

Column Data 作図方法

理化学研究所計算科学研究機構 2014 年 12 月 31 日

1、汎用データフォーマット

例として、添付の history_force.txt の記述内容を示します。

```
... スペースや他の文字列行
...Column_Data_** によりデータブロックを識別します。
    今後、複数のブロックを識別できる予定です。
...Column_Data_00 からデータの最後まで、空行挿入は不可。

Column_Data_00
step    time[sec]      Fx[04]      Fy[04]      Fz[04]
  1    1.666667e-01    5.2481e-02    5.1491e-10    0.0000e+00
  2    3.333333e-01    1.7474e-02   -1.9127e-08    0.0000e+00
  3    5.000000e-01    7.4777e-03   -6.7478e-08    0.0000e+00
  4    6.666667e-01    2.8733e-03   -2.2000e-08    0.0000e+00
  5    8.333334e-01    7.9059e-04    1.9516e-08    0.0000e+00
```

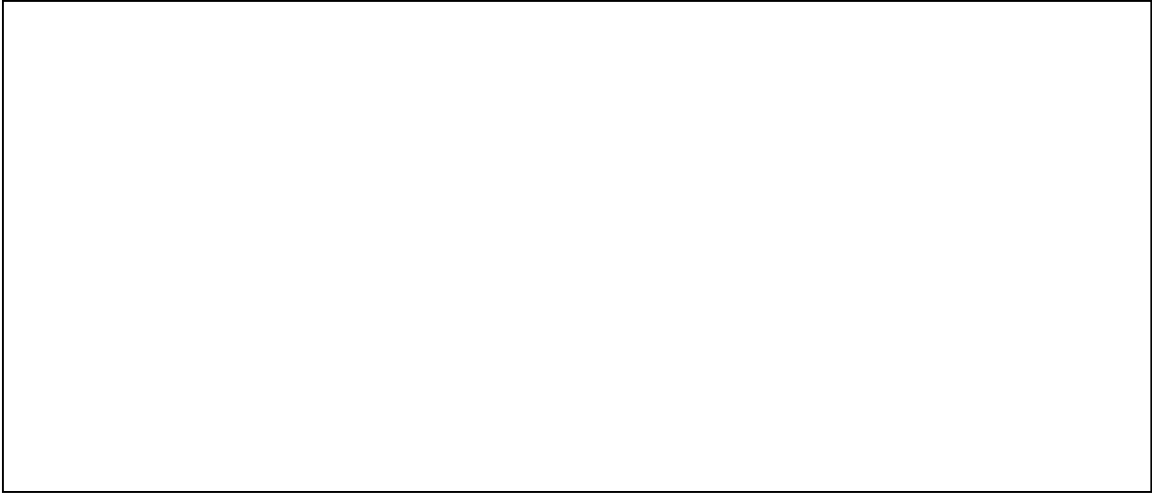
2、Python プログラムの構成

Python スクリプト	説 明
PlotColumnData.py	上記の汎用カラムデータの作図
FileIOColumnData.py	汎用カラムデータを読み込む
PlotOption.py	作図オプションの設定
FileIO.py	FFV Performance Data を読み込む
ParamDef.py	内部のパラメータの定義
PlotMatplotlib.py	MatPlotLib を駆動する部分
PlotPerformanceData.py	FFV Performance Data 作図用
Quantile.py	箱型（BoxPlot）図作図用の計算ツール

Jsage: PlotColumnData.py [options]
記の options の入力順番は自由です。-f と-y は必須のオプションです。
Options:
データ列 index の範囲を指定する際に、必ず "" を使用下さい。例えば、-y "1 2 5-9"
-h, --help show this help message and exit
また Y データ列 index が自動的に分析され、重複指定があったり列数が超過していても構いません。
-f INPUT_FILE, --file=INPUT_FILE (入力ファイル名、省略不可)
役的には、短い options の使用を推奨します。長い options を使用する場合、先頭に--を使用して
input file name
さい。例えば、--file, --xindex, --yindex, --title, --ylabel, --log, --img, --xx, --yy など、
-x X_INDEX, --xindex=X_INDEX (X データ列 index、省略可、Default=0)

