

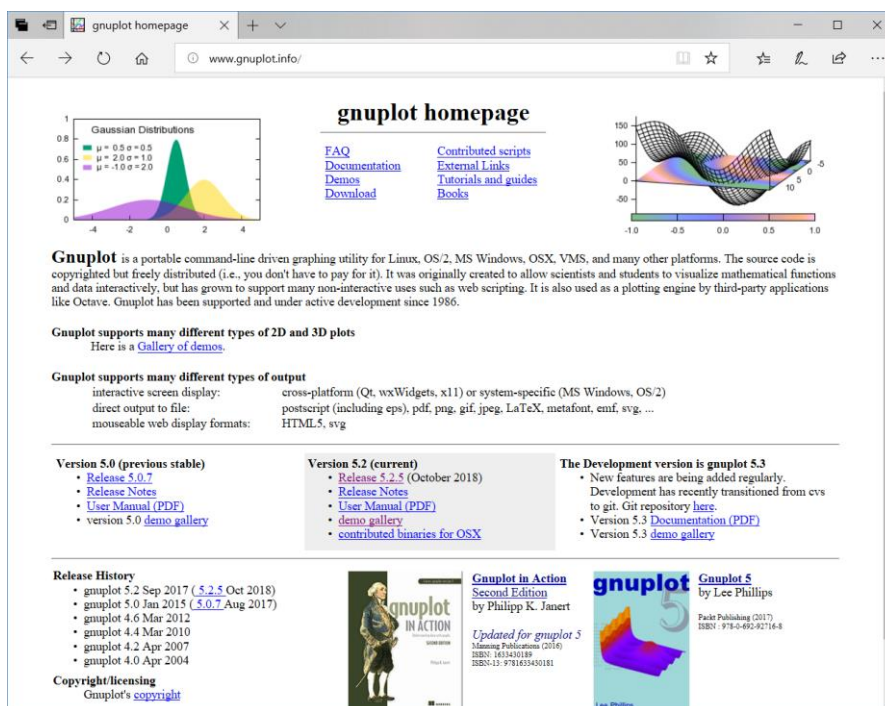
## gnuplot 用スクリプトの使い方

### 1. はじめに

本書に掲載したグラフのいくつかについては、グラフ作成用のアプリケーションである **gnuplot**(version 5.2)を用いて作成している。**gnuplot** にはグラフ作成に関する非常に強力な命令が多数準備されており、それらを使いこなすことで実験データのクイックビューから論文に掲載するような美しい仕上がりのグラフまで作成することができる。グラフを作成する際に用いる複数の命令やパラメータをテキストファイルとしてまとめたものをスクリプトという。**gnuplot** からそのスクリプトを呼び出すことで、複雑なグラフをいつでも作成することができる。本書のグラフを作成した際のスクリプトファイルを使用する際に必要な手順を以下に解説する。なお、以下は Windows 版に対する解説なので、他の OS については適宜読み替えてほしい。また、この解説はコマンドラインを使用したことが無い読者を対象としているが、Linux などコマンドラインになじみのある読者についても、**gnuplot** 特有の書式がいくつかあるため、一度目を通しておくことをお勧めする。

