FrontISTR HECMW_Partitioner マイナー修正のご報告

東京大学大学院 奥田研究室
森田 直樹

目次

- 問題背景
- 修正点
- 動作比較
- 展望

問題背景

- 膨大な数の!SECTION, !EGROUP, !ORIENTATIONを定義した入力ファイルの読込に時間がかかる.
- 名前解決を行うサブルーチンが$O(n^2)$のアルゴリズムであったことが原因.

[例] foo.msh（直交異方性の解析メッシュ等）

```
…
!SECTION, TYPE=SOLID, EGRP=EGRP001, MATERIAL=M001
!EGROUP, EGRP=EGRP001
    10001
!SECTION, TYPE=SOLID, EGRP=EGRP002, MATERIAL=M001
!EGROUP, EGRP=EGRP002
    10002
!SECTION, TYPE=SOLID, EGRP=EGRP003, MATERIAL=M001
!EGROUP, EGRP=EGRP003
    10003
…
```

修正点

- 修正を行ったソースは以下の通り
  - fstr_ctrl_common.f90　　fstr_ctrl_get_SECTION()
  - hecmw_io_mesh.c　　HECMW_io_get_egrp()
  - hecmw_io_mesh.c　　HECMW_io_add_egrp()
  - hecmw_dist.c　　HECMW_dist_get_egrp_id()
- ただし, **一時的な修正**である.
  - !SECTION, !EGROUP, !ORIENTATIONの入力順が全て一致していることが条件
- fstr_ctrl_get_SECTION(), HECMW_io_get_egrp(), HECMW_dist_get_egrp_id()は，データキャッシュを用いて高速化した.
  - データキャッシュは，前回読み込んだ位置を記憶しておく
  - 例： EGRP008の次は，まずEGRP009を読み込み比較する
- HECMW_io_add_egrp() は，ハッシュテーブルを用いて高速化した.

動作比較

- 節点数100,000, 要素数80,000, !SECTION定義数80,000
- 直交異方性を考慮した円筒モデル. OpenMPI 8並列.
- 使用計算機　東京大学 奥田研究室 TCクラスタ

- 全体計算時間 (内訳:Partitioner時間+FrontISTR時間)
- 修正前:680秒　(455秒+125秒)
- 修正後:　14秒　(　　4秒+　10秒)

- 48.5倍の高速化

動作比較

- 節点数950,000, 要素数860,000, !SECTION定義数860,000
- 直交異方性を考慮した円筒モデル. OpenMPI 8並列.
- 使用計算機　東京大学 奥田研究室 TCクラスタ

- 全体計算時間 (内訳:Partitioner時間+FrontISTR時間)
- 修正前：12時間 (30407秒(8時間半)+12251秒(3時間半))
- 修正後：371秒　(　　88秒+283秒)

- 114.9倍の高速化

展望

- 今回の修正で動作が高速化するのは，
  整った入力データの場合のみであるが，
  ソース修正によって動作高速化を行った．

- 将来的に，ハッシュテーブル等の最適なデータ構造を用いて，
  更なる堅固な高速化を行う予定である．

付録:修正前のアルゴリズム

- 名前解決を行う$O(n^2)$のアルゴリズム

```
for ( p=_sect; p; p=p->next ) {
   name = p->name;
   for ( s=_egrp; s; s=s->next ) {
      if ( strcmp ( s->name, name ) == 0 )  return s;
   }
}
```

_sect	!sectionの線形リストの先頭アドレス
_egrp	!egroupの線形リストの先頭アドレス
name	!sectionに定義された要素グループ名
strcmp	cの文字列比較組み込み関数