンド名 : コマンドの絶対パスを表示

```
hinohara.nobuo.ga@physicslinux1:~$ type cat
cat は /bin/cat です
hinohara.nobuo.ga@physicslinux1:~$ type history
history はシェル組み込み関数です
hinohara.nobuo.ga@physicslinux1:~$ type ls
ls は `ls -FG' のエイリアスです
```

PATHで登録されているか、ビルトインか、エイリアスかを表示

```
hinohara.nobuo.ga@physicslinux1:~$ which cat
/bin/cat
hinohara.nobuo.ga@physicslinux1:~$ which history
hinohara.nobuo.ga@physicslinux1:~$ which ls
/bin/ls
```

絶対パスが表示される
ビルトインコマンドではパスが表示されない

- コマンドが存在するはずなのにwhichでコマンドが見つからないとき
→PATHにコマンドのディレクトリが登録されていない(PATHが通っていない)
- PATHの内容は変更可能。通常は.bashrcで設定。ソフトウェアインストール時などに行う。
(方法は自分で調べてください。ただし適切に設定しないとよく使うコマンドが見つからなくなるので注意)

LaTeXによる文書作成

LaTeX

- 文章作成の組版システム。ラテフ、ラテックなどと呼ばれる
- 数式がきれいに書ける。数学・物理向け
- 物理ではMicrosoft Wordよりも標準的に使われる
- 授業のレポート、卒業論文などにも使用可能

- Windows, Macでも(TeXシステムをインストールすれば)使用可能
- 最近はOverleafというオンラインで共同編集できるTeXシステムもよく使われる
- この授業では**emacs**と**シェル**の練習を兼ねて全学計算機システムのLaTeXを利用

TeXによる文章作成の流れ

- TeXソースファイルを作成(拡張子 tex)
- 図を含む場合は図のファイル(eps, pdfなど)も準備
- TeXのコマンドでコンパイルして文章pdfファイルなどを作成

サンプルのTeXファイルからpdfを作成

<https://wwwnucl.ph.tsukuba.ac.jp/~hinothara/compphys2-20/doc/texsample.tgz>

からTexのサンプルをダウンロードして解凍する。

```
hinothara.nobuo.ga@physicslinux1:~$ mkdir tex
hinothara.nobuo.ga@physicslinux1:~$ cd tex
hinothara.nobuo.ga@physicslinux1:~/tex$ ls
hinothara.nobuo.ga@physicslinux1:~/tex$ wget https://wwwnucl.ph.tsukuba.ac.jp/~hinothara/compphys2-20/doc/texsample.tgz
--2020-07-01 17:38:00-- https://wwwnucl.ph.tsukuba.ac.jp/~hinothara/compphys2-20/doc/texsample.tgz
wwwnucl.ph.tsukuba.ac.jp (wwwnucl.ph.tsukuba.ac.jp) をDNSに問いあわせています... 130.158.70.169
wwwnucl.ph.tsukuba.ac.jp (wwwnucl.ph.tsukuba.ac.jp)|130.158.70.169|:443 に接続しています... 接続しました。
HTTPによる接続要求を送信しました、応答を待っています... 200 OK
長さ: 15873 (16K) [application/x-gzip]
`texsample.tgz' に保存中

texsample.tgz                               100%[=====]

2020-07-01 17:38:00 (209 MB/s) - `texsample.tgz' へ保存完了 [15873/15873]

hinothara.nobuo.ga@physicslinux1:~/tex$ ls
texsample.tgz
hinothara.nobuo.ga@physicslinux1:~/tex$ tar zxvf texsample.tgz
texsample/
texsample/._sinxcosx.pdf
texsample/sinxcosx.pdf
texsample/sample.tex
hinothara.nobuo.ga@physicslinux1:~/tex$ ls
texsample/ texsample.tgz
hinothara.nobuo.ga@physicslinux1:~/tex$ cd texsample/
hinothara.nobuo.ga@physicslinux1:~/tex/texsample$ ls
sample.tex sinxcosx.pdf
```

作業ディレクトリを作ってその中で作業するとよい
wgetコマンドで直接URLからダウンロード
tar.gz圧縮ファイルがダウンロードされた
texsampleという名前のディレクトリが解凍された
texソースファイルとpdfファイル(罫)が入っている

TeXソースファイル

- sample.texがソースファイル。通常はemacsなどのテキストエディタで作成。
- まずは修正せずにこのままコンパイルしてみよう。

```
File Edit Options Buffers Tools TeX Text Help
\documentclass{jsarticle}
\usepackage[dvipdfmx]{graphicx} %図を組み込む場合に必要のパッケージ
\usepackage{amsmath,amssymb} %数式関係
\usepackage{bm} % 数式で太字を使うのに使います。

\title{TeXのテストドキュメント} %タイトル
\author{日野原 伸生} %名前
\date{\today} %\todayで本日の日付が入ります。

\begin{document} %ここから本文
\maketitle %これでタイトルなどが表示されます。

\begin{abstract}
ドキュメントに概要(アブストラクト)がある場合はここに記載します。
\end{abstract}

\section{はじめに}
TeXで文章を書いてみましょう。
半角スペースを 1つ だけ入力しても 複数 入力しても1つとしてカウントされます。
文章を途中で改行する場合はTeXソースファイルで1行空けます。

これで改行されます。

\section{数式の書き方}

\subsection{インライン数式モード}
インライン数式モードとして文章中に数式を入れることもできます。  $f(x)=x$ などできます。

\subsection{ディスプレイ数式モード}
ディスプレイ数式モードでは数式用に1行使われます。
\begin{align}
f(x) = \sin x
\end{align}
この場合は自動的に式番号が振られます。
align環境のすべての式で式番号がいらぬ場合は
\begin{align*}
\mathbf{\nabla} \cdot \mathbf{A} = \mathbf{\nabla} \cdot \mathbf{A} = \mathrm{div} \mathbf{A}
\end{align*}
またはその行のみ式番号がいらぬ場合はnonumberも使えます。
\begin{align}
\hat{\rho} = \frac{\hbar}{i} \frac{d}{dx} \nonumber
\end{align}
```