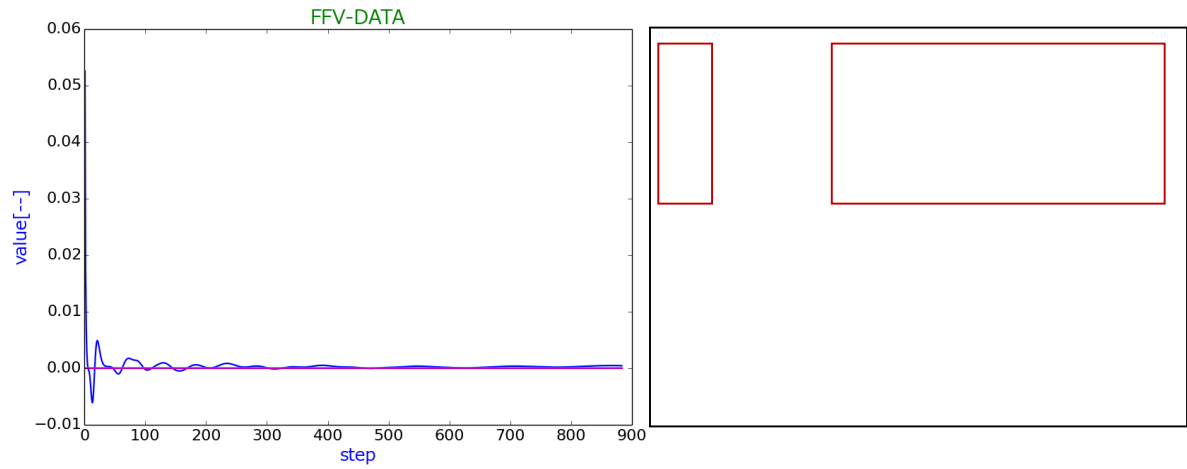
3、作図コマンド
column index used as X data
range の設定は二つの値が必要です、例えば--xrange "0.1 0.3"は、最大値と最小値を示します。
-y Y_INDEX, --yindex=Y_INDEX (Y データ列 index の範囲、省略不可)
ラメータが一つの値の場合、例えば--xrange 3は、暗黙で最小値は0.0になり、入力値は最大値
column index(indices) used as Y data, e.g. 1 2 3-8

よります。--yrange も同様です。
-t TITLE, --title=TITLE (図のタイトル、省略可、Default=入力ファイル名)
title of the plot, by default, it is input file name
-l Y_LABEL, --ylabel=Y_LABEL (Y 軸のラベル)
label of Y axis
--log or --logy (対数 Y 軸のスイッチ、省略する場合、線形 Y 軸)
switch for logarithmic y-scale
--xx or --xrange (X 軸の範囲、省略する場合、データの範囲になる)
range of X axis
--yy or --yrange (Y 軸の範囲、省略する場合、データの範囲になる)
range of Y axis
--img or --image (出力画像の拡張子、省略する場合、png になる)
extension name of output image (eps, pdf, png, ps)
--windize (出力画像のサイズ、省略する場合、10"× 7")
size of the plot window (unit: inch)

