OpenFOAM ログファイルからの情報取り出しなど

オープンCAE勉強会@富山　June 18, 2016,　富山県立大学　中川慎二

まえがき

Disclaimer: OPENFOAM® is a registered trade mark of OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM ® and OpenCFD ® trade marks. This offering is not approved or endorsed by OpenCFD Limited.

OpenFOAMユーザーガイド，プログラマーズガイド，OpenFOAM Wiki，CFD Online，その他多くの情報を参考にしています。開発者，情報発信者の皆様に深い謝意を表します。

この講習内容は，講師の個人的な経験（主に，卒研生等とのコードリーディング）から得た知識を共有するものです。この内容の正確性を保証することはできません。この情報を使用したことによって問題が生じた場合，その責任は負いかねますので，予めご了承ください。

本文書での表記方法について

端末から入力するコマンド

端末（ターミナル）で実行するコマンドは，次のように表記する。

```
cp a b
```

ファイルやソースコードの内容

ファイル・ソースコード記載事項は次のように表記する。インデント（字下げ）は，必ずしもここに記載通りとは限らない。記入するファイルに合わせて，適切にインデントしてください。

```
    solve
    (
        fvm::ddt(T)
        - fvm::laplacian(DT, T)
    );
```

Table of Contents

目的

linux の基本的なコマンドを使って，OpenFOAM 実行後のポスト処理等を効率的に実施する方法を学ぶ。

主に，ログファイルの処理を考える。OpenFOAM 付属のfoamLogユーティリティだけでは取り出せない情報の処理を行なう。

[目次に戻る]

要約

floatingObject の実行時ログファイルから、時刻の行だけを抜き出すコマンド

```
cat log.interDyMFoam | grep -v 'ExecutionTime' | grep 'Time = '
```

floatingObject の実行時ログファイルから、Center of rotation の座標値を抜き出すコマンド。ただし、各時刻で3回繰り替えされているので、3回目だけを抜き出す。

```
cat log.interDyMFoam | grep 'Centre of rotation'| cut -d' ' -f8-10 | tr -d '(' | tr -d ')' | sed '0~3!d'
```

上記は次のリンク先を参照した。

http://www.cfd-online.com/Forums/openfoam-solving/64146-tutorial-how-plot-residuals-3.html#post520028

何行目を抜き出しているかは、下記コマンドで確認できる

```
seq 10 | sed '0~3!d'
```

参照

http://stackoverflow.com/questions/9894986/how-can-i-delete-every-xth-line-in-a-text-file

http://www.cfd-online.com/Forums/openfoam-solving/64146-tutorial-how-plot-residuals-3.html#post520032

[目次に戻る]

パイプライン処理について

Linuxの端末上で，複数のコマンドを組み合わせて連続処理することで，単純なコマンドによって複雑な処理を実行することが可能となる。

あるコマンドの出力を，別のコマンドの入力に引き渡して連続的に処理を実行することを，パイプライン処理と呼ぶ。

複数のコマンドは，記号 | を使って組み合わせる。これによって，前のコマンドの出力を，次のコマンドの入力に渡すことができる。

[目次に戻る]

時刻行に関する操作の例

例題 multiphase/interDyMFoam/ras/floatingObject 例題を，./Allscript コマンドにより実行したとする。計算実行時のログファイルが，log.interDyMFoam というファイルに書き出されている。このファイルは，多くの行から構成されている。このファイルから，必要な情報だけを抜き出すことが目的である。

まず，ファイルの内容を端末に表示する。そのために，cat コマンドを実行する。log.interDyMFoam の存在するディレクトリにおいて，下記コマンドを実行する。端末上に，ファイル内容が次々と表示される。

```
cat log.interDyMFoam
```

上記のコマンドを，パイプライン処理の先頭にする。そうすると，先ほど端末上に表示された内容が，次のコマンドに渡されることになる。

次のコマンドには，入力から任意のパターンに合致する行を抜き出すコマンド grep を使う。例えば，時刻を表示している行に含まれる 'Time =' という表現を指定して，grepを実行するとする。下記のコマンドを，端末上で実行する。

```
cat log.interDyMFoam | grep 'Time ='
```

