熱磁気モータの解析マニュアル兼引継ぎ書

田中　壮汰

1.COMSOLのversion更新方法

COMSOLのversionの更新方法を説明する．なお，centOSにおける更新方法を説明するためWindows版とは違うことに留意すること．基本的にLinuxはコマンド操作を多用する．ターミナルの使い方がわからないときはネット検索すればわかる．そのため，コマンドの説明までは本引継ぎではしない．

COMSOLの更新をするためにはsetup画面を開く必要がある．Setupのディレクトリーの場所は$ /usr/local/comsol60/Multiphysics/setupである．このsetupを開くのであるがターミナルで以下のようにコマンドをうつ．

グラフィカル ユーザー インターフェイス, テキスト

自動的に生成された説明

ここで，留意することはsudoで管理者権限にて実施することである．Userで実行すると後にerrorがでる．実行すると以下のインストール画面が確認できる．

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

後はGUI操作で進んでいくだけで簡単なので説明は省略する．

なお，マイナーアップデートの場合は$ /usr/local/comsol60/Multiphysics/updateである．

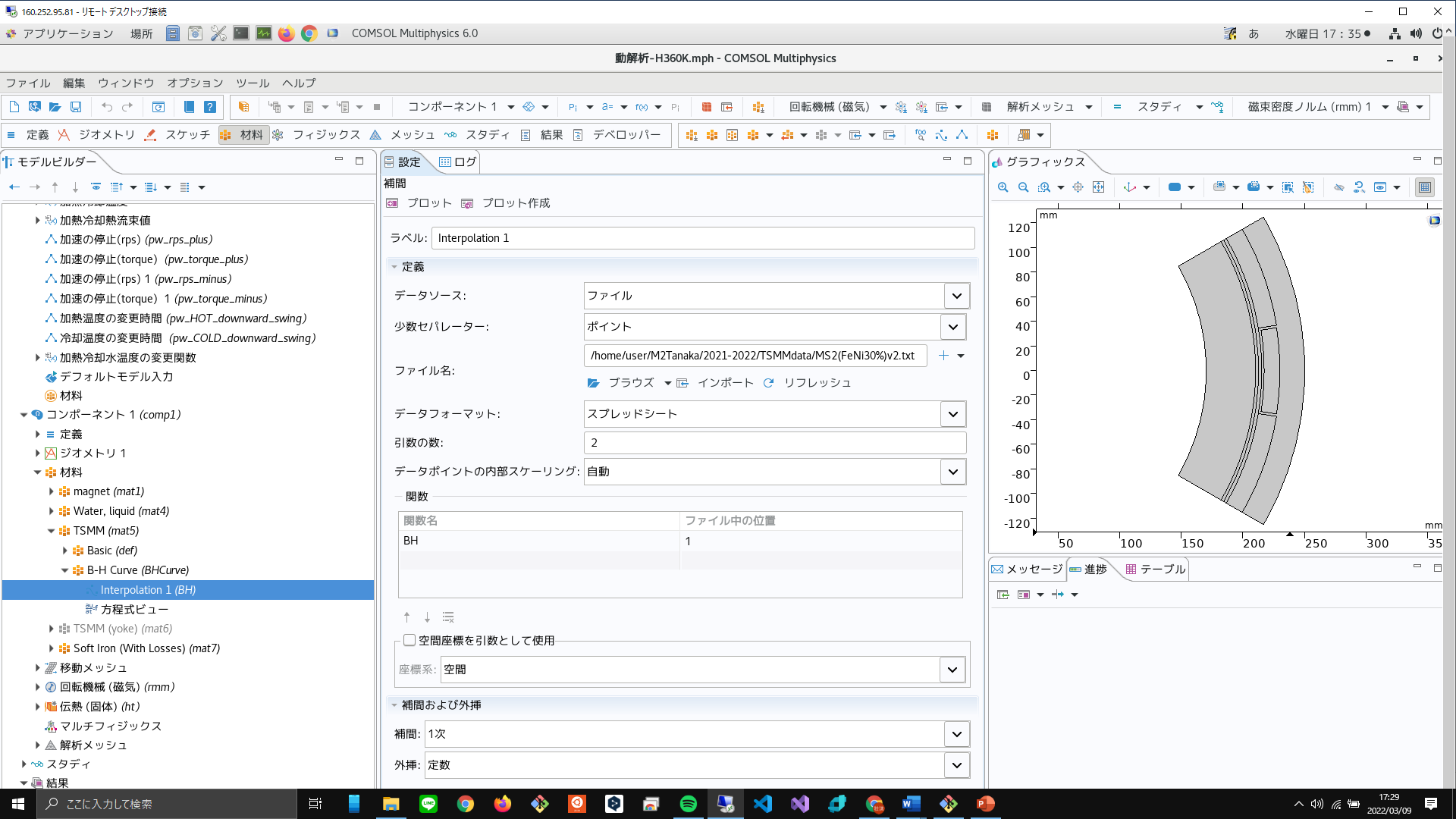
２.熱磁気モータの解析方法の説明

基本的なCOMSOLの操作方法はKESCOのHPなどに詳細に記載されているので省略する．ここでは，温度場と磁場の連成解析方法の設定について記述する．感温磁性体は温度によって磁気特性が変化する性質がある．この性質を表現するためには以下のようなテキストデータを作成する必要がある，拡張子はtxtでもdatでもよい．このテキストでは各温度でのBHデータが書かれている．もちろん，このデータは材料によって変わるので自作すること，本来は材料試験をするのが望ましい．この，材料データの精度によって，非線形性が大きくことなり計算時間および計算精度に影響する．計算が破綻する原因にもなる．自分で試行錯誤し，経験を積むとよい．

