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スクリップト** | **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 |

３、作図コマンド

python PlotColumnData.py　–h　を実行すると、下記のメッセージが表示されます：

Usage: PlotColumnData.py [*options*]

*Options*:

-h, --help show this help message and exit

-f INPUT\_FILE, --file=INPUT\_FILE　 （入力ファイル名、省略不可）

input file name

-x X\_INDEX, --xindex=X\_INDEX （Xデータ列index、省略可、Default=0）

column index used as X data

-y Y\_ INDEX, --yindex=Y\_INDEX （Yデータ列indexの範囲、省略不可）

column index(indices) used as Y data, e.g. 1 2 3-8

-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）

上記の*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。

Y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”

指定した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。

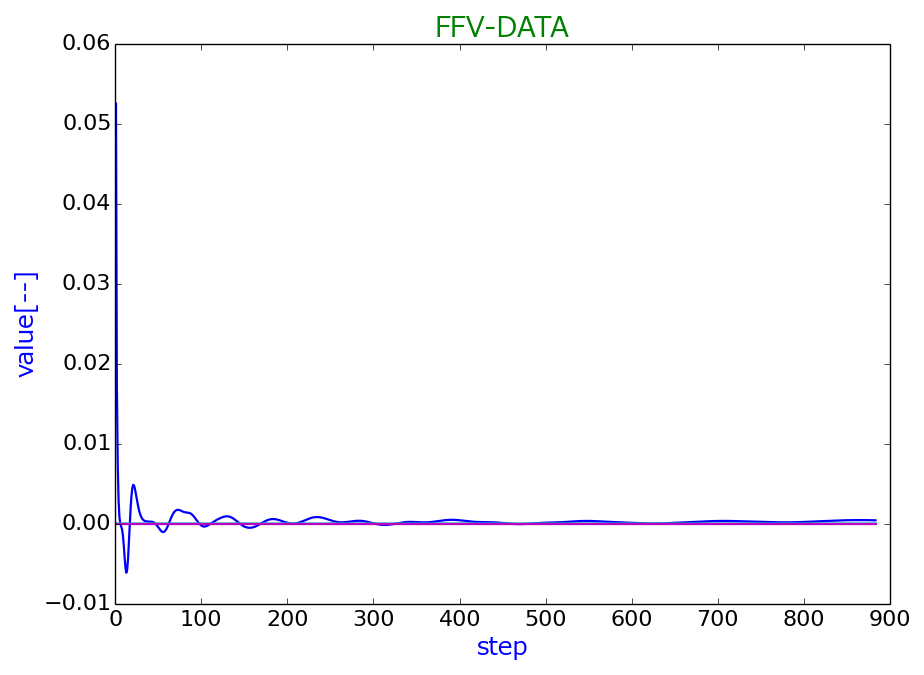
一般的には、短いoptionsの使用を推奨します。長いoptionsを使用する場合、先頭に--を使用して

下さい。例えば、--file, --xindex, --yindex, --title, --ylabel, --log, --img, --xx, --yy など。

--xrange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。

パラメータが一つの値の場合、例えば、--xrange 0.3は、暗黙で最小値は0.0になり、入力値は最大値となります。--yrangeも同様です。

python PlotColumnData.py –f history\_force.txt -x 0 -y “2 5” -Y value[--]　–t FFV-DATA



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

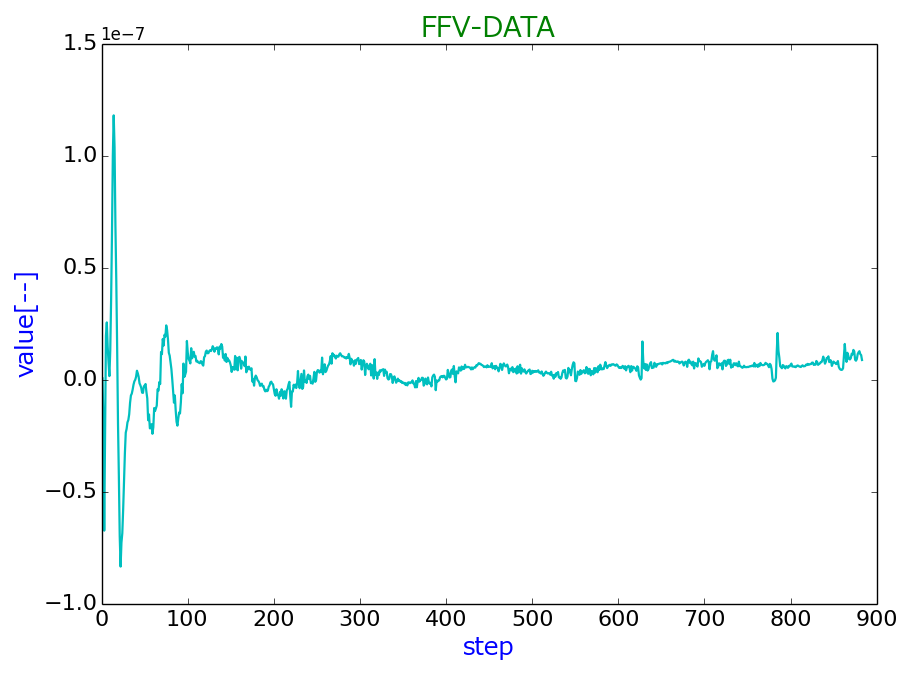
3 5.000000e-01 7.4777e-03 -6.7478e-08 0.0000e+00