上記コマンドを実行すると，ログファイルの中で，'Time =' という表現が出現する行だけが表示されるはずである。

今回の例では，計算上の時刻を表す行と，計算に要した時間を表す行とが，交互に表示される。この結果から，計算に要した時間の行を除くにはどうすれば良いだろう。そのためには，grep のオプションである -v, –invert-match（逆マッチ）を使う。下記のコマンドを実行して，その働きを確認する。

```
cat log.interDyMFoam | grep -v 'ExecutionTime'
```

上記2つの grep を組み合わせて実行すれば，計算上の時刻だけを表す行が取り出せる。

```
cat log.interDyMFoam | grep -v 'ExecutionTime' | grep 'Time = '
```

このようにして取り出した結果をファイルに保存したい場合には，下記のようにして実行する。ここで，リダイレクト > と呼ばれる方法を使っている。

```
cat log.interDyMFoam | grep -v 'ExecutionTime' | grep 'Time = ' > timeFile
```

[目次に戻る]

Center of rotation 座標の取り出し

floatingObject の実行時ログファイルから、Center of rotation の座標値を抜き出す方法を考える。前節と同様に，下記コマンドを実行する。リダイレクトにより，ファイルに書き出す。

```
cat log.interDyMFoam | grep 'Centre of rotation' > centerOfRotation01
```

この結果として作成されたファイルと，先に作成した時刻ファイルの行数を比べる。時刻の行数よりも，centerOfRotationの行数が3倍多い。これは，元のlogファイルにおいて，1時刻に対して PIMPLE: iteration が3回実行されるためである。

1時刻の最後の Centre of rotation 情報だけを取り出す方法を考える。これには，sed の d コマンドが使える。dコマンドは，行を削除するものである。0行目から始めて，3行目以外を削除する場合には，`sed '0~3!d'` となる。

まず，この `sed '0~3!d'` の働きを確認する。確認用のデータとして，連続番号を作り出すコマンド seq を利用する。下記コマンドを実行すると，端末に1から10までの数字が出力される。

```
seq 10
```

この出力に対して，3行目だけを残すように，下記を実行する。

```
seq 10| sed '0~3!d'
```

3行目毎に残される（それ以外の行が削除される）ことが確認できる。

上記を組み合わせることで，各時刻で繰返される3回の内の最後の情報だけを取りだすことが可能となった。

```
cat log.interDyMFoam | grep 'Centre of rotation'| sed '0~3!d' > centerOfRotation02
```

centerOfRotation02 のファイルを開いて，行数を確認する。時刻と同じ数になっているはずである。

このままでは，数字以外の不要な情報も含まれている。ここから，必要な数字だけを抜き出す。そのために，cutコマンドを使用する。区切り文字を空白' 'と指定し，区切られた8番目から10番目のフィールドを取り出すためには，cut -d' ' -f8-10 として使う。フィールドの番号については，図1を参照してください。

```
cat log.interDyMFoam | grep 'Centre of rotation'| cut -d' ' -f8-10
```

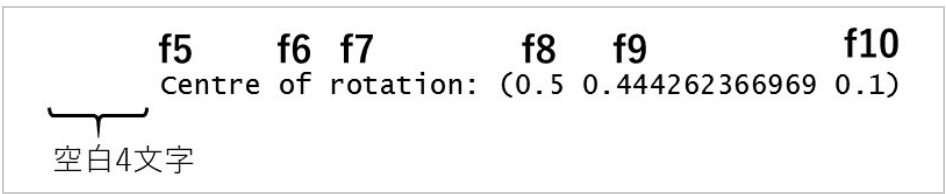

図1　行の中のフィールド番号(区切り記号を空白とした場合)