dviファイルの作成(コンパイル)

- platexを使った方法ではまずdvi (device-independent file format)ファイルを作成する。
- platex texファイル でtexファイルからdviファイルを作成
- texソース・ファイルの中で参照をしている場合は2回platexコマンドを実行

```
hinohara.nobuo.ga@physicslinux1:~/tex/texsample$ ls
sample.tex  sinxcosx.pdf
hinohara.nobuo.ga@physicslinux1:~/tex/texsample$ platex sample.tex
This is e-pTeX, Version 3.14159265-p3.8.2-190131-2.6 (utf8.euc) (TeX Live 2019) (preloaded format=platex)
restricted \write18 enabled.
entering extended mode
```

中略

```
LaTeX Warning: There were undefined references.

LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.

)
Output written on sample.dvi (4 pages, 9776 bytes).
Transcript written on sample.log.
hinohara.nobuo.ga@physicslinux1:~/tex/texsample$ platex sample.tex
```

中略

```
[2] [3] [4] (./sample.aux)
Output written on sample.dvi (4 pages, 9756 bytes).
Transcript written on sample.log.
```

定義されていない参照がある
というエラーが出たらplatexを
もう一度実行

2回目はエラーが出ない
dviファイルが作成される

dviファイルからpdfファイルを作成する

- dviファイルを表示する必要はない。
- **dvipdfmx dviファイル名** でdviファイルからpdfファイルを作成

```
hinohara.nobuo.ga@physicslinux1:~/tex/texsample$ dvipdfmx sample.dvi  
sample.dvi -> sample.pdf  
[1][2][3][4]  
133030 bytes written  
hinohara.nobuo.ga@physicslinux1:~/tex/texsample$ ls  
sample.aux  sample.dvi  sample.log  sample.pdf  sample.tex  sinxcosx.pdf
```

同名のpdfファイルが作成された

pdfファイルを表示

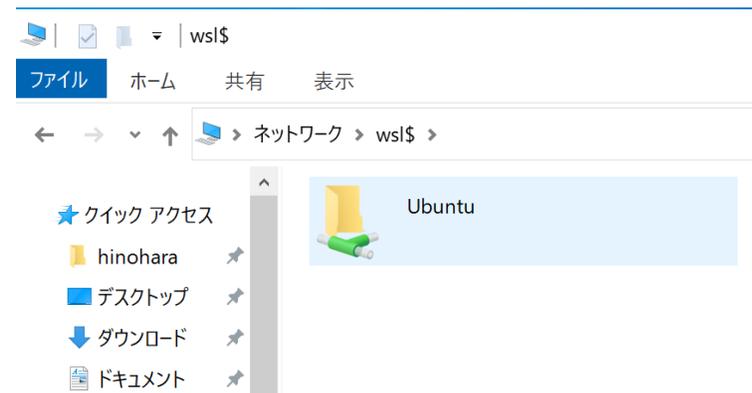
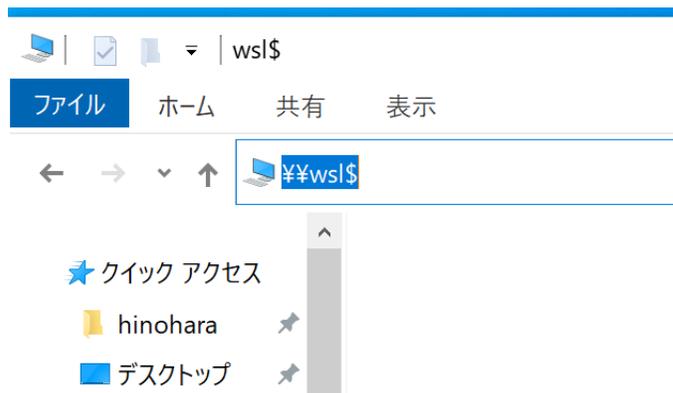
- サテライト室なら直接画面に表示できるが
リモートで作業する場合は自分のコンピュータにscpで転送する(ちょっと面倒)
- Windows: WinSCPでpdfファイルをWindowsに転送
- Mac, WSL: scpコマンドで転送。
- サーバの~/tex/texsample/sample.pdfファイルをカレントディレクトリに転送したい場合は
- **scp s1234567@physicslinu1.u.tsukuba.ac.jp:tex/texsample/sample.pdf .**