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

　… … …

python PlotColumnData.py –f history\_force.txt -x 0 -y 3 -Y value[--] –t FFV-DATA



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

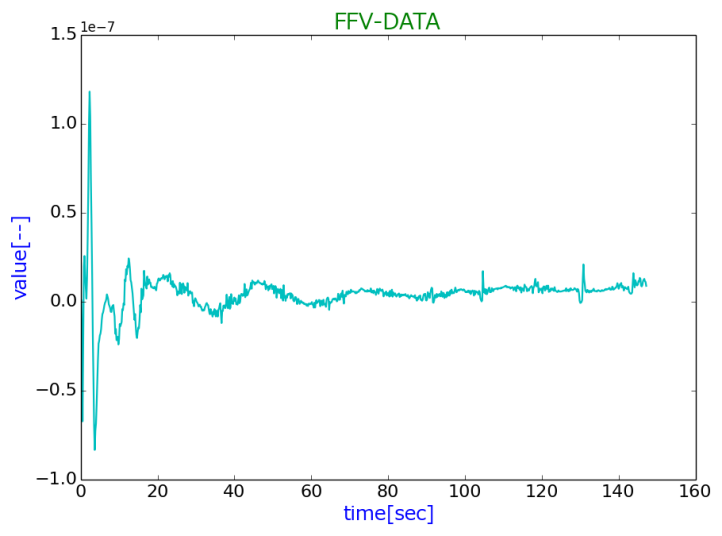
3 5.000000e-01 7.4777e-03 -6.7478e-08 0.0000e+00

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

　… … …

python PlotColumnData.py –f history\_force.txt -x 1 -y3 -Y value[--]　–t FFV-DATA



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

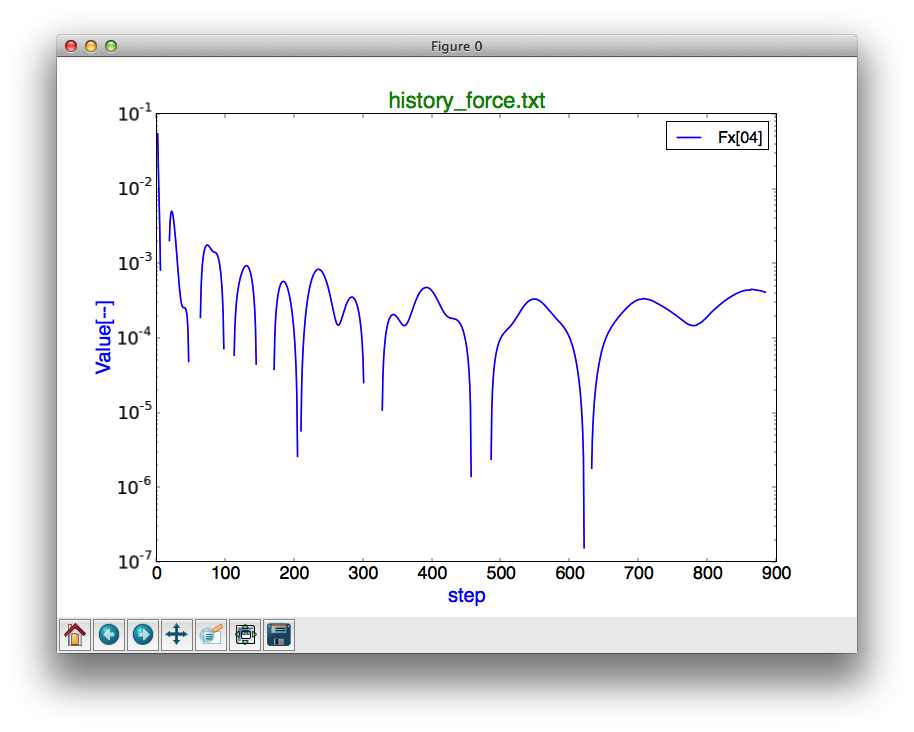
3 5.000000e-01 7.4777e-03 -6.7478e-08 0.0000e+00

