オープンCAE勉強会＠岐阜
2015/06/13

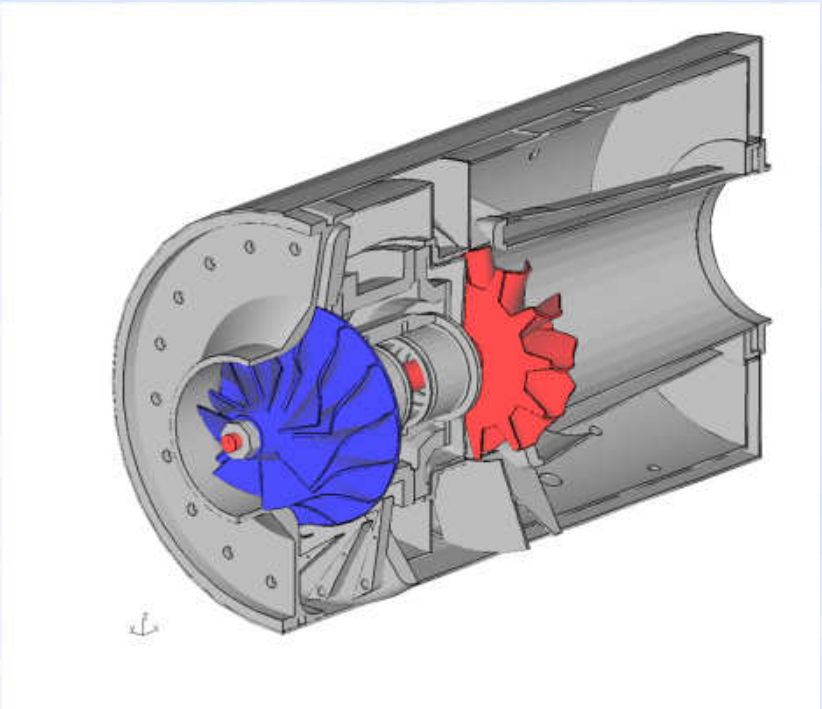
構造解析ソフトCalculix-extras版のLinux(Ubuntu)へのインストールについて

OpenCAE勉強会@岐阜
ＳＨ

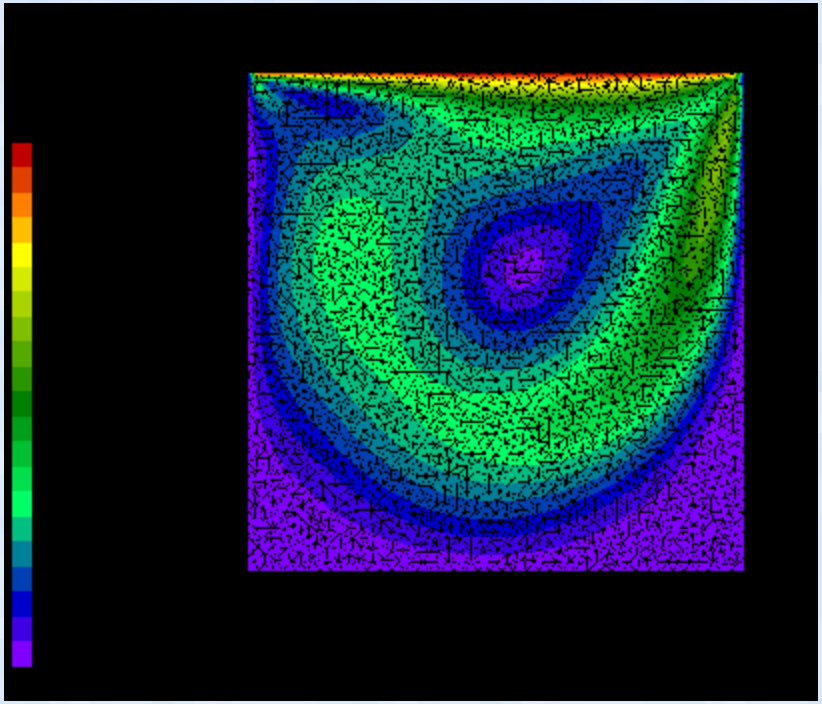
本日の発表内容

- Calculixとは?
- Calculix-Extrasとは?
- Calculixインストール(バイナリ編)
- Calculixインストール(ソースから編)
  - -Calculix通常版
  - -Calculix Extras版
- 各行列ソルバの計算速度比較
- まとめ

Calculixとは?