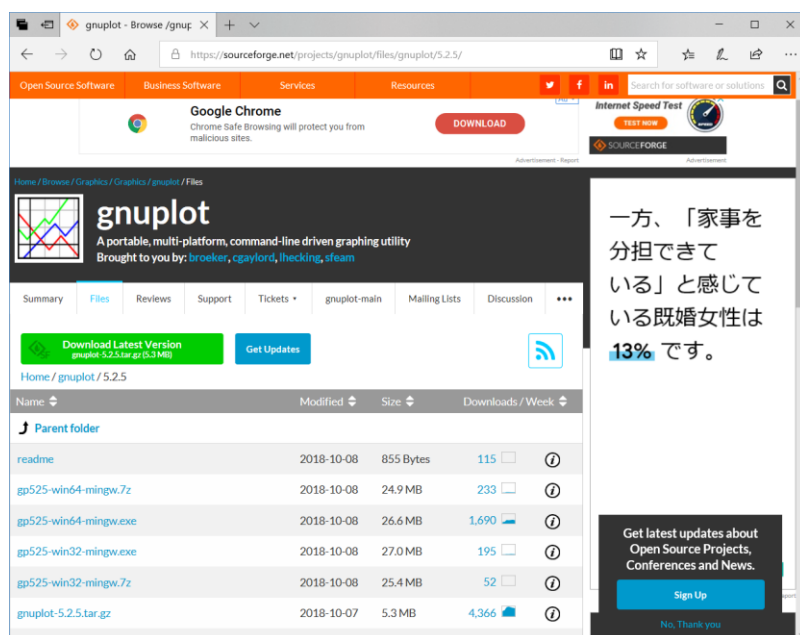
### 2. gnuplot の準備

**gnuplot** はフリーウェアの一つであり、各自が使用しているコンピュータに自由にインストールして使用することができる。**Windows, Mac, Linux** など主要な OS に対応しており、汎用性が極めて高い。**gnuplot** のアプリケーションは **gnuplot homepage** (<http://www.gnuplot.info/>)よりダウンロードできる。

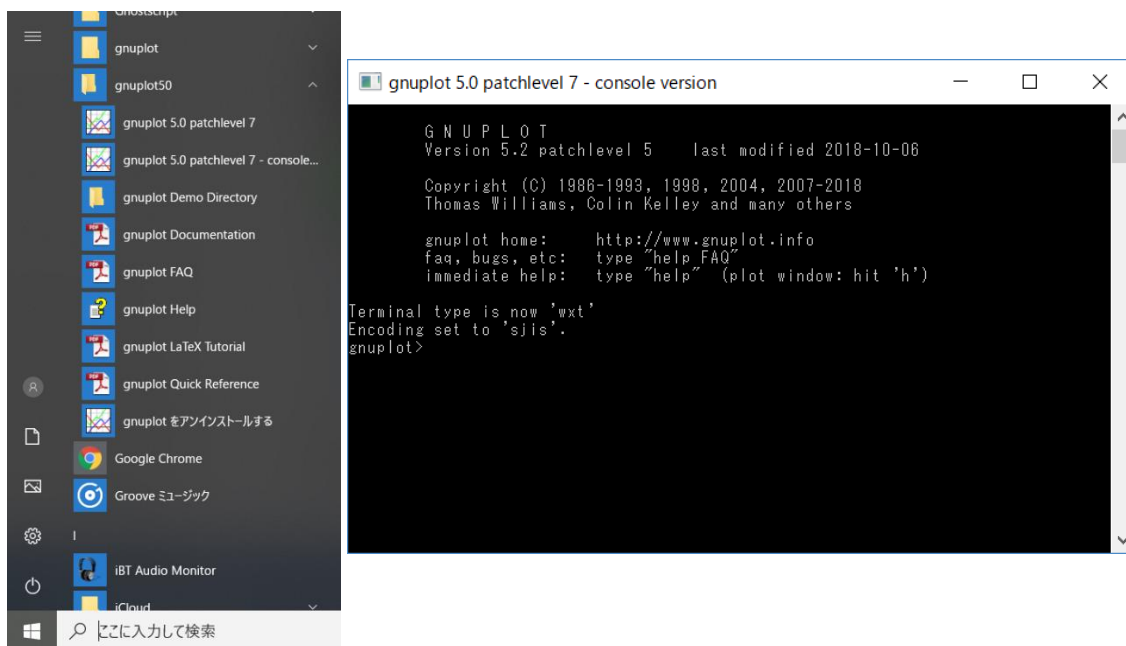


例えば、2018 年 12 月の時点では上記のようなホームページが表示されるが、最新の安

定版である Version 5.2 (current)の Release 5.2.5 をクリックすると、ダウンロードのページ (<https://sourceforge.net/projects/gnuplot/files/gnuplot/5.2.5/>) に移動する。