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

　… … …

python PlotColumnData.py –f history\_force.txt -x 0 -y “2 2” --log



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

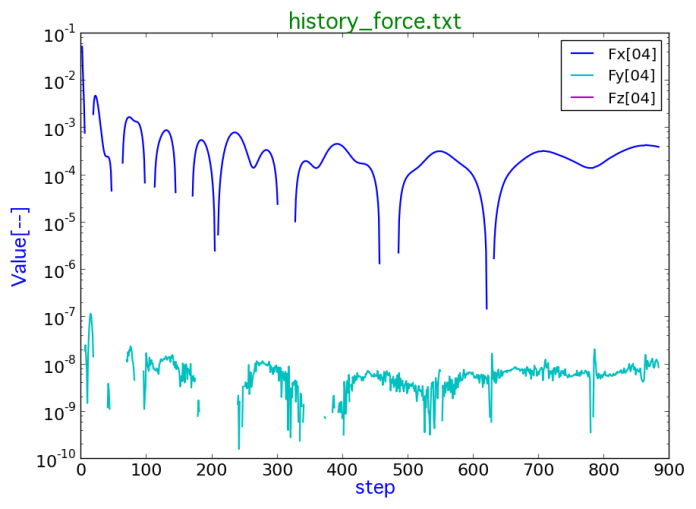
4 6.666667e-01 2.8733e-03 -2.2000e-08 0.0000e+00

5 8.333334e-01 7.9059e-04 1.9516e-08 0.0000e+00

　… … …

（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



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

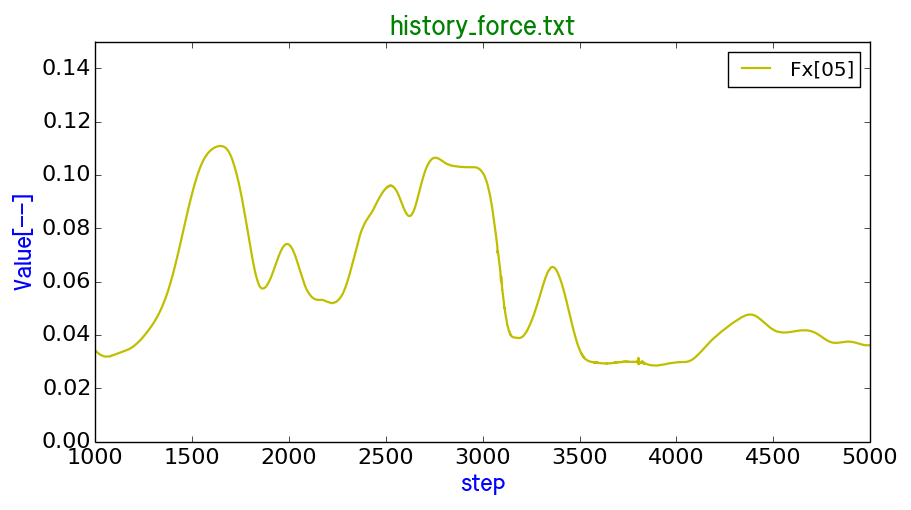
4 6.666667e-01 2.8733e-03 -2.2000e-08 0.0000e+00

5 8.333334e-01 7.9059e-04 1.9516e-08 0.0000e+00

　… … …

（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”