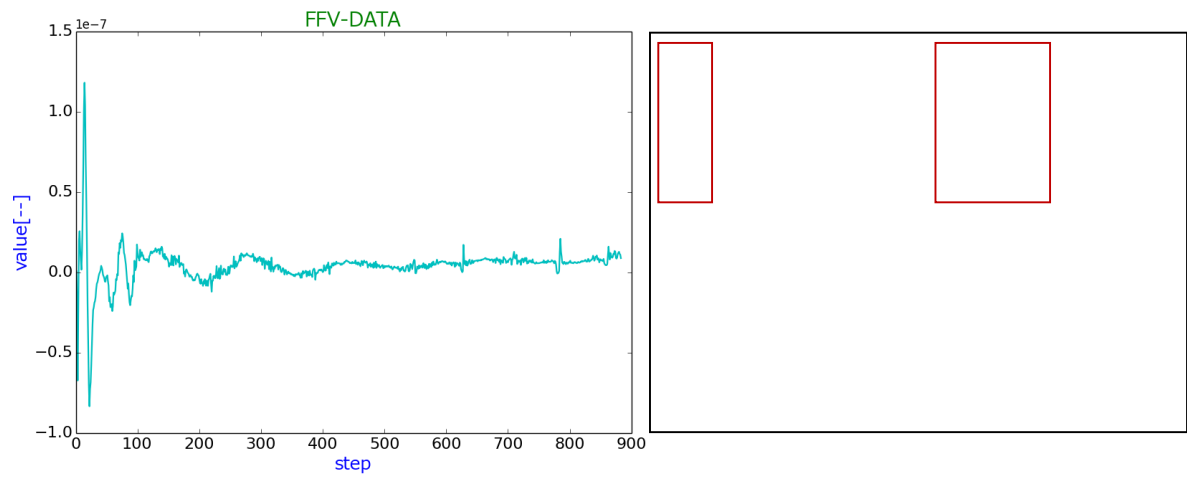


step	time[sec]	Fx[04]	Fy[04]	Fz[04]
1	1.666667e-01	5.2451e-02	5.1491e-10	0.0000e+00
2	3.333333e-01	1.7474e-02	-1.0127e-08	0.0000e+00
3	5.000000e-01	7.4777e-03	-8.7470e-08	0.0000e+00
4	6.666667e-01	2.8733e-03	-2.2000e-08	0.0000e+00
5	8.333334e-01	7.9050e-04	-1.9510e-08	0.0000e+00

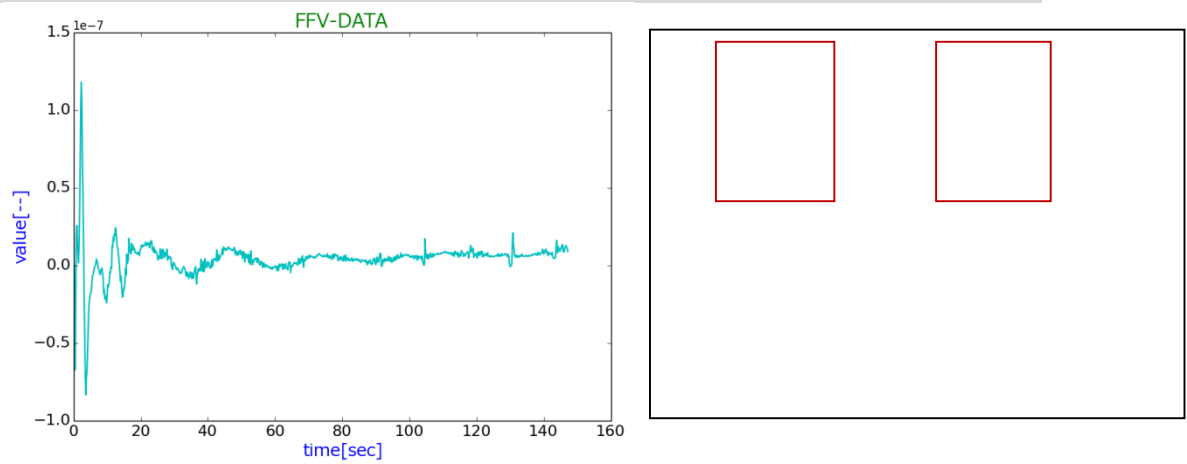
```
python PlotColumnData.py -f history_force.txt -x 0 -y "2 5" -Y value[--] -t FFV-DATA
```



```
python PlotColumnData.py -f history_force.txt -x 0 -y 3 -Y value[--] -t FFV-DATA
```



```
python PlotColumnData.py -f history_force.txt -x 1 -y3 -Y value[--] -t FFV-DATA
```