Calculix HP から

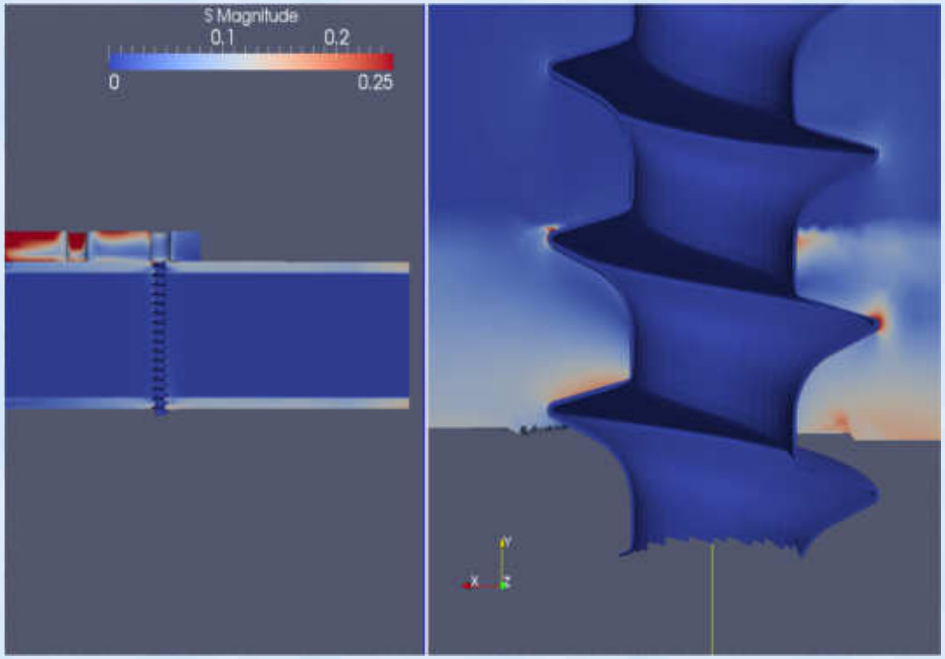


流体解析の例Cavity FLOW in Calculix

- 商用ソフトABAQUSと同様の入力書式をもつオープンソース ABAQUSを仕事で使っている人は文法を勉強しないでそのまま使える。
  知らない人もABAQUSのマニュアルを見れば大体使い方が分かる。
  (ﾃｷｽﾄ入力ﾍﾞｰｽのﾓﾃﾞﾗｰ, ﾒｯｼｬｰ, ｿﾙﾊﾞ, POSTを包含した非線形構造解析ソフト、一部流体解析も可能)
- http://www.bconverged.com/calculix にてWindows実行バイナリも公開
- Linux で利用する場合は本家のHP からソースをダウンロードしてコンパイル→ http://www.dhondt.de/
  するかCaelinux(DVD-iso)などのバイナリを利用する。 **ソースからのコンパイルは結構大変。**
- 現在の最新版は2.8p2 ソースコードが公開中

- 非線形(大変形、接触解析、材料非線形(塑性、クリープ、温度依存etc)が可能
- 課題:使っている標準行列ソルバ(Spools)→ アウトコアメモリ計算に対応していないのであまり
  大規模な計算(数10万～100万メッシュ程度限界)には対応できなかった(メモリ搭載量による)。
  (→ 他ソルバを使えばある程度規模の大きな計算ができますが、計算時間はかかる)

Calculix-Extrasとは

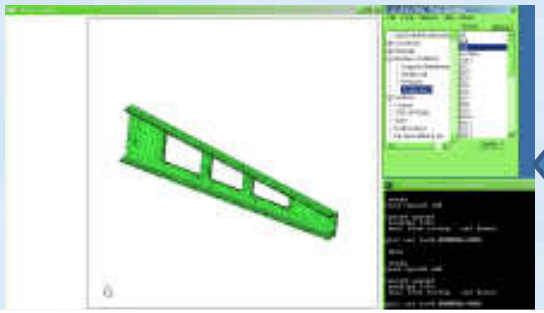


CalculiX Extras
project 解析事例

- 標準Calculixに以下の機能を追加するプロジェクト
- ①　GPGPU対応行列ソルバプログラムインターフェース
  CUDAベース行列ソルバCuda-CUSPと Cholmod
  のインターフェースプログラム
- ②　ParaViewでの結果処理インターフェース(Exodus IIフォーマットでの結果ファイル出力)
- http://ccx.openaircraft.com/ccx_extras-dl.html
  →　こちらも最新版2.8p2のみに現在は対応している
- 今回発表内容はこのインストール方法に偏っているので、面白い内容では無いです！

Calculixインストール(バイナリ編)①

- 本日はExtras版のソースからのビルドを主眼に説明するのでバイナリインストールについてはメモ書き程度ですので、ご了承ください。
- Windows版：下記の2種類のWindows環境でのバイナリが公開されている。
- ① http://www.bconverged.com/calculix
(安定しており良い。商用として有償版を$55を買うことも可能、有償版はStepなどのCADインターフェースやABAQUSコンバータなどが付属してるらしい?)
→ 有償版は買ったことが無いのでどの程度のものかわかりません！
Windowsインストラーがついているので、実行するだけでインストールできる(非常に簡単です こちらおすすめ)。
- ② http://www.calculixforwin.com/
(cygwinでビルドしたバージョンと思われる?Windowsでのビルド方法も書かれているので、自力でビルドすることも可能と思われる?)
インストールは圧縮されたzipファイルを展開するだけ。V2.6ベース。
Ansysメッシュ形式ファイルのコンバータや独自GUIなどが付属している



CalculixforwinのGUI 通常のGUIの横にメニューが表示され使い易くなっている

Calculixインストール(バイナリ編)②

- Linux版バイナリインストール
  こちらも3つくらい方法がある。
- ①　HPにアップされているLinuxバイナリを使う
  http://www.dhondt.de/　の　a Linux executable
  ダイナミックライブラリが自環境に一致しなくてほとんどそのままでは動かないと思います～のでこれは使わない方が無難。
- ② http://sourceforge.net/projects/calculix-rpm/files/のものを使う。これはGUI部分のみっぽい? こちらもインストールしたことが無いので詳細は不明