プロットされたデータの最大値・最小値が表示され、X,Y軸の範囲を設定する例です。

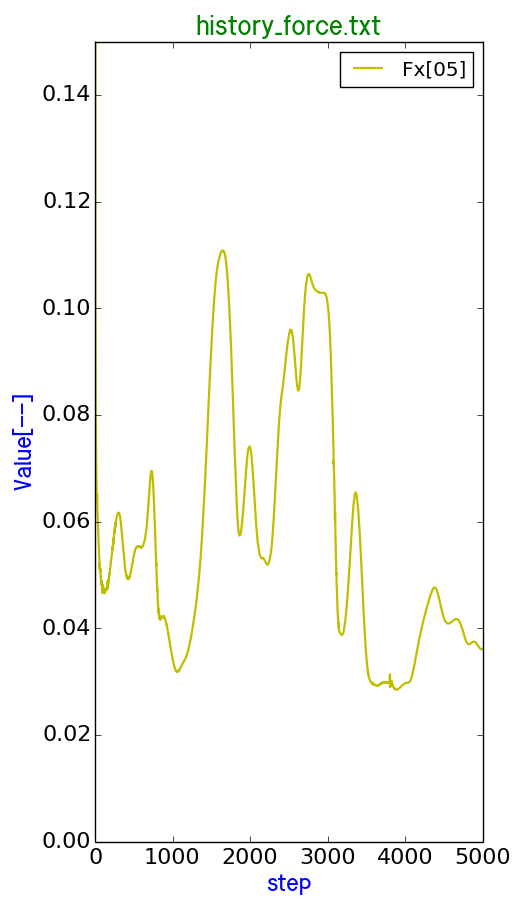
plot: Fx[05] index= 5

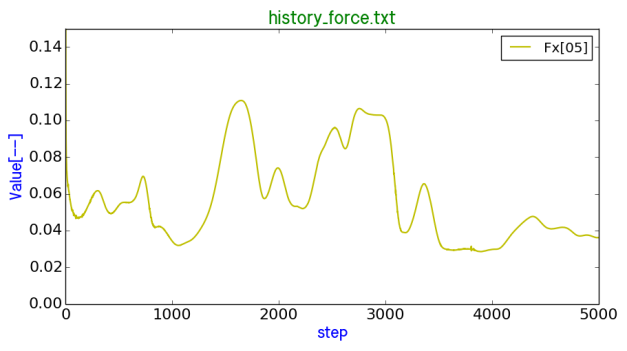
X range: min= 1.0 max= 39999.0 avg= 20000.0

Y range: min= 0.0005 max= 9.595 avg= 0.047

PlotColumnData.py --file history\_force.txt -x 0 - y 5 –yy 0.15 –xx 5000 –winsize”10 5”

　PlotColumnData.py --file history\_force.txt -x 0 - y 5 –yy 0.15 –xx 5000 –winsize”5 10”





–winsize”5 10”

–winsize”10 5”

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。

# 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～ 桝田道也 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\_K == 1 : # for Linux of K-Computer

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'

　　　　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、桝田道也氏の『小夏』("Konatu") フォントを使用します。（Copyright (C) 2002～ 桝田道也 All rights reserved. MIT License）

　　　　同梱のkonatu.ttfは、Windows, MacOSに使用できます。G\_WINDOWS == 1の場合、Mac OSXの環境でも、プログラムは正常動作します。（確認済）

Linuxの環境で、フォントのインストールティレクトリは異なる可能性がありますので、確認してから、上記のパスを設定して下さい。一般のLinuxの環境でも、『小夏』フォントkonatu.ttfを使用できる可能性があります。（要確認）

②　ParamDef.pyの設定

#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にして下さい。

#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0にして下さい。G\_DECODE\_STR = 1

#プラットフォームによって、下記のいずれか一つを１にして下さい。

G\_WINDOWS = 1

G\_MACOSX = 0

G\_LINUX = 0

G\_LINUX\_K = 0

G\_LINUX\_FOCUS = 0

#Please use 'Agg' If it issues following error:

#"no display name and no $DISPLAY environment"

#Agg is a non-interactive backend, it won't display

#on the screen, Show() does not work, it saves to files.

if G\_LINUX\_K == 1 : G\_USE\_AGG = 1

else: G\_USE\_AGG = 0

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。

############ 作図仕様 ##########################

#----------------------------------------------

#出力PNG画像ファイル名中に使えない文字を '\_' に置換

png\_name = 'image\_plot' + graph\_title

png\_name = verify\_filename(png\_name)

png\_name = png\_name + '.png'

# 作図仕様を指定する

opt2 = PlotOption()

opt2.set\_figsize( 10, 7 )

opt2.set\_title( graph\_title, 20, 'green' )

opt2.set\_label( x\_label, y\_label, 18, 'blue' )

#opt2.set\_xrange( 0, n\_max )

#opt2.set\_yrange( 0, 0.025 )

#opt2.set\_draw( 'b', 'k', 0.6 )

#opt2.set\_text( graph\_title, 0.80, 0.90, 16, 'red', 0.3 )