プロットで1列目のデータの最大値・最小値が表示され、x, y 軸の範囲を設定する。

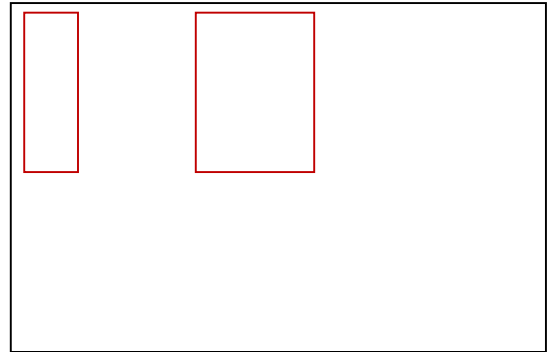
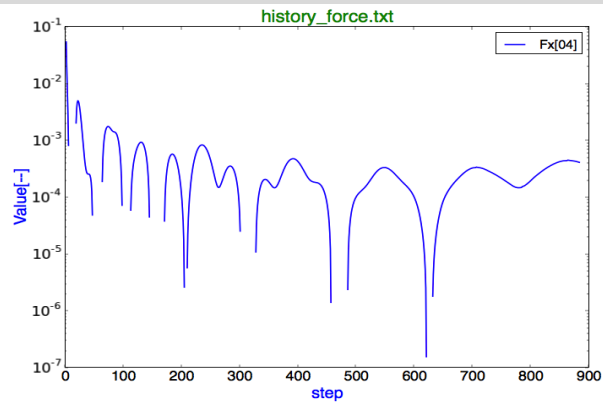
plot: Fx[05] index= 5

X range: min= 1.0 max= 39999.0

Y range: min= 0.0005 max= 9.595200E+00

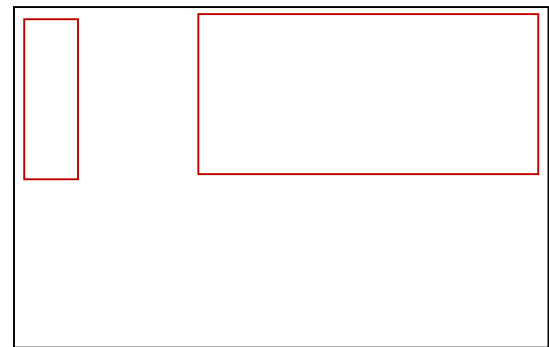
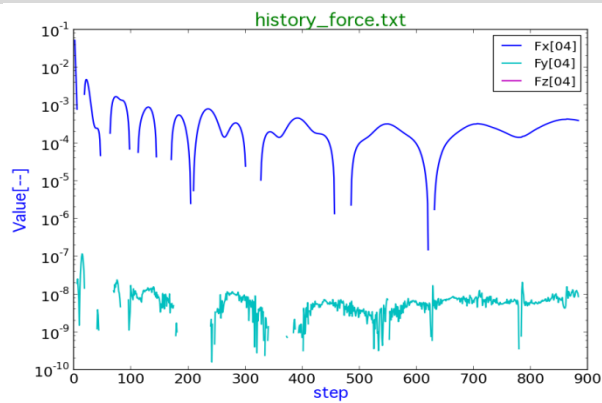
step	time(sec)	Fx[04]	Fy[04]	Fz[04]
1	1.66667E-01	5.2481E-02	5.1491E-10	0.0000E+00
2	3.33333E-01	-1.9127E-08	-1.9127E-08	0.0000E+00
3	5.00000E-01	-6.7277E-08	-6.7277E-08	0.0000E+00
4	6.66667E-01	-2.2000E-08	-2.2000E-08	0.0000E+00
5	8.33333E-01	7.9039E-04	1.9516E-08	0.0000E+00