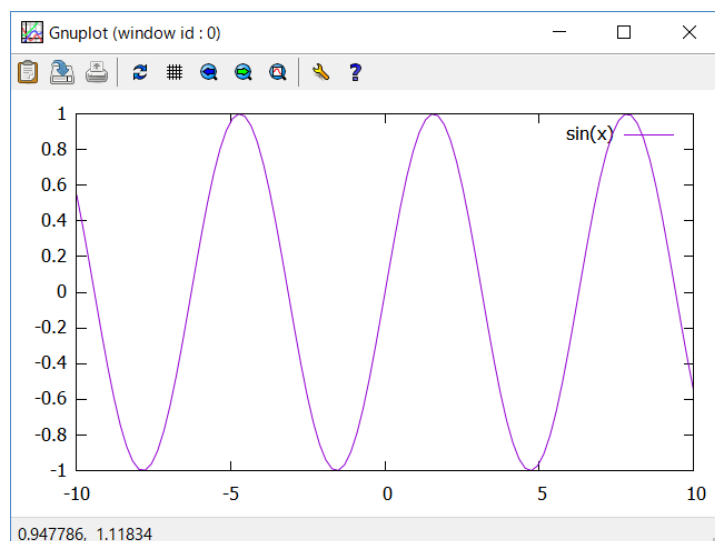
ダウンロードできるファイルがいくつか表示されるが、64bit の windows10 を使用している場合であれば、**gpg525-win64-mingw.exe** を選択する。ダウンロード完了後にそれをダブルクリックすると、インストールが開始される。表示に従って（特にデフォルトから変更をせずに）インストールを完了する。（画面左下の旗のマークを左クリックで表示される）スタートメニューに **gnuplot50** が作られているので、**gnuplot 5.0 patchlevel 7 – console version** を選択して起動すると、コマンドウインドウが表示される。



gnuplot では、このコマンドウィンドウに **gnuplot** のコマンドを打ち込むことでグラフを作成する。ここで、**gnuplot** が正しく動作するかどうか確認するために、コマンドラインの **gnuplot>**に続けて以下のように、

**gnuplot> plot sin(x) [Enter]**

と打ち込んでみよう。ここで、**[Enter]**はエンターキーを押すことを意味する。正しく動作すると、以下のようなグラフが表示される。



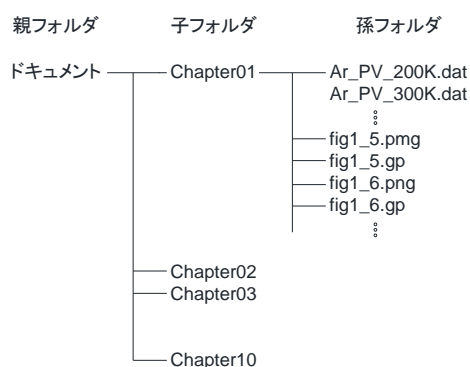
**gnuplot** を終了するときには、コマンドラインの **gnuplot>**に続けて以下のように、

**gnuplot> exit [Enter]**

と打ち込む。

### 3. 本書のスクリプトファイルの実行方法

コロナ社のホームページからダウンロードしたファイル(ITDM01.zip)を各自のドキュメント (Documents) にコピーして展開すると、ITDM という名前のフォルダが作られる。その中には、以下に示すように **Chapter01** から **Chapter10** という名前のフォルダとさらにそれらの中に各章のスクリプトと数値データが収納されている。



ここで、必須ではないがファイルの拡張子を表示するように設定を変えておこう。ITDM フォルダをダブルクリックして開いたウインドウの上の部分に「表示」のタブがあるので、そこをクリックすると以下のようなメニューが表示される。このファイル名拡張子にチェックが入っていないときにはチェックを入れる。



次に、gnuplot を起動しよう。まず作業ディレクトリの場所を確認しよう。コマンドラインに、

```
gnuplot> pwd [Enter]
```

と打ち込むと、現在の作業ディレクトリ（作業フォルダ）が表示される。Gnuplot の初期設定では起動時にはドキュメント（Documents）になるはずである。例えば、私の場合であれば、

```
C:\Users\masaki\Documents
```