自分のコンピュータで実行

```
hinohara@MacBookPro compphys2tex % ls
hinohara@MacBookPro compphys2tex % scp hinohara.nobuo.ga@physicslinux1.u.tsukuba.ac.jp:tex/texsample/sample.pdf .
hinohara.nobuo.ga@physicslinux1.u.tsukuba.ac.jp's password:
sample.pdf                                     100% 130KB 1.3MB/s 00:00
hinohara@MacBookPro compphys2tex % ls
sample.pdf
```

pdfファイルを表示

- Windows: エクスプローラーで該当のフォルダを開いてAdobe Readerなどで表示
- Mac: Finderで該当のフォルダを開いてAdobe Reader, skimなどで表示
- WSL: **WindowsとUbuntuのファイルシステムは別**
WindowsからUbuntuのファイルにアクセス
→ エクスプローラーで **¥¥wsl\$** と入力するとUbuntuのファイルシステムが見える
ホームディレクトリにあるpdfファイルを探してAdobe Readerなどで表示
UbuntuからWindowsのファイルシステムにコピー
→ Ubuntu側からWindowsのCドライブは /mnt/c として見えるのでcpコマンドでpdfファイルをwindows側にコピー。



TeX のテストドキュメント

日野原 伸生

2020 年 7 月 1 日

概要

ドキュメントに概要 (アブストラクト) がある場合はここに記載します。

1 はじめに

TeX で文章を書いてみましょう。半角スペースを 1 つ だけ入力しても 複数 入力しても 1 つとしてカウントされます。文章を途中で改行する場合は TeX ソースファイルで 1 行空けます。これで改行されます。

2 数式の書き方

2.1 インライン数式モード

インライン数式モードとして文章中に数式を入れることもできます。 $f(x) = x$ などとできます。

2.2 ディスプレイ数式モード

ディスプレイ数式モードでは数式用に 1 行使われます。

$$f(x) = \sin x \tag{1}$$

sample.pdfの表示例

TeXソースファイルの構造

sample.texをemacsで開いて中身をチェック。

バックslash記号(\)を出すには日本語キーボードでは¥マーク(キーボードの右上)を使う。

冒頭でドキュメントクラスを指定。日本語でplatexを使う場合はjsarticleなど

```
\documentclass{jsarticle}
\usepackage[dvipdfmx]{graphicx} %図を組み込む場合に必要なパッケージ
\usepackage{amsmath,amssymb} %数式関係
\usepackage{bm} % 数式で太字を使うのに使います。

\title{TeXのテストドキュメント} %タイトル
\author{日野原 伸生} %名前
\date{\today} %\todayで本日の日付が入ります。

\begin{document} %ここから本文
```

¥documentclassから¥begin{document}までの領域をプリアンブルと呼ぶ

¥begin{document}から¥end{document}までが本文。

texでは¥begin{xxx} ~ ¥end{xxx}で囲まれた部分をxxx環境と呼ぶ。

```
\end{document}
```

%記号より右側はコメント文。コンパイル時には無視されるので説明を書く。

プリアンブルに書くこと

- 使うパッケージを指定
 - `\usepackage{パッケージ名}`
- 用紙の余白サイズ指定など
- タイトル、著者、日付などはプリアンブルに記載
 - `\title{タイトル名}`
 - `\author{著者名}`
 - `\date{日付}` %本日の場合は `\date{\today}` とする
- タイトル・著者・日付を本文に表示するには
`\begin{document}`の直後に**`\maketitle`**と書く

本文

- 概要がある場合はabstract環境に記述。
- ¥sectionで節番号が自動的に割り振られる。大きい順に部: ¥part, 章: ¥chapter, 節: ¥section, 小節: ¥subsection, ¥subsubsectionのように分割できる。

```
\begin{document} %ここから本文  
\maketitle %これでタイトルなどが表示されます。
```

```
\begin{abstract}  
ドキュメントに概要(アブストラクト)がある場合はここに記載します。  
\end{abstract}
```

```
\section{はじめに}
```

TeXで文章を書いてみましょう。
半角スペースを 1つ だけ入力しても 複数 入力しても1つとしてカウントされます。
文章を途中で改行する場合はTeXソースファイルで1行空けます。

これで改行されます。

```
\section{数式の書き方}
```

```
\subsection{インライン数式モード}
```

インライン数式モード

- 文章中に数式を用いる場合は\$ \$で囲んで中に数式を記述

文章中に数式を入れることもできます。\$f(x)=x\$などとできます。



文章中に数式を入れることもできます。 $f(x) = x$ などとできます。

ディスプレイ数式モード

- 独立行に数式を用いる場合
equation環境、eqnarray環境などがあるが
おすすめはamsmathパッケージ(アメリカ数学会)の**align環境**を使う
プリアンブルに **`\usepackage{amsmath}`**
- 式番号は自動で振られる

```
ディスプレイ数式モードでは数式用に1行使われます。  
\begin{align}  
  f(x) = \sin x  
\end{align}
```



ディスプレイ数式モードでは数式用に1行使われます。

$$f(x) = \sin x$$

(1)

数式での改行

- align環境では複数の数式を入れることができる。
- 縦位置を揃えるには & を使う(通常=のところで揃える)
- 改行は \\

```
\begin{align}
e^x &= \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots \\
F(x) &= \int_0^{\infty} f'(x)g(x) dx \\
&= \Big[ f(x)g(x) \Big]_0^{\infty} - \int_0^{\infty} f(x)g'(x) dx
\end{align}
```