python PlotColumnData.py -f history_force.txt -x 0 -y "2 2" --log



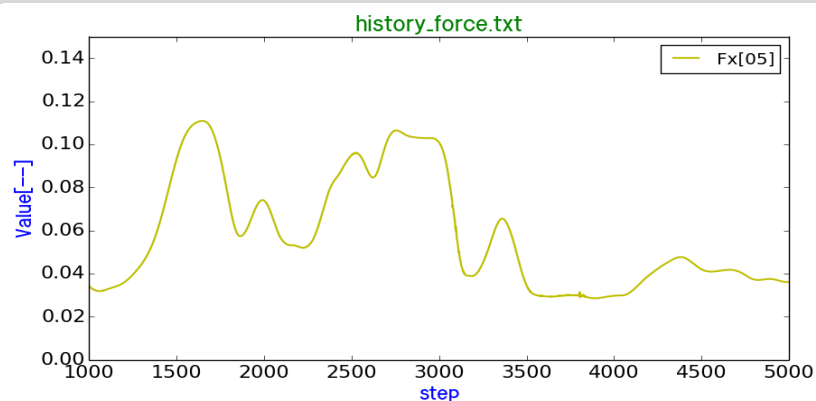
(注: Python2.7.5 使用、Python2.7.2 作図エラー)

python PlotColumnData.py --file history_force.txt -xindex 0 -yindex "2 4" --log



(注: Python2.7.5 使用、Python2.7.2 作図エラー)

PlotColumnData.py --file history_force.txt -x 0 -y 5 -yy 0.15 -xx "1000 5000"



-winsize"5 10"

-winsize"10 5"

```
# 日本語 font の設定
g_fontsize = 18
```

```
if param.G_WINDOWS == 1: # for Windows
    font_path = 'konatu.ttf'
```

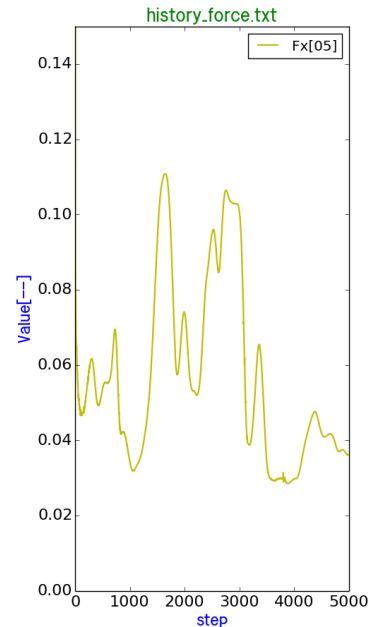
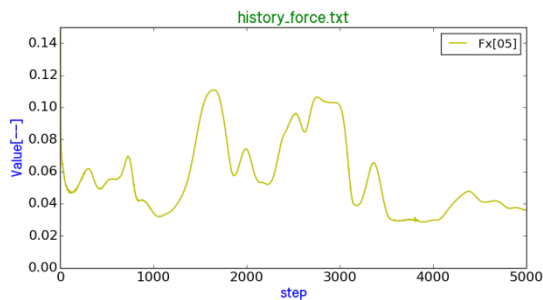
```
# -----
# 『小夏』("Konatu") Copyright (C) 2002～ 榊田道也 All rights reserved. MIT License
# http://avoidnotes.org/urlmemo/cache/20050722191344.html#about
# -----
```

```
elif param.G_MACOSX == 1: # for Mac
    font_path = '/Library/Fonts/Osaka.ttf'
```

```
elif param.G_LINUX == 1: # for Linux
    font_path = '/Please/Specify/ProperFont.ttf'
```

```
elif param.G_LINUX_FOCUS == 1: # for Linux of FOCUS
    font_path = '/Please/Specify/ProperFont.ttf'
```

```
else: # for CentOS
    font_path = '/usr/share/fonts/ja/TrueType/kochi-gothic-subst.ttf'
```