となる。この状態でコマンドラインに命令を打ち込むと、ドキュメント内のファイルに対する操作やドキュメントフォルダへの出力が行われる。ここで、作業ディレクトリを先ほどの ITDM へ移動しよう。ディレクトリ間の移動（フォルダー間の移動）のコマンドは **cd**（Change Directory の略）を使用する。cd に続いて、“移動先のフォルダー名”を入力し、Enter キーを押すと、そのフォルダー内が作業ディレクトとなる。

```
gnuplot> cd "ITDM" [Enter]
```

確認のため、もう一度 **pwd** と打ち込むと、

```
gnuplot> pwd [Enter]
```

```
C:\Users\masaki\Documents\ITDM
```

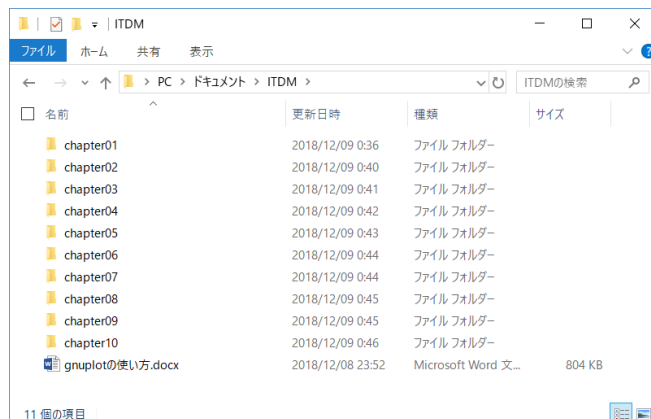
と表示されるはずである。ここで、作業ディレクトリ（作業フォルダ）内のファイルやフォルダーの一覧を表示させてみよう。コマンドラインに、

```
gnuplot> !dir [Enter]
```

と入力すると、以下のように表示される。

```
gnuplot 5.0 patchlevel 7 - console version
C:\Users\tadahiko\Documents\ITDM のディレクトリ
2018/12/09 00:48 <DIR>      .
2018/12/09 00:48 <DIR>      ..
2018/12/09 00:36 <DIR>      chapter01
2018/12/09 00:40 <DIR>      chapter02
2018/12/09 00:41 <DIR>      chapter03
2018/12/09 00:42 <DIR>      chapter04
2018/12/09 00:43 <DIR>      chapter05
2018/12/09 00:44 <DIR>      chapter06
2018/12/09 00:44 <DIR>      chapter07
2018/12/09 00:45 <DIR>      chapter08
2018/12/09 00:45 <DIR>      chapter09
2018/12/09 00:46 <DIR>      chapter10
2018/12/08 23:52      822,678 gnuplotの使い方.docx
1 個のファイル      822,678 バイト
12 個のディレクトリ 40,787,070,208 バイトの空き領域
gnuplot>
```

ドキュメント内の ITDM フォルダと比較すれば、その内容が表示されていることがわかる。



第一章のグラフを作成してみよう。次のコマンドで作業ディレクトリを chapter01 に移動する。

```
gnuplot> cd "chapter01" [Enter]
```