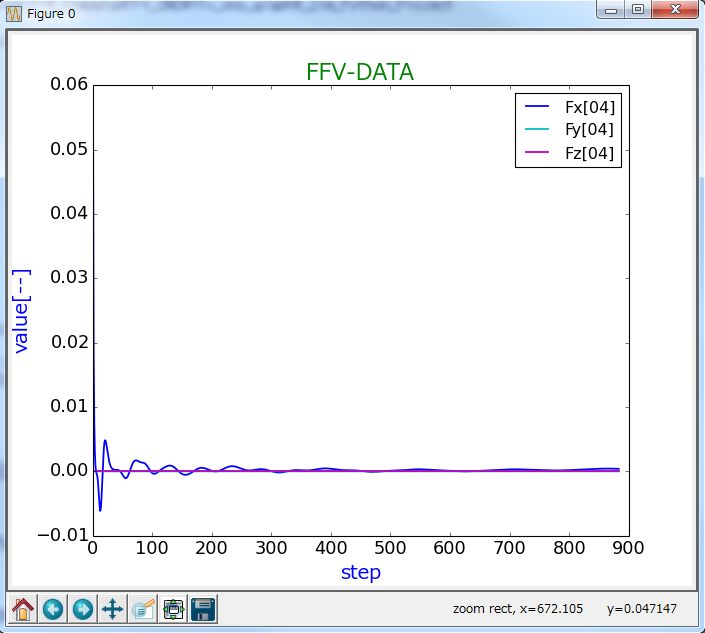
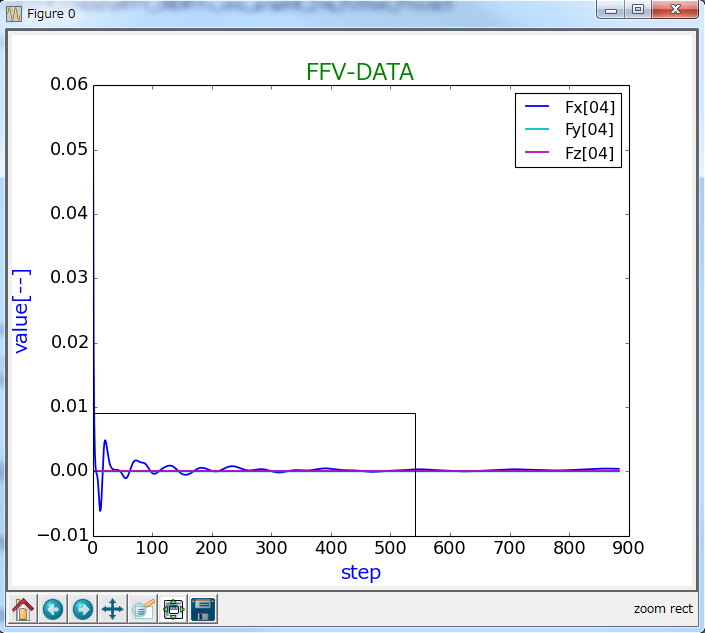
#opt2.set\_figcolor( 'lightgoldenrodyellow', 0.2 )

#opt2.set\_tick( 'red', 16, 0, 0 )

opt2.set\_pngname(png\_name)

# ------------------------------------------------

　　④　図の調整方法

マウスで局部のデータ

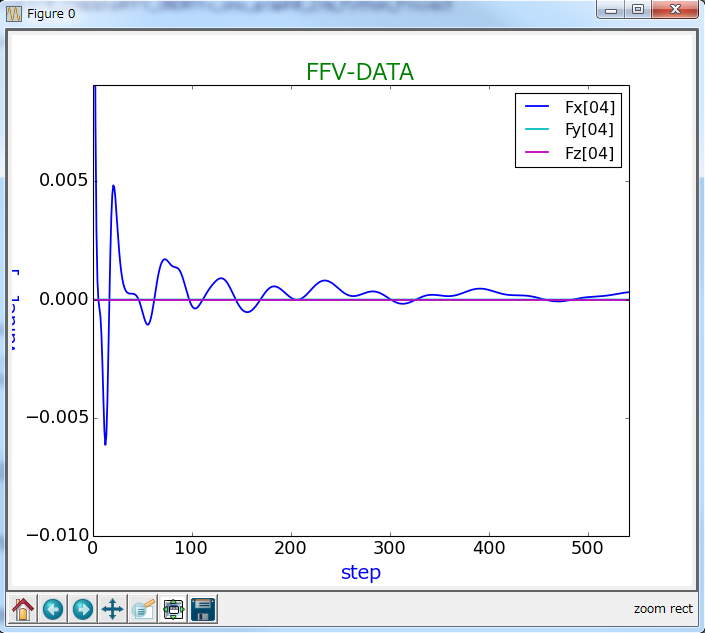
を囲む枠を指定する

を押して、描画範囲を

指定することができます



次の頁の図のように表示されます



描画データの範囲の調整

を押して、元の描画範囲に戻ることができます。

　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：

File "C:\Python27\lib\site-packages\matplotlib\mathtext.py", line 2193 - ((lbrace + float\_literal + rbrace)

TypeError: unsupported operand type(s) for +: 'NoneType' and 'NoneType'

　　　Python2.7.5を使用すれば、図が正しく表示されます。