さらに，ここから，括弧記号を取り去るためのコマンド tr -d '(' などを組み合わせて，下記コマンドを実行する。

```
cat log.interDyMFoam | grep 'Centre of rotation'| cut -d' ' -f8-10 | tr -d '(' | tr -d ')' | sed '0~3!d' > centerOfRotation03
```

このようにして，ログファイルから必要な情報を取り出すことができる。取り出した結果を使うことで，簡単にグラフを作成することができる。

[目次に戻る]

注意

logファイルから一般的な情報をグラフ用に変形するのは，foamLog コマンドが使える。

PyFoamでも，同様な機能あり。

[目次に戻る]

各コマンドについて　（各コマンドのhelp情報）

[目次に戻る]

cat

```
使用法: cat [オプション]... [ファイル]...
ファイル、または標準入力を連結し、標準出力に出力します。
  -A, --show-all           -vETと同じ
  -b, --number-nonblank    空行を除いて行番号を付け加える。-n より優先される
  -e                       -vEと同じ
  -E, --show-ends          行の最後に $ を付け加える
  -n, --number             全ての行に行番号を付け加える
  -s, --squeeze-blank      連続した空行の出力を抑止する
  -t                       -vTと同じ
  -T, --show-tabs          TAB文字を`^I'で表示
  -u                       (無視)
  -v, --show-nonprinting   非表示文字と`^'や`^'を付けて表示 (LFDとTABは除く)
      --help     この使い方を表示して終了する
      --version  バージョン情報を表示して終了する

ファイルの指定がなかったり, - であった場合, 標準入力から読み込みます.

例:
  cat f - g  最初に f の中身を出力し、次に標準入力を出力します。
             そして g の中身を出力します。
  cat        標準入力を標準出力に複製します。

cat のバグを発見した場合は bug-coreutils@gnu.org に報告してください。
GNU coreutils のホームページ: <http://www.gnu.org/software/coreutils/>
GNU ソフトウェアを使用する際の一般的なヘルプ: <http://www.gnu.org/gethelp/>
cat の翻訳に関するバグは <http://translationproject.org/team/ja.html> に連絡してください。
完全な文書を参照する場合は info coreutils 'cat invocation' を実行してください。
```

[目次に戻る]

