接触問題向け反復法線形ソルバーの解説と適用事例

2015年9月28日
第21回FrontISTR研究会

合同会社PExProCS・代表
後藤 和哉

内容

- 構造解析における線形ソルバー
- FrontISTRによる接触解析の概要
- 接触解析向け反復法線形ソルバー
- 適用事例
- 今後の開発計画

構造解析における線形ソルバー

線形ソルバーに求められるもの

- 大規模解析では線形ソルバーが計算時間の大部分を占めるため，高速化が重要
- 線形ソルバに求められる性能（理想）
  - 計算規模に対してスケーラブルであること
    - “Numerically scalable”
    - マルチグリッド系の手法が実現（AMG, FETI-DP, BDDC, etc.）
  - 並列計算への適合性が高いこと
  - ロバストであること

なぜ反復法線形ソルバーか

直接法

◎ ロバストである

△ 効率的な並列化が困難

× 問題規模の増大とともに計算量・メモリ使用量が爆発

反復法

△ ロバスト性に課題がある

○ 並列計算への適合性が高い

○ 問題規模の増大に対して、計算量・メモリ使用量の増大が緩やか

大規模問題では反復法線形ソルバーが必須

接触解析における
反復法線形ソルバーの難しさ

- 接触に伴う制約条件を考慮しなければならない
  - FrontISTRではラグランジュ乗数法を採用
- 解くべき行列そのものが反復法で解きにくいものとなる
  - ラグランジュ乗数法では、対角に0を含む非正定値行列について解くことになる
- 何らかの工夫が必要

参考： 構造解析における行列

行列の性質を決める要因

- 静解析 vs. 動解析
  - 静解析：剛性行列を解く
  - 動解析：剛性行列に質量行列が足し込まれる
    - dtが小さいほど質量行列の寄与が大きく，行列は解き易い
- 線形 vs. 非線形
  - 線形解析：同じ行列に対して求解
  - 非線形解析：解く行列が変化していく
- 厚肉 vs. 薄肉
  - 球や立方体に近い形状：固有値分布が小さめ
  - 板や梁のような形状：固有値分布がばらつく

問題によって最適な線形ソルバーが異なる

- 問題に応じて適切な解法や前処理を選択することが重要

FRONTISTRによる接触解析の概要

FrontISTRの
接触解析アルゴリズム

拡張ラグランジュ法 (Augmented Lagrange Method)

- FrontISTRに最初に実装された接触アルゴリズム
- 線形ソルバーには直接法も反復法も利用可能
- 並列は未実装

ラグランジュ乗数法 (Lagrange Multiplier Method)

- 現在よく使われているアルゴリズム
- 線形ソルバーは直接法
- 2タイプの並列化

← **Ver.4.4で反復法を有効化（試験的）**

FrontISTRにおける接触解析の処理の流れ

- **荷重・変位増分ループ**
  - 外力計算
  - **接触ループ**
    - **Newton-Raphson法ループ**
      - 接線剛性行列の計算
      - **線形ソルバーによる求解**
      - 内力計算、収束判定
    - 接触状態の確認、収束判定

FrontISTRによる接触解析の手順

1. **メッシュファイルにて接触面ペアを定義**
   – マスター面は面グループ、スレーブ面は節点グループ
2. **解析制御ファイルにて接触アルゴリズム等を設定**
   – 線形ソルバーの設定もここで行う
3. **並列計算の場合**
   方法1：領域分割を行う方法 （12ページ参照）
   方法2：“paracon” （13ページ参照）
4. **FrontISTRを実行**

並列計算1：領域分割を行う場合

手順

1. パーティショナ制御ファイルで追加オプション CONTACT=AGGREGATE を指定
   - 個々の接触面ペアが単一領域に集められらる
2. パーティショナにより領域分割

特徴

◯ 通常の並列計算とほぼ同じ流れ

× モデルによって負荷バランスが悪くなる

並列計算2：“paracon”