改行

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2} + \frac{x^3}{3!} + \cdots \quad (2)$$
$$F(x) = \int_0^{\infty} f'(x)g(x)dx$$
$$= \left[f(x)g(x) \right]_0^{\infty} - \int_0^{\infty} f(x)g'(x)dx \quad (3)$$

数式モードで使える記号

- インターネットに多くの情報があるので検索してください
- よく使うものだけ掲載。

二項演算子

表記	出力	表記	出力
¥pm	±	¥div	÷
¥mp	(マイナスプラス)	¥cdot	•
¥circ	○	¥ast	*
¥times	×		

関係記号

表記	出力	表記	出力	表記	出力
<	<	¥in	∈	¥ll	≪
>	>	¥neq	≠	¥gg	≫
¥leq	≦	¥sim	∼	¥cong	≈
¥geq	≧	¥approx	≈	¥perp	⊥

数式モードで使える記号

矢印

表記	出力	表記	出力
\uparrow	↑	\leftrightarrow	↔
\downarrow	↓	\rightarrow	→
\leftarrow	←	大文字から始めると二重線になる	

括弧

表記	出力	表記	出力
\langle	⟨	\rangle	⟩

量子力学で出てくるブラ(bra)は \langle 、ケット(ket)は \rangle で表現

その他の記号

表記	出力	表記	出力
\hbar	\hbar (プランク定数を 2π で割ったもの)	∞	∞
∇	∇	∂	∂
\prime	' (上付きではない)	\sum	Σ (和の記号)
\int	\int	\prod	Π (積の記号)

数式モードで使えるギリシャ文字

表記	出力	表記	出力	表記	出力	表記	出力
¥alpha	α	¥iota	ι	¥sigma	σ	¥Delta	Δ
¥beta	β	¥kappa	κ	¥varsigma	ς	¥Theta	Θ
¥gamma	γ	¥lambda	λ	¥tau	τ	¥Lambda	Λ
¥delta	δ	¥mu	μ	¥upsilon	υ	¥Xi	Ξ
¥epsilon	ϵ	¥nu	ν	¥phi	ϕ	¥Pi	Π
¥varepsilon	ε	¥xi	ξ	¥varphi	ϕ	¥Sigma	Σ
¥zeta	ζ	¥pi	π	¥chi	χ	¥Upsilon	Υ
¥eta	η	¥varpi	π	¥psi	ψ	¥Phi	Φ
¥theta	θ	¥rho	ρ	¥omega	ω	¥Psi	Ψ
¥vartheta	θ	¥varrho	ρ	¥Gamma	Γ	¥Omega	Ω

数式モードで使う文字のアクセント・数学関数など

表記	出力	表記	出力
x^a	x^a	\hat{x}	xの上に^
x^{a+b}	x^{a+b}	\bar{x}	xの上に横線
x_a	x_a	$\frac{a}{b}$	a/b (分数)
x_{a+b}	x_{a+b}	\sqrt{x}	\sqrt{x}
\vec{x}	xベクトル	\mathbf{x}	x
\log	log	\lim	lim
\sin	sin	\sinh	sinh
\cos	cos	\cosh	cosh
\tan	tan	\tanh	tanh

表の書き方

表の出力位置。h: texソースのその場所, t: ページ上端, b: ページ下端, p: 単独ページ

```
\begin{table}[h]
\caption{表の説明をここに書きます。table環境の中のtabular環境で表を作成します。
tabularの横にのcは中央揃え、
lは左揃え、rは右揃えの列を表します。列は&で区切り、hlineで横線を引きます。
物理では表に縦線は通常使いません。説明はcaptionに記載し、表の上に配置します。 \label{table:fg}}
\begin{center}
\begin{tabular}{clr} % 3列の表。すべて中央揃えなら{ccc}とする
\hline\hline %横線を描く
 $x$  &  $f(x)$  &  $g(x)$  \\ \hline % 列は & で区切る。¥¥は改行
0.000 & 0.000 & 0.000 \\
10.0 & 10.0 & 10.0 \\
20 & 20 & 20 \\ \hline\hline
\end{tabular}
\end{center}
\end{table}
```

本文中で表を参照するときには使うラベルはcaptionの中に記入する(出力されない)

表の本体

表 1 表の説明をここに書きます。table環境の中のtabular環境で表を作成します。tabularの横にのcは中央揃え、lは左揃え、rは右揃えの列を表します。列は&で区切り、hlineで横線を引きます。物理では表に縦線は通常使いません。説明はcaptionに記載し、表の上に配置します。

¥begin{tabular}{...}に列数を書く。

↑の例の{clr}では3列、
1列目は中央揃え、2列目は左揃え、3列目は右揃え。
通常は{ccc}や{rrr}などになる。

x	$f(x)$	$g(x)$
0.000	0.000	0.000
10.0	10.0	10.0
20	20	20

図の入れ方

- 図を用意する(今の例ではsinxcosx.pdfファイルを使う)
- **プリアンブルでgraphicxパッケージを読み込む**

```
\usepackage[dvipdfmx]{graphicx} %図を組み込む場合に必要なパッケージ
```