- ③ CAE Linux のバイナリを使う
  CAE Linux PJでビルドされているCalculixは一部Extrasプロジェクトに対応しているので、これを流用するとParaView出力機能は使うことができる。しかし,最新CAELINUX2013のISOイメージにはこのバージョンが入っていないので以下手順で入手
  →　詳細は次ページ

Calculixインストール(バイナリ編)③

・以下のCAELINUXのレポジトリからCalculix のバイナリ(debパッケージファイル)をダウンロードする、
http://sourceforge.net/projects/caelinux/files/dists/2013/stable/binary-amd64/
の中の calculix-2.6.1-mt-caelinux-0.1-cae2013-amd64.deb をダウンロードする。CAElinux2013ISOイメージには一つ前バージョンのcalculix2.5が入っているが、こちらのバージョンはEXTRAS版機能(ParaView出力)に対応していない.

・以下で必要なライブラリをインストールする

```
$ sudo apt-get install libhdf5-openmpi-dev
$ sudo apt-get install f2c
$ sudo apt-get install libextutils-f77-perl
$ sudo apt-get install libcnf-dev
```

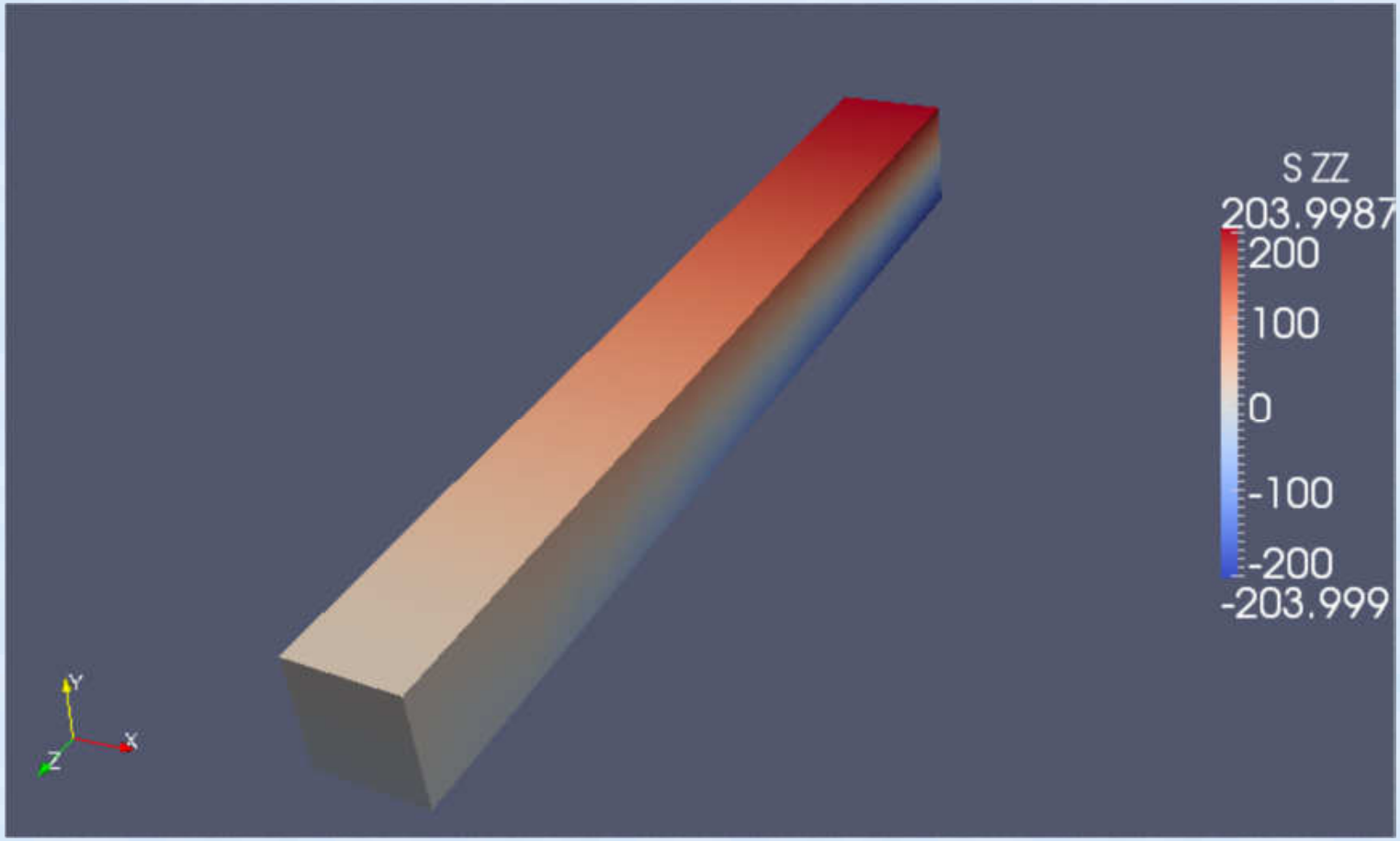
最後にダウンロードしたパッケージをインストール

```
$ sudo dpkg -i calculix-2.6.1-mt-caelinux-0.1-cae2013-amd64.deb
```