4、補足

① 日本語フォントの設定

PlotMatplotlib.py に各 OS に使用されるフォントのパスが定義されています。



Windows の環境で、日本語フォントを使用する場合、TTF フォントを使用しないと、pdf、eps フォーマットの画像を保存できません。ここで、榊田道也氏の『小夏』("Konatu") フォントを使用します。(Copyright (C) 2002～ 榊田道也 All rights reserved. MIT License)

同梱の konatu.ttf は、Windows, MacOS に使用できます。G_WINDOWS == 1 の場

```

#日本語の文字化けを防止するために、このパラメータを1にしてください。
#出力先がファイルで、エンコードエラーを起す時にこれを回避して下さい。
G_DECODE_STR = 1
png_name = 'image_plot' + graph_title
png_name = verify_filename(png_name)
#プラットフォームによって、下記のいずれか一つを1にしてください。
G_WINDOWS = 1
G_MACOSX = 0 #図仕様を指定する
G_LINUX = 0 #opt2 = PlotOption()
G_LINUX = 0 #opt2.set_figsize(10, 7)
G_LINUX = 0 #opt2.set_backend('macosx', 20, 'green')
#Please do not use the following environment"
#no display environment
#Agg is a good choice for saving figures if you don't want to display
#on the screen. If you are using a desktop environment, it saves to files.
if G_LINUX_K == 1: G_USE_AGG = 1 #Linuxの環境で、フォントのインストールディレクトリは異なる可能性がありますので、
else: #opt2.set_figsize(10, 7) #Linuxの環境でも、『小夏』フォントを使用できる可能性があります。（要確認）
#opt2.set_ticklabelsize(16)
#opt2.set_pngname(png_name)
# -----

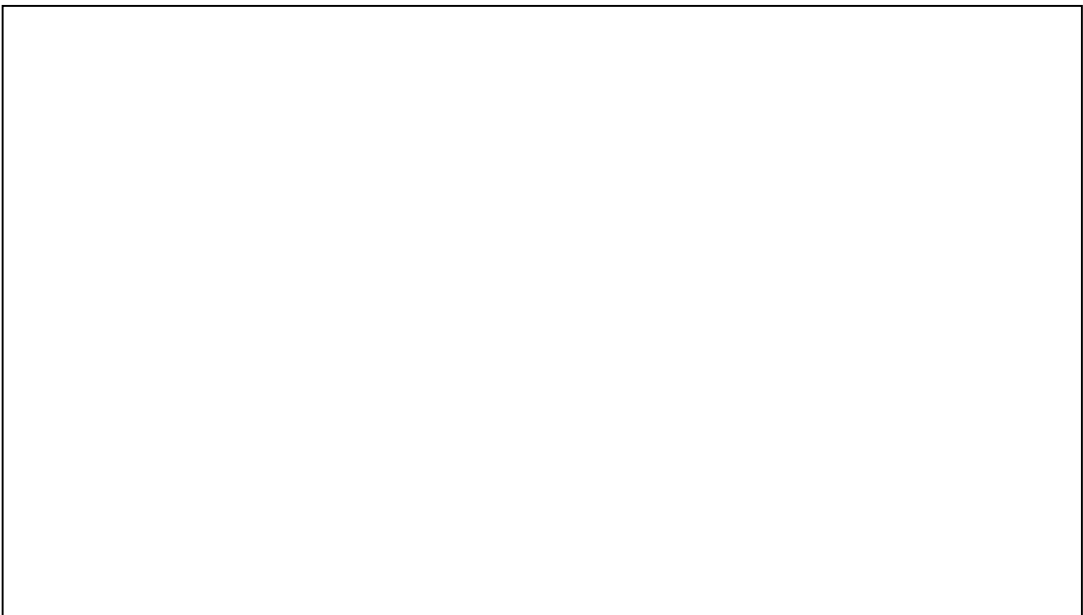
```