- 図はfigure環境で `\includegraphics{図のファイル名}`として呼び出す

```
\begin{figure}[h] 図の表示位置  
  \begin{center}  
    \includegraphics[width=7cm]{sinxcosx.pdf}  
  \end{center}  
  \caption{図の説明をここに書きます。図のcaptionは図の下側に配置します。 \label{fig:sincos}}  
\end{figure}
```

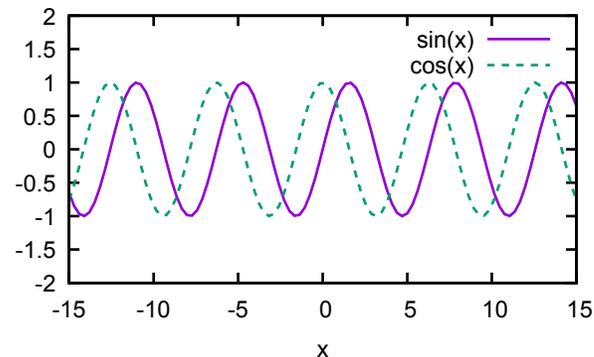


図1 図の説明をここに書きます。図のcaptionは図の下側に配置します。

参照

“式(4)”のような式・表・図・節などを指定する番号は、式の数編集して増減させると番号が変わるため本文中では番号を直接書かず`\ref{}`によって指定する。

行列は`pmatrix`などを数式モードの中で使います。いくつかの括弧が選べます。

```
\begin{align}
\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \quad
\sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \quad
\sigma_z = \begin{Bmatrix} 1 & 0 \\ 0 & -1 \end{Bmatrix}. \quad \quad \label{eq:sigma}
\end{align}
```

`align`環境の中に`label`を埋め込む(ラベルの名前は何でもよい)

行列は `pmatrix` などを数式モードの中で使います。いくつかの括弧が選べます。

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{Bmatrix} 1 & 0 \\ 0 & -1 \end{Bmatrix}. \quad (4)$$

式や図表の番号を参照することができます。式や図表(図表は`caption`の中が望ましい)に適切な名前のラベルを置き、文章中で`\ref{sec:reference}`節、`式(\ref{eq:sigma})`、表`\ref{table:fg}`、図`\ref{fig:sincos}`のようにすることで参照できます。

本文中で`\ref{}`としてこの式の番号を参照できる。

式や図表の番号を参照することができます。式や図表(図表は`caption`の中が望ましい)に適切な名前のラベルを置き、文章中で5節、`式(4)`、表1、図1のようにすることで参照できます。参照を行うには2回コン

参考文献

参考文献は本文の最後のthebibliography環境に記述。このbibitemでつけた名前で本文中からciteすることで参考文献の番号を表示できる。

参考文献がある場合に文章の途中でciteによって引用出来ます `\cite{compphys2}`。こちらも参照情報がauxファイルに保存されるため2回コンパイルを行ってください。複数の文献をまとめて引用することも出来ます `\cite{compphys2,linux,calc}`。

```
%参考文献リスト
\begin{thebibliography}{99} 文献の番号の桁数。2桁まで行きそうであれば{99}としておく
\bibitem{compphys2} 計算物理学II授業資料
\bibitem{linux} 三宅英明、大角祐介、新しいLinuxの教科書 (SB Creative)
\bibitem{calc} 伊理正夫、藤野和建、数値計算の常識 (共立出版)
\end{thebibliography}
```

参考文献がある場合に文章の途中でciteによって引用出来ます `[1]`。こちらも参照情報がauxファイルに保存されるため2回コンパイルを行ってください。複数の文献をまとめて引用することも出来ます `[1, 2, 3]`。参

参考文献

- [1] 計算物理学 II 授業資料
- [2] 三宅英明、大角祐介、新しいLinuxの教科書 (SB Creative)
- [3] 伊理正夫、藤野和建、数値計算の常識 (共立出版)

TeXのエラー

- platexでコンパイルするときエラーが出ることがある。(dvipdfmxでのエラーはほとんどない)
- align環境の中に空行があるとエラーになる。
 - 見やすさの観点から空行を入れたい場合は %としてコメント行としておけばOK
- 数式で{ }の左右の括弧の数が合わないとエラーが出る。
- 二重に上付き・下付きをするとエラーになる 例： $x^a_b^c$ 。 $x^{\{ac\}}_b$ とする。
- 長い数式でエラーの場所がわからない場合はコメントアウトをしてエラーが出ない部分だけを表示し、少しずつ表示箇所を増やすことで、エラーの箇所を見つける。

エラーの例(align環境に空行がある)

```
[hinohara.nobuo.ga@physicslinux1:~/tex/texsample$ platex sample.tex
This is e-pTeX, Version 3.14159265-p3.8.2-190131-2.6 (utf8.euc) (TeX Live 2019) (preloaded format=platex)
restricted \write18 enabled.
entering extended mode
(./sample.tex
pLaTeX2e <2020-02-02>+3 (based on LaTeX2e <2020-02-02> patch level 5)
L3 programming layer <2020-02-25>
(/usr/local/texlive/2019/texmf-dist/tex/platex/jsclasses/jsarticle.cls
Document Class: jsarticle 2020/02/02 jsclasses (okumura, texjporg)
(/usr/local/texlive/2019/texmf-dist/tex/platex/jsclasses/jslogo.sty))
```