(/opt/Calculix の下に展開されます)

Calculixインストール(バイナリ編)④

$ /opt/CalculiX_2.6.1/ccx_2.6.1/bin/ccx_2.6.1 -i hari9c3d8i-ccx -o exo
にて 計算実行 (hari9c3d8iは入力ファイル) -o exo のオプションをつけると ParaViewで
結果処理できる Exodus-IIのファイルが出力される。↓計算結果こんな感じ



Calculixインストール(ソースから編) ①
- extra版では無いケース①-

この辺の設定はTSUNODAKO さんのブログ↓が詳しいので参考にしてください
http://freecaetester.blog62.fc2.com/blog-entry-237.html **LINUXにCalculiX ccxをインストールする その1 ダウンロードとSPOOLESのコンパイル**

1 : SourceCode圧縮ファイルを下記からDLする。
(最新版以外は公開されていない模様)
http://www.dhondt.de/ の下の the source code
ccx_2.8p2.src.tar.bz2 を DL これを自分の作業環境に移動
(例えば /home/dexcs の直下にて以下作業を実施する)
$ bunzip2 ccx_2.8p2.src.tar.bz2 (圧縮解凍)
$ tar -xvf ccx_2.8p2.src.tar (tar ファイル展開)
(ここから先は必要なライブラリのインストールが必要なためしばらく放置)

2 : 必要なライブラリを先にインストールします。
必要なライブラリは直接法行列ソルバのSPOOLESと固有値計算の先に使用するARPACKである。これをソースから先にビルドします。
SPOOLES: http://netlib.sandia.gov/linalg/spooles/ spooles.2.2.tgz のTGZファイルを
DL とりあえず /home/dexcs/Software/spooles に展開します。

```
$ mkdir /home/dexcs/Software/spooles
$ mv ./spooles.2.2.tgz /home/dexcs/Software/spooles/.
$ cd /home/dexcs/Software/spooles/
$ gunzip spooles.2.2.tgz
$ tar -xvf spooles.2.2.tar
```

ここでSpooles ソースにバグなどがあるので、以下２箇所を修正する
① /Tree/src/makeGlobalLib の9行目くらいにdrawTree.c のファイルを呼び出す箇所があるが、このファイルはもう無いのでdraw.c に変更する。変更しないとこの場所でコンパイルエラーで落ちる。

```
OBJ = Tree
SRC = basics.c ¥
      compress.c ¥
      draw.c ¥
      init.c ¥
```

② Make.incファイルのC コンパイラの指定を修正
CC = /usr/lang-4.0/bin/cc (大昔のSOLARIS OSの仕様)→ CC = /usr/bin/cc または単純に CC = gcc など
修正終わったら make lib を実行 make では無いので注意！！

```
$ make lib
```

Calculixインストール(ソースから編) ②
- extra版では無いケース②-

- Spooles インストールの続き

  make lib が上手くいくと直下に spooles.a ができるので,これをCalculix make 時に指定する。
- 並列実行を行うためにはマルチスレッド版のビルドも必要なため、こちらも同時におこなう
  spooles のインストールディレクトリ下で
  $ cd MT
  $ make lib
  以上でMT/src 下に spoolesMT.a が出来るのでこれをCalculix make 時利用する
- なお Spooles のビルドは必須では無く、apt-get にてライブラリをインストールすることが可能である。
  (ただし、自分でMakeしたものをリンクしないと私の場合はスレッド並列計算がうまくできなかったので注意！)
  spooles をバイナリライブラリから利用する場合は下記にて
  $ sudo apt-get install libspooles-dev
  で/usr/libにspooles のライブラリをインストールして Calculix make 時に –lspooles のフラグを指定する
  (Makefile に LDFLAGS += -lspooles -lpthread などの行を追加する)

- 次に固有値計算時に使用するARPACKをビルドします。

Calculixインストール(ソースから編) ③
- extras版では無いケース③-

・ARPACKインストール：下記からARPACKのパッケージをダウンロードする。
http://www.caam.rice.edu/software/ARPACK/
・arpack96.tar.gz とpatch.tar.gz をDLする。
・とりあえず　/home/dexcs/Software/ARPACK
に展開します。

```
$ mv arpack96.tar.gz /home/dexcs/Software
  /ARPACK/.
$ mv patch.tar.gz /home/dexcs/Software
  /ARPACK/.
$ cd /home/dexcs/Software/ARPACK
$ gzip -d arpack96.tar.gz
$ tar -xvf arpack96.tar
```

この辺の設定はTSUNODAKO さんのブログ↓が詳しいので参考にしてください
http://freecaetester.blog62.fc2.com/blog-entry-238.html LINUXにCalculiX ccxをインストールする　その2　ARPACKとccxのコンパイル

・パッチファイルの中身をarpackの展開先にそのまま上書き
・ARmake.inc の中身を編集
Fortranコンパイラが デフォルトで　”f77” とか昭和時代のコンパイラが指定されているので
FC　　= gfortran　に変更
その他下記修正
PLAT = SUN4 → PLAT = INTEL (または PLAT = linux）
FFLAGS = -O -cg89 →　FFLAGS = -O2（または行を削除でも可)
・ARPACK/UTIL/second.f の中身を編集 24行目の下記の行をコメントアウト(または行削除)
*** EXTERNAL　　ETIME　　(消しておかないとCalculix make 時にエラーとなる）
・ARPACK のMAKE→ $ make lib　これで　libarpack_INTEL.a　などができる。