grep

```
使用法: grep [OPTION]... PATTERN [FILE]...
各 FILE または標準入力内の PATTERN を検索します。
PATTERN はデフォルトでは基本正規表現 (BRE) です。
例: grep -i 'hello world' menu.h main.c

正規表現の選択および解釈:
  -E, --extended-regexp     PATTERN を拡張正規表現 (ERE) とする
  -F, --fixed-strings       PATTERN を改行で区切られた固定文字列の組とする
  -G, --basic-regexp        PATTERN を基本正規表現 (BRE) とする
  -P, --perl-regexp         PATTERN を Perl 正規表現とする
  -e, --regexp=PATTERN      一致処理に PATTERN を使用する
  -f, --file=FILE           FILE から PATTERN を取得する
  -i, --ignore-case         大文字と小文字を区別しない
  -w, --word-regexp         強制的に単語全体で PATTERN の一致処理を行う
  -x, --line-regexp         強制的に行全体で PATTERN の一致処理を行う
  -z, --null-data           データの行末を改行ではなく NULL とする

Miscellaneous:
  -s, --no-messages         suppress error messages
  -v, --invert-match        select non-matching lines
  -V, --version             print version information and exit
      --help                display this help and exit
      --mmap                deprecated no-op; evokes a warning

出力の制御:
  -m, --max-count=NUM       NUM 回一致後に中断する
  -b, --byte-offset         出力行と併せてバイトオフセットを表示する
  -n, --line-number         出力行と併せて行番号を表示する
      --line-buffered       行ごとに出力を flush する
  -H, --with-filename       一致するごとにファイル名を表示する
  -h, --no-filename         出力の先頭にファイル名を付けない
      --label=LABEL         標準入力のファイル名の接頭辞として LABEL を使用する
  -o, --only-matching       行の中で PATTERN に一致した部分のみ表示する
  -q, --quiet, --silent     通常出力を全て抑止する
      --binary-files=TYPE   バイナリファイルの形式を TYPE と仮定する
                            TYPE は 'binary'、'text' または 'without-match'
  -a, --text                --binary-files=text と等価
  -I                        --binary-files=without-match と等価
  -d, --directories=ACTION  ディレクトリの扱い方を指定する
                            ACTION は 'read'、'recurse' または 'skip'
  -D, --devices=ACTION      デバイス、FIFO およびソケットの扱い方を指定する
                            ACTION は `read' または `skip'
  -r, --recursive           --directories=recurse と等価
  -R, --dereference-recursive  上と同様だがシンボリックリンクを辿る
      --include=FILE_PATTERN  FILE_PATTERN に一致したファイルのみ検索する
      --exclude=FILE_PATTERN  FILE_PATTERN 一致したファイル・ディレクトリをスキップする
      --exclude-from=FILE   FILE から読み込んだファイル名のパターンに一致するファイルをスキップする
      --exclude-dir=PATTERN  PATTERN に一致したディレクトリをスキップする
  -L, --files-without-match  PATTERN に一致しない FILE の名前のみ表示する
  -l, --files-with-matches  PATTERN に一致する FILE の名前のみ表示する
  -c, --count               FILE ごとに一致した行数のみ表示する
  -T, --initial-tab         タブを使用して整列する (必要な場合)
  -Z, --null                FILE の名前を表示した後に値が 0 のバイトを出力する

前後の表示に関する制御:
  -B, --before-context=NUM  一致した前の NUM 行を表示する
  -A, --after-context=NUM   一致した後の NUM 行を表示する
  -C, --context=NUM         一致した前後 NUM 行を表示する
  -NUM                      --context=NUM と等価
      --color[=WHEN],
      --colour[=WHEN]       一致した文字列をハイライトするための印を使用する。
                            WHEN は 'always'、'never' または 'auto'
  -U, --binary              行末にある CR を削除しない (MSDOS/Windows)
  -u, --unix-byte-offsets   CR が無いものとしてオフセットを表示する
                            (MSDOS/Windows)

'egrep' は 'grep -E' を意味します。'fgrep' は 'grep -F' を意味します。
'egrep' または 'fgrep' による直接起動は廃止予定です。
When FILE is -, read standard input.  With no FILE, read . if a command-line
-r is given, - otherwise.  If fewer than two FILEs are given, assume -h.
Exit status is 0 if any line is selected, 1 otherwise;
if any error occurs and -q is not given, the exit status is 2.
```

```
バグを発見したら <bug-grep@gnu.org> に報告して下さい。
翻訳に関するバグは<translation-team-ja@lists.sourceforge.net>に報告してください。
GNU Grep のホームページ: <http://www.gnu.org/software/grep/>
GNU ソフトウェアを使用する際の一般的なヘルプ: <http://www.gnu.org/gethelp/>
```

[目次に戻る]

sed

```
使用法: sed [OPTION]... {script-only-if-no-other-script} [input-file]...

  -n, --quiet, --silent
                 パターン空間の自動出力を抑制する
  -e script, --expression=script
                 実行するコマンドとして script を追加する
  -f script-file, --file=script-file
                 実行するコマンドとして script-file の中身を追加する
  --follow-symlinks
                 処理の際にその場でシンボリックリンクを辿る
  -i[SUFFIX], --in-place[=SUFFIX]
                 edit files in place (makes backup if SUFFIX supplied)
  -l N, --line-length=N
                 `l' コマンドの行折り返しの長さを指定する
  --posix
                 全ての GNU 拡張を無効にする
  -r, --regexp-extended
                 スクリプトで拡張正規表現を使用する
  -s, --separate
                 複数のファイルを処理する際に連続した単一の長いストリーム
                 としてではなく、個別に取り扱う
  -u, --unbuffered
                 入力ファイルからデータをごく少量ずつ取り込み、頻繁に出力
                 バッファに出力 (flush) する
  -z, --null-data
                 separate lines by NUL characters
      --help     このヘルプを表示して終了する
      --version  バージョン情報を表示して終了する

-e、--expression、-f または --file オプションのいずれも与えられない場合、
最初のオプションでない引数が解釈する sed スクリプトとして扱われます。
全ての残りの引数は入力ファイル名として扱われます。入力ファイルが指定さ
れていない場合は、標準入力から読み込みます。

GNU sed ホームページ: <http://www.gnu.org/software/sed/>.
GNU ソフトウェアを使用する際の一般的なヘルプ: <http://www.gnu.org/gethelp/>.
電子メールによるバグ報告の宛先: <bug-sed@gnu.org>
報告の際、“Subject:” フィールドのどこかに “sed” を入れてください。
翻訳に関するバグは<translation-team-ja@lists.sourceforge.net>に報告してください。
```