中略

```
(./sample.aux) (/usr/local/texlive/2019/texmf-dist/tex/latex/amsfonts/umsa.fd)
(/usr/local/texlive/2019/texmf-dist/tex/latex/amsfonts/umsb.fd)
```

Runaway argument?

```
f(x) = \sin x
```

このあたりでエラー？

```
! Paragraph ended before \align was complete.
<to be read again>
```

```
\par
```

```
l.33
```

line 33あたりでエラー

```
?
```

qまたはC-dで終了できる

```
\subsection{ディスプレイ数式モード}
ディスプレイ数式モードでは数式用に1行使われます。
\begin{align}
f(x) = \sin x
```

←ここが33行目

```
\end{align}
```

この場合は自動的に式番号が振られます。

align環境のすべての式で式番号がいない場合は

```
\begin{align*}
```

```
\bm{\nabla}\cdot\bm{A} = \vec{\nabla}\cdot\vec{A} =
```

またはその行のみ式番号がいない場合はnonumberも使えます

```
\begin{align}
```

```
\hat{p} = \frac{\hbar}{i}\frac{d}{dx} \nonumber
```

```
\end{align}
```

```
---UUU:----F1 sample.tex Top L33 (LaTeX) -----
```

カーソルのある場所の行数が表示される

エラーメッセージを見てemacsでエラーの場所を調べる

TeXのエラー

- **texファイルを修正したのにpdfファイルに変更が反映されない**
platexの後、エラーなく新しいdviファイルが作成されているか？
dvi2pdfで新しいpdfファイルが作成されているか？
新しいpdfファイルをscpなどでもう一度自分のコンピュータに転送したか？
自分のコンピュータで以前のpdfファイルを開いたままではないか？
- 出てきた英語のメッセージを読んでみる。ファイルが作られた・転送した書いてあるか？
- **ls -ltr** オプションでファイルを更新時間の古い順に並べる。dviやpdfファイルの更新時間は正しいか？

```
hinohara.nobuo.ga@physicslinux1:~/tex$ ls -ltr
合計 304
-rw-r--r-- 1 hinohara.nobuo.ga 664 6月 29 14:40 sample.tex
-rw-r--r-- 1 hinohara.nobuo.ga 308 6月 29 14:41 sample.aux
-rw-r--r-- 1 hinohara.nobuo.ga 197899 6月 29 14:41 sample.pdf
-rw-r--r-- 1 hinohara.nobuo.ga 23059 6月 29 14:41 sample.log
-rw-r--r-- 1 hinohara.nobuo.ga 1078 6月 30 18:30 sample2.tex~
-rw-r--r-- 1 hinohara.nobuo.ga 1084 6月 30 18:32 sample2.tex
-rw-r--r-- 1 hinohara.nobuo.ga 420 6月 30 18:32 sample2.aux
-rw-r--r-- 1 hinohara.nobuo.ga 1808 6月 30 18:32 sample2.dvi
-rw-r--r-- 1 hinohara.nobuo.ga 7124 6月 30 18:32 sample2.log
-rw-r--r-- 1 hinohara.nobuo.ga 37993 6月 30 18:32 sample2.pdf
```

dviやpdfファイルがtexファイルより古い場合はおかしい

gnuplot

gnuplot

- グラフを作成するソフトウェア
- 画面にグラフを表示・eps,pdfなどの形式のファイルに保存が可能
- リモートから使う場合はファイルに保存を推奨
- Windows/Macにインストールすることも可能(省略)
- Linuxで図をpdfファイルとして出力する方法を解説
- インタラクティブ(対話型)に使う方法とバッチ処理する方法がある

gnuplotの使い方(インタラクティブ)

- gnuplot コマンドで起動。gnuplotのプロンプトが表示される (gnuplot>)
- 一行ずつgnuplotにコマンドを送る
- ちょっと結果を確認するだけのグラフを書くだけならこちらが便利。

```
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ ls
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ gnuplot

      G N U P L O T
      Version 5.0 patchlevel 3      last modified 2016-02-21

      Copyright (C) 1986-1993, 1998, 2004, 2007-2016
      Thomas Williams, Colin Kelley and many others

      gnuplot home:      http://www.gnuplot.info
      faq, bugs, etc:   type "help FAQ"
      immediate help:   type "help" (plot window: hit 'h')

Terminal type set to 'qt'
gnuplot> set term pdfcairo 出力形式をpdfにする
Terminal type set to 'pdfcairo'
Options are ' transparent enhanced fontscale 0.5 size 5.00in, 3.00in '
gnuplot> set output "x.pdf" 出力ファイル名をx.pdfにする
gnuplot> plot x y=xをプロット
gnuplot> exit 終了
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ ls
x.pdf
hinohara.nobuo.ga@physicslinux1:~/gnuplot$
```

pdfが作成される

gnuplotの使い方(バッチ処理)