手順

1. 予め “paracon” を有効にしてFrontISTRを
   ビルドしておく

特徴

◯ 接触面が複数領域にまたがってもOK

× 全プロセスが全体メッシュを読むため、
計算できるモデルの規模が制限される

接触解析機能の
開発状況と今後の予定

Ver.4.4 （現在のバージョン）

- ラグランジュ乗数法で反復法を有効化（試験的）
  - 並列計算は「領域分割を行う方法」のみ
- ただし、リリース後に複数の不具合
  - 内部版では修正済み

Ver.4.4.1 or 4.5 （次期バージョン）

- 不具合修正版

Ver.5.0

- 2種類の並列化を統合化し、反復法を利用可能とする

接触解析向け反復法線形ソルバー

概要

ラグランジュ乗数法における、対角にゼロを含む非正定値行列への対処方法

1. 前処理として、ラグランジュ乗数の自由度を消去することで**正定値行列に関する方程式に変換**

2. 変換後の方程式を通常の線形ソルバーで求解
   - 問題に応じて、適切な解法・前処理を選択

3. 後処理として、消去した自由度を計算

Lagrange乗数法のための前処理 (1/3)

Lagrange乗数法を用いる場合，解くべき方程式は

$$\begin{bmatrix} K & B^T \\ B & 0 \end{bmatrix} \begin{Bmatrix} u \\ \lambda \end{Bmatrix} = \begin{Bmatrix} f \\ g \end{Bmatrix} \tag{1}$$

ここで，スレーブ自由度(s)とその他の独立自由度(p)を分離することにより

$$\begin{bmatrix} K_{pp} & K_{ps} & B_p^T \\ K_{sp} & K_{ss} & B_s^T \\ B_p & B_s & 0 \end{bmatrix} \begin{Bmatrix} u_p \\ u_s \\ \lambda \end{Bmatrix} = \begin{Bmatrix} f_p \\ f_s \\ g \end{Bmatrix} \tag{2}$$

と書く．(2)の第３式から

$$u_s = B_s^{-1} \left(g - B_p u_p \right) \tag{3}$$

これと，(2)の第２式から，

$$\lambda = B_s^{-T} \left(f_s - K_{ss} B_s^{-1} g \right) + B_s^{-T} \left(K_{ss} B_s^{-1} B_p - K_{sp} \right) u_p \tag{4}$$

を得る．これらを(2)の第１式に代入して次式を得る．

$$\begin{aligned} \left(K_{pp} - K_{ps} B_s^{-1} B_p + B_p^T B_s^{-T} \left(K_{ss} B_s^{-1} B_p - K_{sp} \right) \right) u_p = \\ f_p - K_{ps} B_s^{-1} g - B_p^T B_s^{-T} \left(f_s - K_{ss} B_s^{-1} g \right) \end{aligned} \tag{5}$$

Lagrange乗数法のための前処理 (2/3)

ここで

$$C = -B_s^{-1} B_p$$
$$h = -B_s^{-1} g. \tag{6}$$

と定義すると，(5)式は

$$\begin{bmatrix} I_p & C^T \end{bmatrix} \begin{bmatrix} K_{pp} & K_{ps} \\ K_{sp} & K_{ss} \end{bmatrix} \begin{bmatrix} I_p \\ C \end{bmatrix} u_p = \begin{bmatrix} I_p & C^T \end{bmatrix} \begin{Bmatrix} f_p + K_{ps} h \\ f_s + K_{ss} h \end{Bmatrix} \tag{7}$$

と書け，さらに

$$T = \begin{bmatrix} I_p \\ C \end{bmatrix}, \quad \begin{aligned} K' &= T^T K T \\ f' &= T^T \left(f + K \begin{Bmatrix} 0 \\ h \end{Bmatrix} \right) \end{aligned} \tag{8}$$

を定義することにより，解くべき方程式は次式となる

$$K' u_p = f' \tag{9}$$

Lagrange乗数法のための前処理 (3/3)

- アルゴリズムまとめ
  1. スレーブ自由度を選択し，B_s^{-1} を計算する
  2. (6)式により C と h を計算する
  3. (8)式により K' と f' を計算する
  4. (9)を前処理つき反復法を用いて u_p について解く
  5. 得られたu_p を(3)，(4)式に代入して u_s と λ を計算する

- **正定値行列に関する方程式に変換**されたことになる

- B_s^{-1} を計算する必要があるが，マスター・スレーブ型接触においては，B_s が対角行列になるようにスレーブ自由度を選択することが可能であり，計算コストは低い