Calculixインストール(ソースから編) ④
- extras版では無いケース④-

- **Calculix のソース展開先に戻って本体のmakeを行う**
- **マルチスレッド並列版のMakefileのテンプレート(Makefile_MT)があるのでこれをベースに自分の環境用に編集する。基本的には先ほどのSpoolesとARPACKの置き場所だけちゃんと書いてあれば問題無いはず?**
  $ cd /home/dexcs/Calculix/ccx2.8p2/src
  $ cp Makefile_MT Makefile
- Makefile **の編集　大体下記のように編集する。終わったらMakeして完了！**

```
CFLAGS = -Wall -O3  -I /home/dexcs/Software/spooles -DARCH="Linux" -DSPOOLES -DARPACK -DMATRIXSTORAGE -DUSE_MT=1
FFLAGS = -Wall -O3
CC=cc
FC=gfortran
.c.o :
	$(CC) $(CFLAGS) -c $<
.f.o :
	$(FC) $(FFLAGS) -c $<
include Makefile.inc
SCCXMAIN = ccx_2.8p2.c
OCCXF = $(SCCXF:.f=.o)
OCCXC = $(SCCXC:.c=.o)
OCCXMAIN = $(SCCXMAIN:.c=.o)
DIR=/home/dexcs/Software/spooles
LIBS = ¥
	$(DIR)/MT/src/spoolesMT.a ¥
	$(DIR)/spooles.a ¥
	/home/dexcs/Software/ARPACK/libarpack_INTEL.a ¥
	-lpthread -lm
ccx_2.8p2_MT: $(OCCXMAIN) ccx_2.8p2_MT.a  $(LIBS)
	./date.pl; $(CC) $(CFLAGS) -c ccx_2.8p2.c; $(FC) -Wall -O3 -o $@ $(OCCXMAIN) ccx_2.8p2_MT.a $(LIBS)
ccx_2.8p2_MT.a: $(OCCXF) $(OCCXC)
	ar vr $@ $?
```

赤字部分は書き直し箇所

Calculixインストール(ソースから編) ⑤
- extras版①-

- Extras版のビルドにはPatchファイルと他に事前に幾つかのプログラムのインストールが必要である。Extras用のPatch ファイルは下記から入手→ http://ccx.openaircraft.com/ccx_extras-dl.html

- その前にGPGPU対応用のソルバをビルドするためにはCUDAの開発ツールキット(*.cu をコンパイルするための開発環境)が必要 このあたりからダウンロードしてインストールしてください。
https://developer.nvidia.com/cuda-downloads
(最近OpenFOAM もGPGPU対応のSimflow が出たので、GPGPUマシンでも買う？)

- とりあえず、私はLinux x86 Ubunts12.04 版をインストールした(14.04の人は14.04用をインストールする)
cuda_7.0.28_linux.run をダウンロードして下記にて実行
$ sudo sh cuda_7.0.28_linux.run
- めちゃくちゃ長い契約条項が出てくるが全て~~無視して~~(了解して)先に進む
わたくしのマシンはGPGPU搭載していないのでGPGPUドライバはインストールせずに、開発環境だけインストールする。

  Install NVIDIA Accelerated Graphics Driver for Linux-x86_64 346.46? ((y)es/(n)o/(q)uit): n
  Do you want to install the OpenGL libraries? ((y)es/(n)o/(q)uit) [default is yes]: n
  Install the CUDA 7.0 Toolkit? ((y)es/(n)o/(q)uit): y ← ここは必ずYES
  Install the CUDA 7.0 Samples? ((y)es/(n)o/(q)uit): y

- 開発環境は /usr/local/cuda の下にインストールされるので、ここにPATH とライブラリのPATH (LD_LIBRARY_PATH) の環境変数を設定しておく(自分の.bashrcに設定しておく)

  PATH=$PATH:/usr/local/cuda/bin
  LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/cuda/lib64:/usr/local/cuda/lib

- 次にGPGPU 対応の線形ソルバプログラムをインストールする。

Calculixインストール(ソースから編) ⑥
- extras版② -

- 次にGPGPU 対応の線形ソルバプログラムをインストールする。対応している線形ソルバはCUAD CUSPとSuiteSparse の中のCHOLMOD
- CUSPは反復法ソルバでCHOLMODは直接法ソルバのようである。最初にSuiteSparseをビルドする
- SuiteSparseはhttp://faculty.cse.tamu.edu/davis/suitesparse.html からDLする(Calculix extras からのリンクは切れているので注意!) SuiteSparse-4.4.4.tar.gz をDLして展開する。展開場所はとりあえずどこでも良い.
  $ tar -xvzf SuiteSparse-4.4.4.tar.gz
- 環境にあわせてconfigファイルを編集する
  SuiteSparse/SuiteSparse_configの下にあるSuiteSparse_config.mkを自分の環境にあわせて置き換える
  -linux でCPU(GPU無し)しかない人→
  SuiteSparse_config_linux.mk を SuiteSparse_config.mk にコピー
  -linux でGPGPUの使える人
  SuiteSparse_config_GPU_gcc.mk を SuiteSparse_config.mk にコピー
  (その他 MAC版とかLinux INTEL コンパイラ版とかが標準である)