- ファイルにコマンドのリストを書いておき、gnuplot起動時にファイルを指定する。
- コマンドファイルはどんな名前でもよいが .pltや.gplなどの拡張子が使われることが多い
- 論文・レポート用の図を作ったり大量の図を作る場合はこちらが便利

```
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ ls
plot.gpl
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ cat plot.gpl
set term pdfcairo
set output "x.pdf"
plot x
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ gnuplot plot.gpl
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ ls
plot.gpl x.pdf
hinohara.nobuo.ga@physicslinux1:~/gnuplot$
```

gnuplotにわたすコマンドのリストを書いたファイルをあらかじめ用意
(最後のexitまたはquitは不要)

gnuplot plot.gpl

ファイルを引数にしてgnuplotを実行
エラーがなければ標準出力には何も表示されない

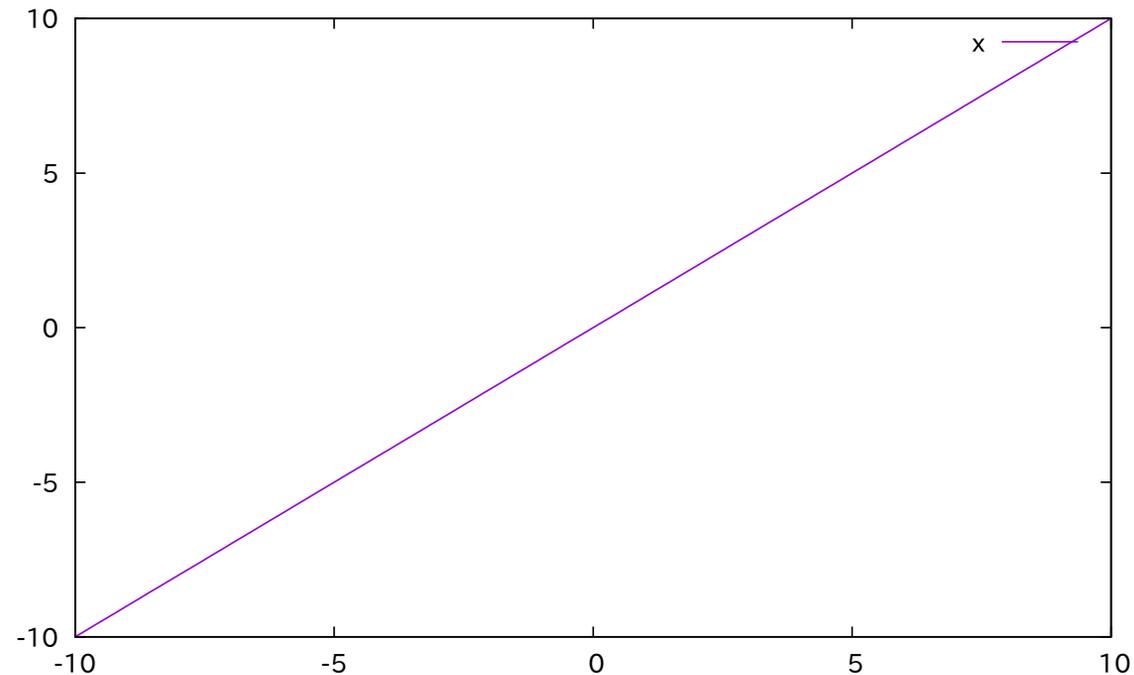
pdfファイルが作成される

gnuplotをインタラクティブに起動したあと load "plot.gpl"としても読み込むことができる。

図の表示

ローカルコンピュータにpdfファイルをscpで転送してAdobe Readerなどで表示
TeXのfigure環境に貼り付けることもできる。

前のページで作った $y=x$ のグラフ



gnuplotのコマンド

```
set term pdfcairo font "Arial,20"
set output "x.pdf"
set xrange[-10:10]
set yrange[-5:5]
set xlabel "x"
set ylabel "y"
set key left top
set border linewidth 3
plot 2*x title "y=2x" linewidth 3,\
     2*sin(x) t "y=2sin(x)" lw 2 dt (10,10)
```

フォントをArialでサイズを20に指定
(デフォルトのサイズは少し小さい)

xやyの描画領域を変更

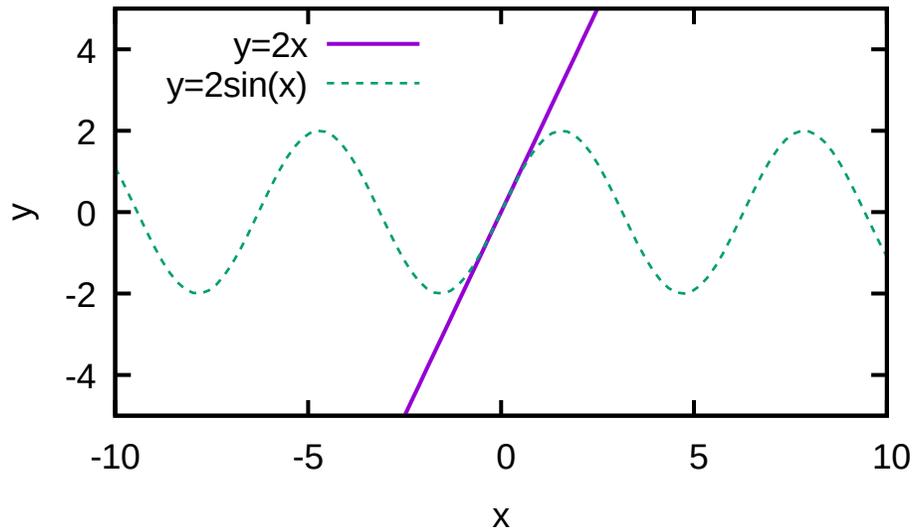
x軸、y軸に名前をつける

凡例の位置を指定(leftまたはright, topまたはbottom)