並列化

- 現状では、１組の接触面ペアが単一領域に含まれる場合にのみ対応
  - 領域分割時は、必ず、パーティショナ制御ファイルにおいて CONTACT=AGGREGATE を指定する必要がある
  - 個々の接触面が小さく、接触箇所が多い場合はOK
  - 逆に、接触面が大きく、接触箇所が少い場合は負荷バランスが悪くなる
- この場合でも、接触状態の変化に応じて、通信パターンが変わる
  - 適宜、マスター節点と制約条件式を隣接領域にマイグレートし、通信テーブルを動的に更新している

適用事例

数値計算例1

- 大変形，弾塑性，摩擦を考慮した静解析
  - 比較的広い面で接触が発生する
- 解析規模は約4万自由度
- 反復解法にはGPBiCG法を適用

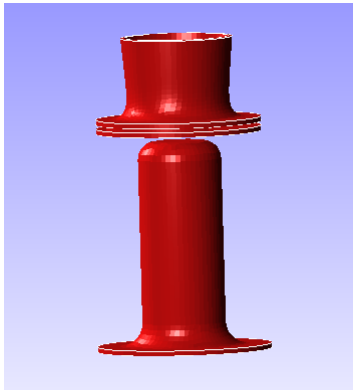


図. 解析モデル1

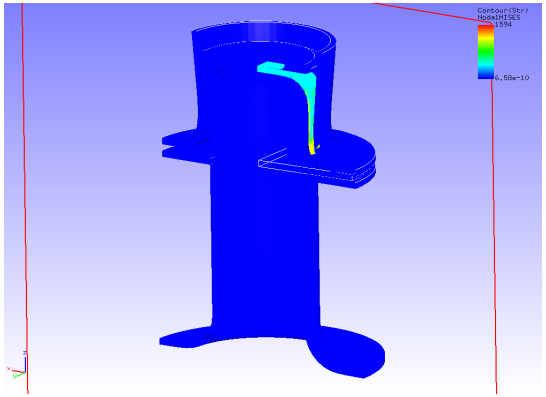
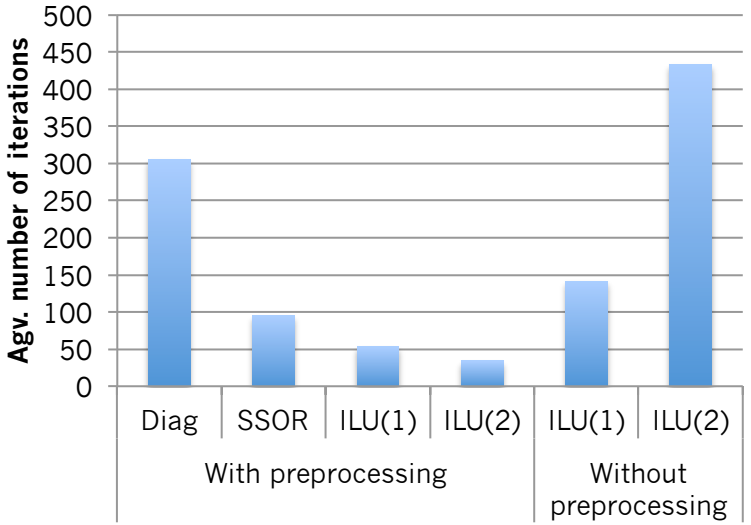