② ParamDef.py の設定



③ 作図の仕様の設定

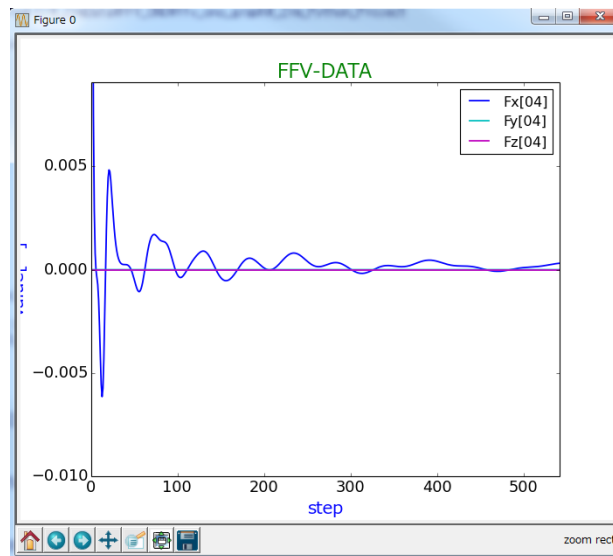
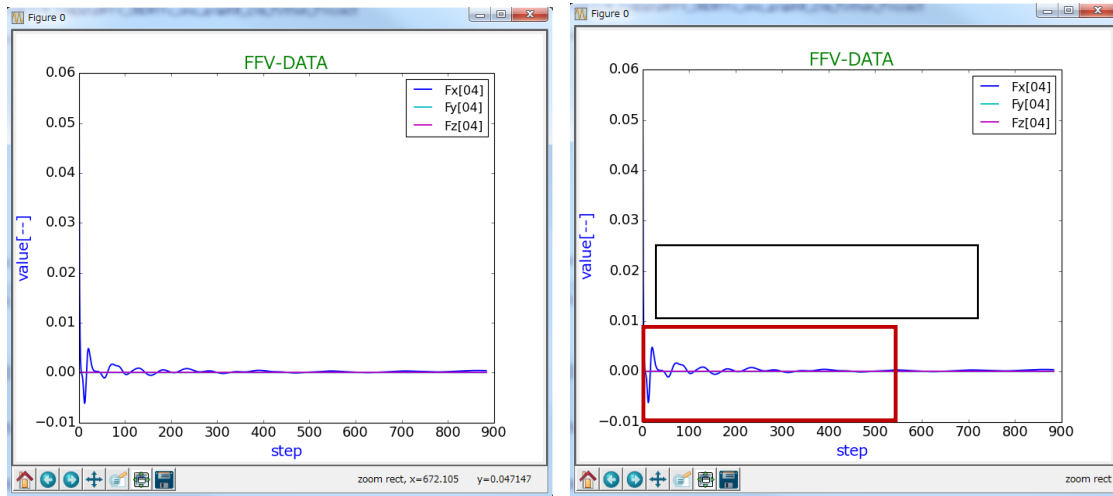
各カラムのデータの範囲が異なるものを同じグラフに入れる場合、見にくくなる可能性がありますので、別々での作図を薦めます。また、[PlotColumnData.py](#) を編集すれば、図の詳細仕様を調整することができます、下の図に赤色の行は修正の一例です。



File "C:\Python27\lib\site-packages\matplotlib\mathtext.py", line 2193, in $\text{brace} + \text{float_literal} + \text{brace}$
TypeError: unsupported operand type(s) for the 描画範囲に戻るNodeがで
指定することができます

④ 図の調整方法

マウスで局部のデータ
を囲む枠を指定する

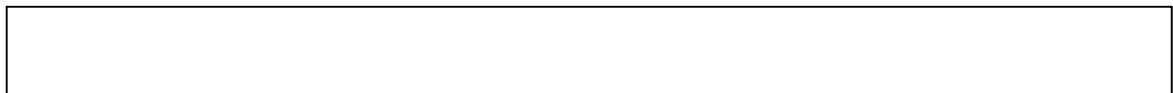


描画データの範囲の調整



⑤ Y 軸は対数軸設定について

Python 2.7.2 でテストしましたが、show()を呼出すと、下記のエラーが発生します：



Python2.7.5 を使用すれば、図が正しく表示されます。