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, Word

自動的に生成された説明

次に材料データの設定方法について説明する．以下の設定項目でテキストデータをインポートし引数を2とする．



作成したテキストファイルをインポート

引数は2とする．

ここまで，設定ができれば次に以下の赤枠内を設定する．ここまでが材料データの設定である．黄色文字は警告を表しているが，無視してよい．そもそも，本来COMSOLが想定していない使い方の可能性が高い．しかし，KESCOに問合せをして確認しているので問題ない．

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

連成させるためには磁場解析において，材料データを参照させる必要がある．そのため，磁場の設定項目において以下の赤枠を設定する．ここで，温度(ht)にすると連成解析ができる．検証の時などはユーザー定義に任意の温度を設定するとよい．

グラフィカル ユーザー インターフェイス, アプリケーション

自動的に生成された説明

他の設定については私の解析ファイルやアプリケーション例などを見て勉強してほしい．

解析において，影響度が高いのがメッシュの作成である．メッシュの出来次第で解析結果は大きく変わる．そのため，メッシュ作成に関して特に集中して取り組むとよいと思われる．ほかの設定は人による差はほとんど発生しない．

３.Javaを用いてCOMSOLを設定する方法

COMSOLのインターフェイスに関してはおそらくJavaにて書かれている，マトリックスなどの計算はC++と思われる．そのため，Javaを用いると多様なことができる．例えば，面倒な境界条件の設定をfor文により設定ができ，GAの適用も可能である．そもそも，COMSOLのファイルのデフォルト拡張子は.mphであるが，.javaに変更して保存するとソースコードが見れるよるになる．

コンパイルするには以下のようにする．

$ comsol compile -jdkroot /opt/-jdk-15.0.2\_linux-x64\_bin filename.java

コンパイルして生成されたclassファイルは以下のコマンドで実行する．

$ comsol batch -np 10 -inputfile filename.class

ここで，-npの後の数字は並列コア数であり，任意に指定できる．もちろん，最大コア数までである．ライセンスの問題により，コマンドで実施する場合他のCOMSOLファイルが起動しているとライセンスエラーとなるので他のファイルは閉じておくことが必要である．

最後に，残念ながらCOMSOLに関するノウハウはインターネット上にほとんど落ちていない．日本語版のノウハウなど皆無に等しい．逆に言えば，COMSOLを知れば知るほど差別化ができるのである．

また，COMSOLは非常に多様なことができる有限要素法のソフトウェアである．PDEモードを用いれば任意の偏微分方程式も解くことができる．そのため，多くのことに挑戦してみてほしい．

4.CentOS環境におけるC++を用いた並列化

現状の中村研究室のワークステーションのほとんどのOSはwindowsであるため，この引継ぎはあまり意味をなさないかもしれないが，一応自己満足のため記載する.Windowsで計算する場合は皆visual studioを使っているはずである.しかし，Linux版はないためそもそも別のコンパイラーを使わざるを得ない.CentOSには現状gccとclangコンパイラーをインストールしてある.著者は主にgccで遊んでいた.さて，現状の設定では並列コードをコンパイルする前にターミナルで以下を入力する必要がある.

export PATH=$PATH:/usr/lib64/openmpi/bin/

上記はOpenMPIのインクルードパスを通している.そもそも，このライブラリーをincludeファイルに移せばこの作業はいらないはずであるがうまくいかなかった.著者の手元のラップトップのWSL(windows for linux)を用いたUbuntu環境ではうまく設定できた.

次にコンパイルするにはシングルの場合，C++であれば以下である.

$ g++ ~.cppである.

コンパイルするとLinuxの場合は.outの実行ファイルができるのであとは./a.outで実行すればよい．

並列コードの場合は以下である．

$ mpic++ ~.cpp

そうすると先ほど同じくa.outが作成される.

このa.outを先ほどと同じように実行するだけではシングルで実行される．並列で実行させるには以下のようにする．

$ mpirun -np 並列数 ./a.out

このようにすれば並列で実行されるはずだ，タスクマネージャーで確認するとa.outが並列数実行されているのが分かるはずだ．なお，遺伝的アルゴリズムを並列化するのであれば個体数/並列数のあまりが0であるように並列数を決めると効率がよいはずだ．

現状では一つのノードでのMPI環境しか構築していないのでこれからはぜひ，複数ノードの並列環境を作ってみると計算も速くなるし楽しいと思う．OBOG会でできたことを教えてくれると嬉しい.