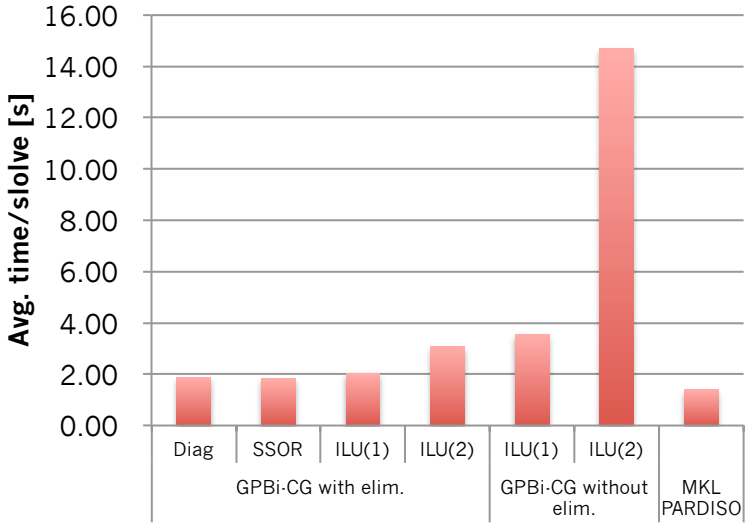
図. 解析モデル1の変形図

数値計算例1 結果

- 消去を行った場合の反復回数が少い
- 軽量な前処理 (SSOR) が速かった
  - 直接法の3~4割増し程度



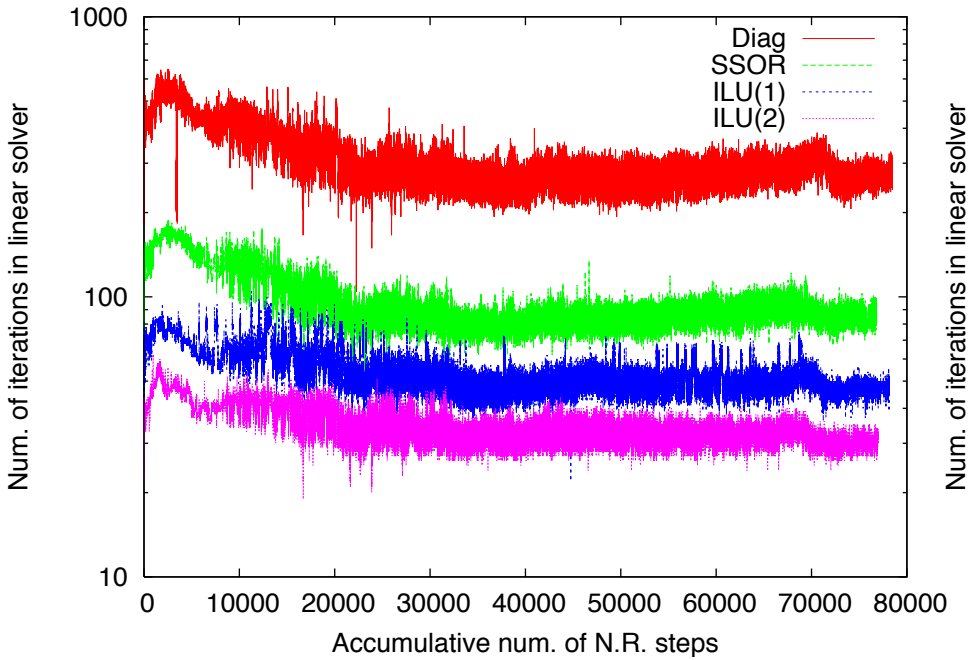
図．反復回数



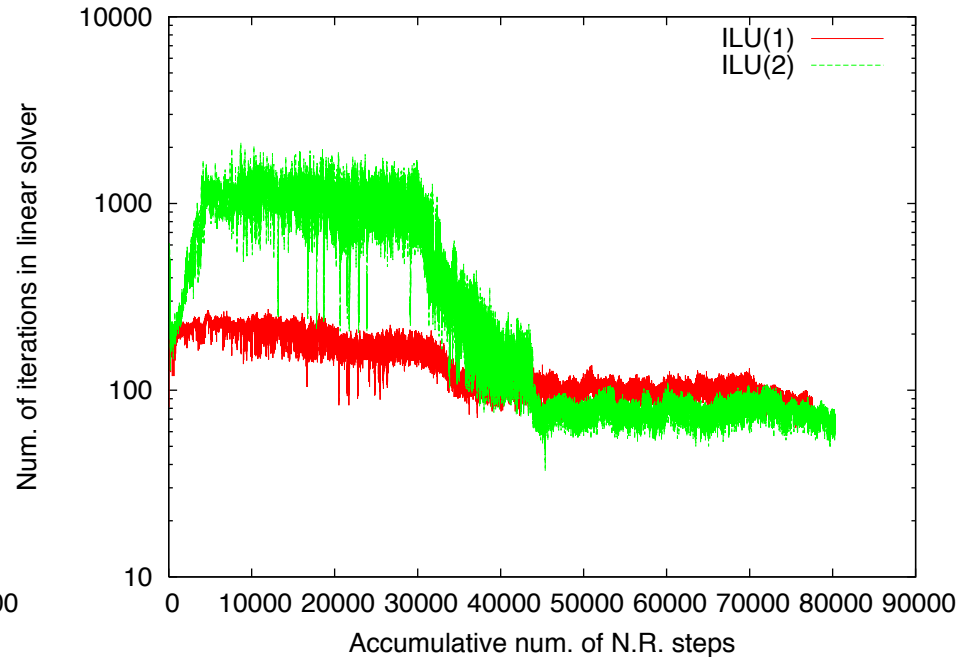
図．計算時間

数値計算例1 反復回数の履歴

- 消去を行った場合は安定して反復回数が少い