  とりあえずmetis はイラネーと思い、SuiteSparse_config.mk の下記アンコメントしました。
  Metis リンクする人はMetisパスを設定する必要ありと思われます
  (GPU の人はリンクしないとダメかな?)
  # uncomment this line to compile CHOLMOD without METIS:
  CHOLMOD_CONFIG = -DNPARTITION

- 終わったらSuiteSparse直下に戻って、make コマンド実行する
- Make 無事完了したらスーパーユーザで make install を実行
  $ sudo make install
  無事全て完了すると /usr/local/lib の下に libcholmod.a が出来る
  calculix extras 版を make する時必要になります。

Calculixインストール(ソースから編) ⑦
- extra版③-

- 次にGPGPU 対応の線形ソルバプログラムCUSPをインストールする。
- こちらは特にコンパイルする必要もなくファイルを展開するだけで良いらしい?
- しかし,Calculix Extrasのリンクに張られている下記↓のCUSP0.3.1のライブラリを使うとCalculix のMake作業中にコンパイルエラーが出てCalculixのビルドが失敗する。
- https://code.google.com/p/cusp-library/downloads/list
でCALCULIX のCUDACUSP.CUソースを中身を調べてみたら、呼び出すファイル構成やその他、色々変わっており、どうやらバージョンが0.3.1に対応していないようである。
- 結論から言うとCalculix 2.8P2のCUSPの対応バージョンはV0.4.0でこれはGITHUBに公開されているので、こちらを使う必要がある。下記からDLしてくる。こちらも最新公開はV0.5.1 で、こちらでもヤッパリエラーになるのでGITの下の方にある旧版のV0.4.0を選択してZIP形式でDLする。
- https://github.com/cusplibrary/cusplibrary

- DLしたらスーパーユーザで /usr/local/cuda/include/cusp の下に全部zipファイルを展開する。これでCUSPのインストールは完了だが
(V0.4.0に対応してるが分かるまで3日ほどかかったわけだが, こんなライブラリバージョン依存のくそプログラムを書くのは迷惑なので本当にやめていただきたい！！しかもリンク先のバージョンでは動かず、対応しているバージョンを探す必要があり、ロールプレイングゲームのようなむずかしさである。)

Calculixインストール(ソースから編) ⑧
- extras版④-

- これでGPGPU線形ソルバ側の準備は完了
- 次にEXODUS-Ⅱ形式ファイル出力(ParaView読み込みのため)するためのライブラリをインストールする。こちらは特に難しいことは無いので、EXTRAS版のHP↓に書かれている手順に従って、apt-get コマンドでライブラリだけインストールする(Ubuntu ユーザ向けインストラクション)。
- http://ccx.openaircraft.com/ccx-ubuntu-instructions.html

$sudo apt-get install libexodusii-dev

(その他にこのライブラリをインストールするのに何か別ライブラリが必要！と怒られたら個別にそのライブラリをinstall お願いします。NetcdfとHDF5のライブラリが必要と言われるかもしれません？いずれにしろapt-get でインストールできます)

Calculixインストール(ソースから編) ⑨
- extras版⑤-

- これで全ての準備は完了。あとはEXTRASのHPから修正PATCHをDLしてCALCULIX2.8のソースにあて、その後 Makefile を編集して Make を実行すれば良い。
- http://ccx.openaircraft.com/ccx_extras-dl.html から Patch ファイルを3個入手する。
  Download the latest patches.
  2.8p2.0.Makefile.patch
  2.8p2.0.exo.patch
  2.8p2.0.solver.patch
- Calculix 2.8P2 のTARファイル(ccx_2.8p2.src.tar.gz)を本家サイト
  http://www.dhondt.de/ から入手し展開する。
  $ tar -xvzf ccx_2.8p2.src.tar.gz
- Patch ファイルを 展開後の ソースにあてる(ここはHPにかかれた通り)。
  $ cd CalculiX/ccx_2.8p2
  $ patch -p2 < ../../2.8p2.0.Makefile.patch
  $ patch -p2 < ../../2.8p2.0.exo.patch
  $ patch -p2 < ../../2.8p2.0.solver.patch

- Makefile が大幅に書き直しされるので、自分の環境にあわせて編集する。CUDA関連のライブラリと事前にビルドしたライブラリの場所を指定するだけで良い。とりあえず適当に編集してMAKE中におこられたら、設定しなおす。関係ありそうなところだけ次ページに張っておきます。
  $ cd CalculiX/ccx_2.8p2/src
  $ vi Makefile