図の枠の線幅を3倍に

set .. の順番はplotより前であれば任意

2*xをy=2xという凡例、線幅3倍でプロット
2*sin(x)をy=2sin(x)という凡例、線幅2倍、
破線(線を描く長さとして描かない長さが10対10)で
プロット



plotで複数の線をプロットするときはカンマで区切る
カンマの後の¥は改行を表す。

(¥のあとにスペースがあるとエラーになる)

改行せず続けて一行で書いてもよい。

plot → p linewidth → lw,

title → t, dashtype → dtなどの省略が可能

データのプロット

- gnuplotではデータのプロットが可能。(数値計算ではこちらをよく使う)
- この授業の後半で行う数値計算で空白で区切られている数列あるデータを作成
このデータからグラフを作成できる
- 以下のデータを講義資料のページからダウンロード
 - <https://wwwnucl.ph.tsukuba.ac.jp/~hinohara/compphys2-20/doc/data.txt>

```
hinohara.nobuo.ga@physicslinux1:~/gnuplot$ cat data.txt
0 0 0 0
1 1 1 0
2 4 2 0
3 9 3 1
4 16 4 1
5 25 5 1
6 36 4 1
7 49 3 1
8 64 2 0
9 81 1 0
10 100 0 0
```

データのプロット

このコマンドファイルでは複数のpdfファイルを順番に作成

```
hinojara.nobuo.ga@physicslinux1:~/gnuplot$ cat plotdata.gpl
set term pdfcairo
set output "data1.pdf"
plot "data.txt" using 1:2 with points

set output "data2.pdf"
plot "data.txt" using 1:3 with lines

set output "data3.pdf"
plot "data.txt" using 1:4 with linespoints

set output "data4.pdf"
plot "data.txt" u 1:3 w lp,\
      "data.txt" u 1:4 w lp,\
      "data.txt" u 1:($3)+($4) w lp
```

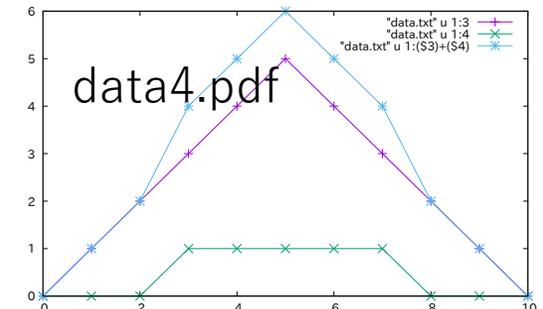
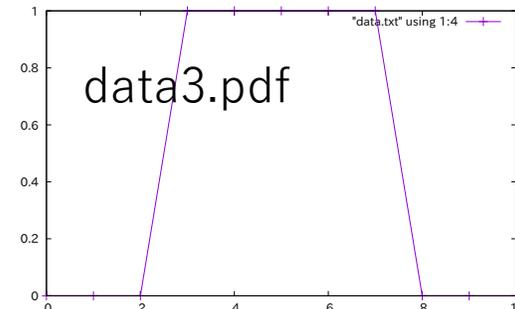
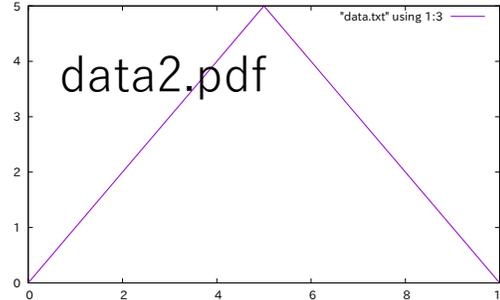
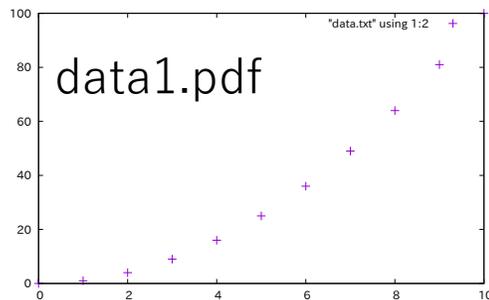
1列目をx, 2列目をyとしてデータを点で描画し data1.pdfに保存

1列目をx, 3列目をyとしてデータ点を線で結んで線を描画し data2.pdfに保存

1列目をx, 4列目をyとしてデータを点で、データを結んだ線も描画し data3.pdfに保存 (データが図の端にあるのでこのままではちょっと見にくい)

using → u, with lines → w l, with points → w p, with linespoints → w lpなどと省略できる。

複数のデータを描画。3つ目はy軸に3列目と4列目の和を描画



gnuplotのその他の機能

- 組み込み関数
 - sin, cos, tan, exp, log, asin, acos, atan, absなどが使用可能
 - 累乗は ** (x^2 は $x^{**}2$)
- すでに設定されたxrangeなどの設定を消去する場合は
 - `set xrange[*:*]` とする(描画領域は次のデータに自動で合わせる)
- 対数目盛
 - `set logscale y` y軸を対数目盛にする
 - `unset logscale y` 元に戻す。
- 図中に文字を書き込む(`set label`), 目盛りの細かい設定(`set xtics/yticks`)、一つのpdfファイルに複数の図を入れる(`multiplot`)、3Dプロット(`splot`)など色々可能。
- インターネットで調べてください。