[目次に戻る]

seq

```
使用法: seq [OPTION]... LAST
または: seq [OPTION]... FIRST LAST
または: seq [OPTION]... FIRST INCREMENT LAST
Print numbers from FIRST to LAST, in steps of INCREMENT.

Mandatory arguments to long options are mandatory for short options too.
  -f, --format=FORMAT      use printf style floating-point FORMAT
  -s, --separator=STRING   use STRING to separate numbers (default: ¥n)
  -w, --equal-width        equalize width by padding with leading zeroes
      --help     この使い方を表示して終了する
      --version  バージョン情報を表示して終了する

FIRST か INCREMENT を省略した場合、デフォルトとして 1 が設定されます。よって
INCREMENT を省略した場合は LAST が FIRST より小さい場合でもデフォルトが 1に
なります。FIRST、 INCREMENT および LAST は浮動小数の値として解釈されます。
FIRST が LAST より小さい場合、通常 INCREMENT を正にします。また、
FIRST が LAST より大きい場合、通常 INCREMENT を負にします。
FORMAT must be suitable for printing one argument of type 'double';
it defaults to %.PRECf if FIRST, INCREMENT, and LAST are all fixed point
decimal numbers with maximum precision PREC, and to %g otherwise.

seq のバグを発見した場合は bug-coreutils@gnu.org に報告してください。
GNU coreutils のホームページ: <http://www.gnu.org/software/coreutils/>
GNU ソフトウェアを使用する際の一般的なヘルプ: <http://www.gnu.org/gethelp/>
seq の翻訳に関するバグは <http://translationproject.org/team/ja.html> に連絡してください。
完全な文書を参照する場合は info coreutils 'seq invocation' を実行してください。
```

[目次に戻る]

cut

```
使用法: cut OPTION... [FILE]...
Print selected parts of lines from each FILE to standard output.

Mandatory arguments to long options are mandatory for short options too.
  -b, --bytes=LIST        バイトで数えた LIST を選択する
  -c, --characters=LIST   文字で数えた LIST を選択する
  -d, --delimiter=DELIM   フィールドの区切り文字として TAB の代わりに DELIM
                            を使用する
  -f, --fields=LIST       LIST のフィールドのみを選択する。-s オプションが
                            指定されない限り、区切り文字を含まない行も表示
                            する
  -n                      (無視される)
      --complement        選択されたバイト数、文字数またはフィールド数の
                            組を補足する
  -s, --only-delimited    区切り文字を含まない行を出力しない
      --output-delimiter=STRING  出力の区切り文字として STRING を使用
                            デフォルトでは入力の区切り文字を使用
      --help     この使い方を表示して終了する
      --version  バージョン情報を表示して終了する

-b、 -c または -f はただ一つのみ、しかも必ず使用してください。各
LIST はコンマで区切られた単一または複数の範囲で構成されます。選択
した入力は読み込まれた順番でただ一度だけ出力されます。
範囲指定は以下のいずれかです。
  N     N 番目のバイト、文字またはフィールド。行頭を1とする
  N-    N 番目のバイト、文字またはフィールドから行末まで
  N-M   N 番目から M 番目(これも含める)までのバイト、文字またはフィールド
  -M    行頭から M 番目(これも含める)までのバイト、文字またはフィールド

FILE が無いまたは - の場合は標準入力から読み込みます。

cut のバグを発見した場合は bug-coreutils@gnu.org に報告してください。
GNU coreutils のホームページ: <http://www.gnu.org/software/coreutils/>
GNU ソフトウェアを使用する際の一般的なヘルプ: <http://www.gnu.org/gethelp/>
cut の翻訳に関するバグは <http://translationproject.org/team/ja.html> に連絡してください。
完全な文書を参照する場合は info coreutils 'cut invocation' を実行してください。
```