そのフォルダー内には以下のようなファイルが入れられている。

```
gnuplot 5.0 patchlevel 7 - console version

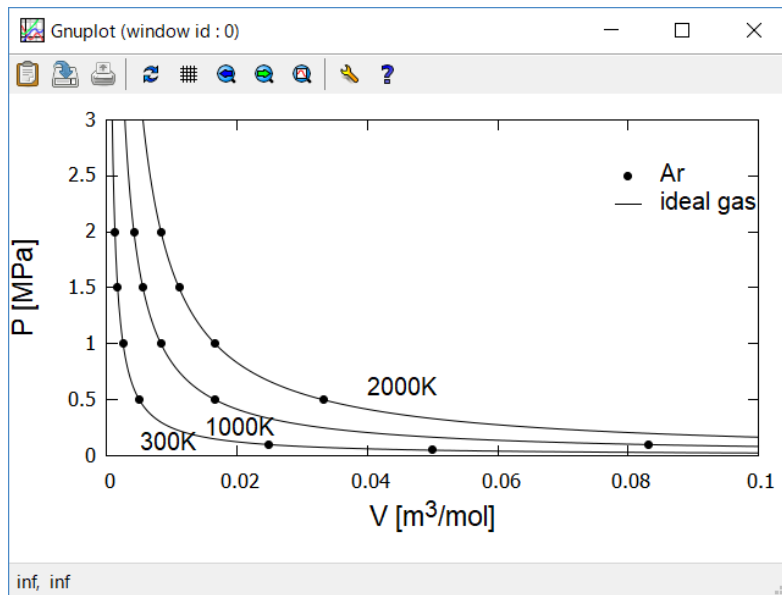
C:\Users\tadahiko\Documents\ITDM\chapter01 のディレクトリ
2018/12/09 15:03 <DIR> .
2018/12/09 15:03 <DIR> ..
2018/03/10 09:50 158 Ar_PV_1000K.dat
2018/03/10 09:54 166 Ar_PV_2000K.dat
2018/03/09 16:47 154 Ar_PV_200K.dat
2018/03/09 16:51 155 Ar_PV_300K.dat
2018/03/10 09:46 157 Ar_PV_500K.dat
2018/03/09 15:31 328 Ar_VT_0_05MPa.dat
2018/03/09 15:32 325 Ar_VT_0_1MPa.dat
2018/03/09 15:41 311 Ar_VT_0_5MPa.dat
2018/03/09 15:12 311 Ar_VT_1_0MPa.dat
2018/12/07 16:19 1,956 fig1_5.gp
2018/12/09 14:51 5,413 fig1_5.png
2018/12/07 16:17 1,878 fig1_6.gp
2018/12/07 16:15 7,029 fig1_6.png
2018/12/07 16:20 1,390 fig1_7.gp
2018/12/06 13:57 8,844 fig1_7.png
2018/12/07 16:21 2,138 fig1_8.gp
2018/12/06 14:02 9,282 fig1_8.png
17 個のファイル 39,995 バイト
2 個のディレクトリ 40,746,835,968 バイトの空き領域

gnuplot>
```

ここで、ファイルの末尾に.gp とあるのが gnuplot のスクリプトファイルである。これを呼び出すにはコマンドの load を使用する。例えば、本書の fig 1.5 を作成するには、対応するスクリプトファイルである fig1\_5.gp を以下のようにして呼び出す。

```
gnuplot> load "fig1_5.gp" [Enter]
```

すると、新たなウィンドウが開かれて以下のグラフが表示される。また、このスクリプトの場合は、同時に同じグラフの画像ファイル (fig1\_5.png) が出力される。なお、スクリプトはテキスト形式で書かれているので、Windows 上で収納されているフォルダーを開き、ワードパットなどのテキストエディタを使用して内容の閲覧や修正を行うことができる。



別の章に移動する際には、少し注意が必要である。コマンドラインでは、子ディレクトリである **Chapter01** から同じ子ディレクトリの **Chapter02** へ直接移動することはできない (\*1) ので、一度親ディレクトリである **ITDM** に移動し、そのあとで **Chapter02** に入る必要がある。具体的には、**Chapter01** から一段上の **ITDM** に移動する際には、以下のコマンドを使用し、

```
gnuplot> cd ".." [Enter]
```

そのあとに

```
gnuplot> cd "chapter02" [Enter]
```

作業ディレクトリの場所がわからなくなった場合は、以下のコマンドを組み合わせて、目的の場所まで移動する。

```
gnuplot> pwd [Enter]
```

```
gnuplot> !dir [Enter]
```

(\*1) 直接移動する際には、`cd` コマンドにフルパスを使用するか、

```
gnuplot> cd "c:/Users/masaki/Documents/ITDM/chapter02" [Enter]
```

親ディレクトリまで含めた行き先を使用する。

```
gnuplot> cd "../chapter02" [Enter]
```

なお、`pwd` などに表示されるディレクトリ名の区切りには“**¥**”が使用されているが、`cd` のダブルクォーテーションの中は“**/**”を使用するので注意が必要である。