Calculixインストール(ソースから編) ⑩
- extras版⑥-

- Makefile

```
CC=nvcc #またはgcc
FC=gfortran
## Multi Threaded and MPI
CFLAGS += -DUSE_MT
#CFLAGS += -DCALCULIX_MPI -fopenmp
## SPOOLES
CFLAGS  += -I/home/dexcs/Software/spooles -I/home/dexcs/Software/spooles/MT -DSPOOLES
LDFLAGS += -lpthread
## ARPACK
CFLAGS  += -DARPACK
LDFLAGS += -larpack
## TAUCS                TAUCS をビルドしない場合下記は全部コメントアウトします
CFLAGS  += -DTAUCS
#LDFLAGS += -ltaucs -lmetis
CFLAGS  += -I/home/dexcs/Software/taucs/src
CFLAGS  += -I/home/dexcs/Software/taucs/build/linux
## LAPACK
LDFLAGS += -llapack -lblas
## Flags for the gpu compiler
NVCCCFLAGS = -arch=sm_35
NVCCLDFLAGS = -lstdc++
##
NVCC=nvcc
# -O3 $(LONGLONG) $(NVCCCFLAGS) -Xcompiler -fopenmp
#-O3 $(LONGLONG) $(NVCCCFLAGS) # -Xcompiler -fopenmp
## CUDACUSP
## This is unique because it a template library rather than binary library
CFLAGS  += -I/usr/local/cuda/include -L/usr/local/cuda/lib64
CFLAGS  += -I/usr/local/include -L/usr/local/lib -DCUDACUSP
#-DCUDACUSP
LDFLAGS += -L/usr/local/cuda/lib64 -L/usr/local/cuda/lib -lcudart
## CHOLDMOD
## This is unique because it can be CPU or GPU based, depending on how
## SuiteSparse was compiled.  Here it is assumed that SuiteSparse also
## uses CUDA
CFLAGS  += -DSUITESPARSE
```

TAUCS というのはCALCULIX のオプションで使える(CPU向けの)線形ソルバ(直接法) です。SPOOLES がデフォルトの線形ソルバになっている

- ちなみに 2.8P2 では TAUCS ビルドはうまくいくが計算がエラーで異常終了する

Calculixインストール(ソースから編) ⑪
- extras版⑦-

- Makefileの続き

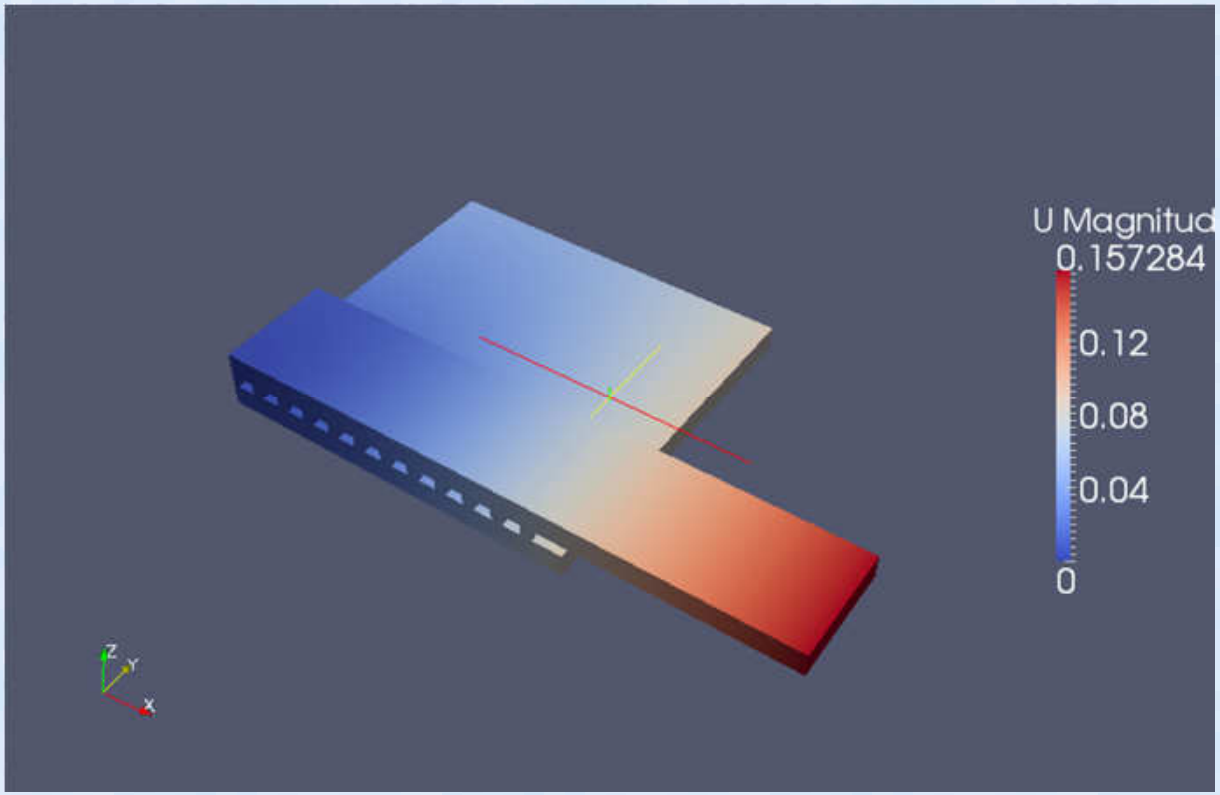
```
## EXODUSII
CFLAGS  += -DEXODUSII
LDFLAGS += -lexoIIv2c -lnetcdf
#CFLAGS  += `pkg-config --cflags exodusii` -DEXODUSII
#LDFLAGS += `pkg-config --libs exodusii`
DIR=/home/dexcs/Software/spooles
LIBS = ¥
      $(DIR)/MT/src/spoolesMT.a ¥
      $(DIR)/spooles.a ¥
      /home/dexcs/Software/ARPACK/libarpack_INTEL.a ¥
      /usr/local/lib/libcholmod.a ¥
      /usr/local/lib/libsuitesparseconfig.a ¥
      /usr/local/lib/libcolamd.a ¥
      /usr/local/lib/libamd.a ¥
      /home/dexcs/Software/taucs/lib/linux/libtaucs.a ¥
      /home/dexcs/Software/metis-4.0.3/libmetis.a ¥
      -lpthread –lm
## .cu file so not have a default implicit rule.  Define all implicit rules used.
.SUFFIXES: .o .c .cu
.c.o :
      $(CC) $(CFLAGS) -c $<
.f.o :
      $(FC) $(FFLAGS) -c $<
.cu.o:
#      $(NVCC) -c $<
      $(NVCC) -DCUDACUSP -c $<
include Makefile.inc
SCCXMAIN = ccx_2.8p2.c
## Define all the object file rules to identify dependencies
OCCXCU = $(SCCXCU:.cu=.o)
OCCXF = $(SCCXF:.f=.o)
OCCXC = $(SCCXC:.c=.o)
OCCXMAIN = $(SCCXMAIN:.c=.o)
## Link to math and standard c
CFLAGS += -lm -lc
ccx_2.8p2: $(OCCXMAIN) ccx_2.8p2.a
      ./date.pl;       $(CC) $(CFLAGS) -c ccx_2.8p2.c $(LDFLAGS); $(FC) -Wall $(FFLAGS) -o $@ $(OCCXMAIN) ccx_2.8p2.a $(LDFLAGS) $(NVCCLDFLAGS) $(LIBS)   以下略
```