[目次に戻る]

tr

```
使用法: tr [OPTION]... SET1 [SET2]
標準入力から読み込んだ文字を置換、切り詰め、削除し、標準出力に書き込みます。

  -c, --complement        SET1 の補集合を使用する
  -d, --delete            SET1 中の文字を削除する。置換は行わない
  -s, --squeeze-repeats   入力の中に SET1 に含まれる文字が連続して存在する
                            場合に 1 個に置換する
  -t, --truncate-set1     最初に SET1 を SET2 の長さまで切り詰める
      --help     この使い方を表示して終了する
      --version  バージョン情報を表示して終了する

SET は文字列によって指定します。多くの場合その文字自身を表現します。
解釈のされ方は以下の通りです:

  ¥NNN            文字の八進数表現(1 から 3 個の 八進数)
  ¥¥              バックスラッシュ
  ¥a              ベル
  ¥b              バックスペース
  ¥f              フォームフィード
  ¥n              改行
  ¥r              復帰
  ¥t              水平タブ
  ¥v              垂直タブ
  CHAR1-CHAR2     CHAR1 から CHAR2 までを昇順に展開した文字列
  [CHAR1-CHAR2]   SET1 と SET2 の両方で指定した場合には CHAR1-CHAR2 と同じ
  [CHAR*]         SET2 として, CHAR を SET1 の長さ分展開した文字列
  [CHAR*REPEAT]   CHAR を REPEAT 個展開した文字列, REPEAT の値を 0 から
                      始めた場合には八進数として解釈する
  [:alnum:]       全てのアルファベットと数字
  [:alpha:]       全てのアルファベット
  [:blank:]       全ての水平方向空白類文字
  [:cntrl:]       全ての制御文字
  [:digit:]       全ての数字
  [:graph:]       全ての表示可能文字。空白は含まない
  [:lower:]       全ての小文字アルファベット
  [:print:]       全ての表示可能文字。空白も含む
  [:punct:]       全ての句読点
  [:space:]       全ての水平及び垂直タブ文字
  [:upper:]       全ての大文字アルファベット
  [:xdigit:]      全ての十六進数数値
  [=CHAR=]        全ての CHAR と等価な文字

置換は -d が与えられず、 SET1 および SET2 の両方が指定されたときに実行されます。
-t は置換の時のみ使用されます。SET2 は必要に応じて SET1 の長さまで最後の文字を
繰り返すことで拡張されます。 SET2 の超過した文字は無視されます。[:lower:] およ
び [:upper:] のみ、置換における SET2 で使用すると昇順であることが保証されます。
これは大文字・小文字の変換を指定する時のみに組み合わせとして使用されます。置換
でも削除でもない場合は -s では SET1 が使われます。切り詰めの場合には SET2 が置
換、削除の後に使用されます。

tr のバグを発見した場合は bug-coreutils@gnu.org に報告してください。
GNU coreutils のホームページ: <http://www.gnu.org/software/coreutils/>
GNU ソフトウェアを使用する際の一般的なヘルプ: <http://www.gnu.org/gethelp/>
tr の翻訳に関するバグは <http://translationproject.org/team/ja.html> に連絡してください。
完全な文書を参照する場合は info coreutils 'tr invocation' を実行してください。
```

[目次に戻る]