- 消去を行わなかった場合はやや不安定



図．消去を行った場合の反復回数の履歴



図．消去を行わなかった場合の反復回数の履歴

数値計算例2

- 大変形，摩擦を考慮した動解析
  - 形状は反復法に向いている
- 解析規模は約18万自由度
- 反復解法にはGPBiCG法を適用

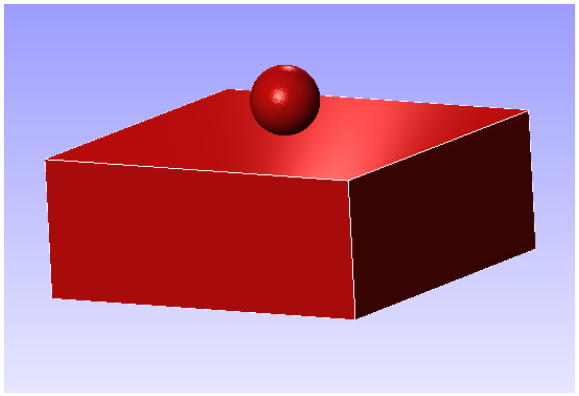


図. 解析モデル4

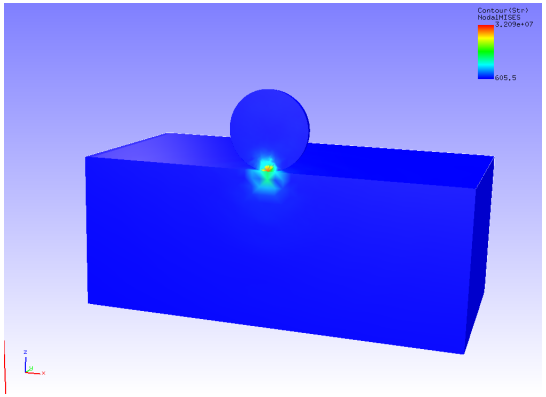
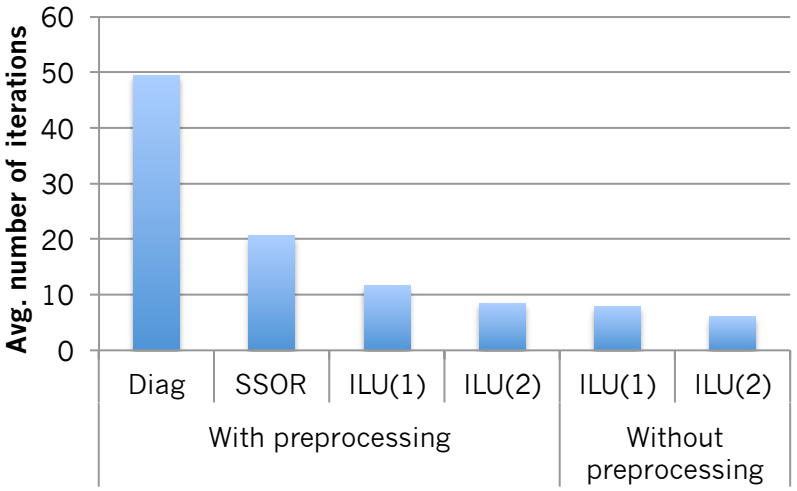