ここはTAUCSをビルドしない場合不要です。

編集が終わったらsrc 直下でMakeコマンドを 実行
$ make
(なかなか一回では上手くいかないと思いますが。。)
これでエラーなければ完了

各行列ソルバの計算速度比較①

- 出来上がったCalculix extras バージョンの各種ソルバで計算時間を比較してみる。問題は熱応力計算の下記のバイメタル状サンプルの熱応力反りの計算問題。非線形の弾塑性温度依存材料物性を使っている非線形解析である(材料非線形のみで接触条件はない)



nodes: 19016
elements: 9933

あまりモデルが大きくないので計算のテストにふさわしくないかも？

各行列ソルバの計算速度比較②

- CUSPで計算してみる。CUSPを呼ぶにはCALCULIX 入力ファイルの中　*STATIC のオプションでSOLVER を指定する
- *static, solver=cudacusp　を指定（他も同様 SOLVER=CHOLMOD, SOLVER= ITARATIVE ＊＊＊）

actual total time=1.000000e-03

iteration 1

Using up to 1 cpu(s) for the stress calculation.

Using CUDA based on CUSP CG SOLVER
CUDA　v7.0
Thrust v1.8
Cusp　v0.4
bad_alloc during transfer of A to GPU

↑ CUSP の接続は上手くいっており、GPUに変換する時エラーですよ～とエラーが出て終了。
これは私のマシンにGPGPUを搭載していないためと思われます。ということで続きは柴田先生にお願いします。
もう一つの直接法CHOLMOD の方はCPUでも計算できるのでこちらだけCPUで試してみた。
（なおCPU TAUCS SOLVERも同時にビルドしたのだが、こちらのソルバも計算中にエラーで止まってしまう?
Calculix V2.8 のTAUCS 呼び出しソースコードに問題ある模様?）

各行列ソルバの計算速度比較③

- 今回は下記のソルバだけの比較ができた. ちなみに計算は全て1CPU(並列数1)で比較してます(LINUX TIME コマンドで取得)。

ソルバ	計算手法	計算時間	備考
SPOOLES	直接法	2m56.040s	CALCULIX標準
CG法対角スケール	反復法	33m30.529s	CALCULIX標準
CG法不完全LU	反復法	11m50.781s	CALCULIX標準
CHOLMOD	直接法	10m46.290s	EXTRAS版 GPU対応
CUDACUSP	反復法	×(GPUのみ)	EXTRAS版 GPU対応
TAUCS	直接法？	×(エラー終了)	CALCULIX標準 (OPTION)

残念ながら、CHOLMODは標準直接法ソルバSPOOLES より3倍くらい遅く、小規模の計算で、CPU計算では使うメリットなさそう？

まとめ

- CALCULIX 2.8P2 のEXTRAS 版をソースからビルドする方法を調査した。
- EXTRAS版のGPGPUソルバ、ParaView出力機能などが使えることを確認した。
- GPGPU搭載装置がないのでGPU 計算速度比較はできなかった。

- CPU向けではIntelMKL のParadiso ソルバが一番早いに決まっているので本当はこれを使いたいのだが？(Intel コンパイラは高いので使えません。)
- 今回はCPU並列数を1で行ったのでスレッド並列数を変化させた場合の比較もそのうち実施予定

- 藤井さんのEasyISTRを使うと Salome Universal ファイルからFrontISTRファイルと同時にCalculix向けのinp ファイル(ABAQUS仕様)も同時に出力されるので、同じメッシュファイルを用いてFrontISTR, CodeAster との比較が可能である。これらの計算速度比較もそのうち実施予定?