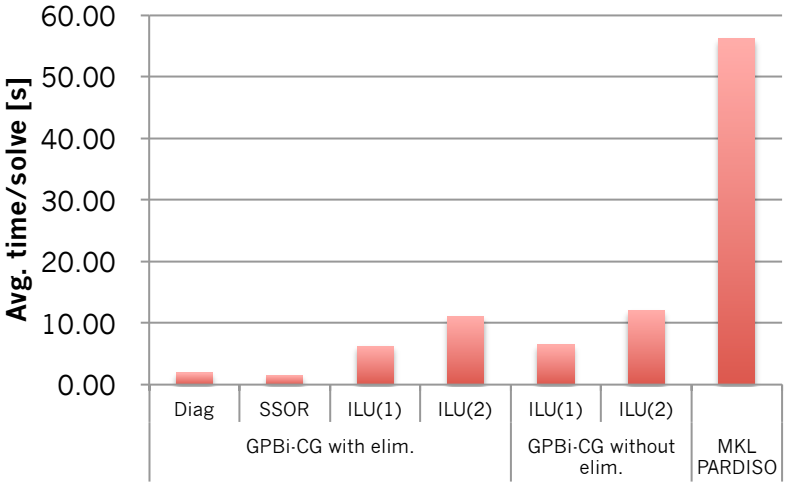
図. 解析モデル4の変形図

数値計算例2 結果

- 消去を行った場合と行わなかった場合の反復回数の有意な差はない
  - 質量行列の影響でいずれの行列も比較的解き易かったものと推測される
- 軽量な前処理 (SSOR) が速かった
  - 適用可能な前処理の幅が広がることによる利点
  - 直接法の37倍



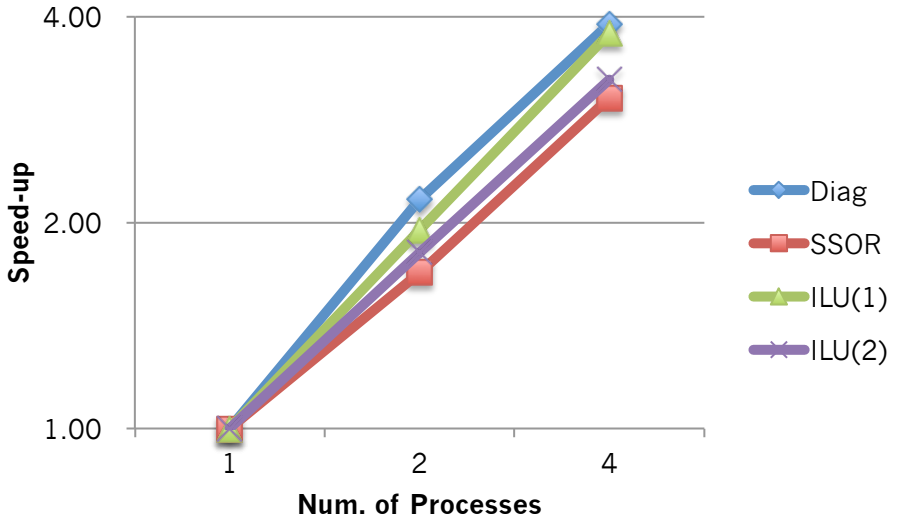
図．反復回数



図．計算時間

数値計算例2 並列計算の結果

- 計算機はPCクラスタ (Xeon X5550*2) のノード内の4コアまでを使用
- ほぼ理想的な並列性能が得られた



図．平均求解時間の並列スピードアップ

まとめ

- 静解析の事例においては,
  行列の性質が改善されることが確認された
- 動解析の事例においては,
  比較的軽量な前処理を適用することで
  大幅な高速化が達成された
- 規模が比較的大きな例題では
  計算時間は直接法と同等か，より速く求解可能であった

今後の開発計画

今後の開発計画

次期メンテナンス・リリース

– 不具合修正済のバージョン

次期マイナー・リリース

– “paracon”を、パーティショナで領域分割を行う形式に変更

次期メジャー・リリース

– “paracon”で反復法線形ソルバーを有効化