平成 25 年度　TSUBAME 産業利用トライアルユース　成果報告書

産業用ゴムベルトの有限要素法による構造解析
Structural analysis of industrial rubber belts by finite element method

徳田　明彦
Akihiko Tokuda

三ツ星ベルト株式会社
Mitsuboshi Belting LTD.
http://www.mitsuboshi.co.jp

当社は自社製品である産業用ゴムベルトの開発段階において有限要素法による構造解析を実施し、性能および耐久性を評価している。近年では製品開発における構造解析の重要性が高まり、より高精度化および高速化を行う必要が生じている。そこで本利用課題では、自社内で使用している商用非線形構造解析ソフトウェア「Marc」（エムエスシーソフトウェア株式会社）の領域分割法による並列計算を TSUBAME にて実行し、CPU の並列化および GPU の適用による高速化の効果を検証した。その結果、CPU 並列による高速化はモデル全体の要素数が多いほど効果が大きく、また GPU 適用による高速化は、分割した各領域のマトリクスサイズが大きいほど効果が大きい事を確認した。

The structural analysis by finite element method has been used in the development of industrial rubber belt products to evaluate performance and durability in our company. As the importance of the structure analysis in product development is increased, the speed and high-precision have been required. So, in this study, I have executed parallel computation in TSUBAME to verify the effect of CPU parallelization and GPU by domain decomposition method of Marc, and the both effects were confirmed.

Keywords: Belt, Rubber, FEM, Marc, DDM, GPU

1. 背景と目的

当社は産業用ベルトの製造販売を行っている。産業用ベルトはゴム材料を主体とし、補強用の心線、構造強化および摩耗防止のための帆布を組み合わせた複合材料製品である（図 1）。主体のゴムに関しても構造強化用繊維等の様々な添加物を配合し、複雑な力学的特性を持つ特殊材料となっている。そのため、その有限要素法による構造解析において十分な精度を得るためには、高精細な要素分割が必要である。

ここで、ベルト全体について理想的に高精細な有限要素分割を実施すると、その要素数は、数十万～数百万程度となる。線形解析であれば1ステップの計算で済むので、この規模でも社内計算機で対応可能である。しかし当社がベルトに対して実施している非線形解析では、これを数百～数千ステップ繰り返して計算するため、社内計算機レベルでは、実用的時間内に計算することは困難である。そのため、現状はベルト全体を粗いメッシュで計算するか、もしくは製品の一部分を切り出したモデルで計算している。そうなると、前者の場合は精度が、後者の場合は実際の力学的条件との差異による解の信頼性が低下する。

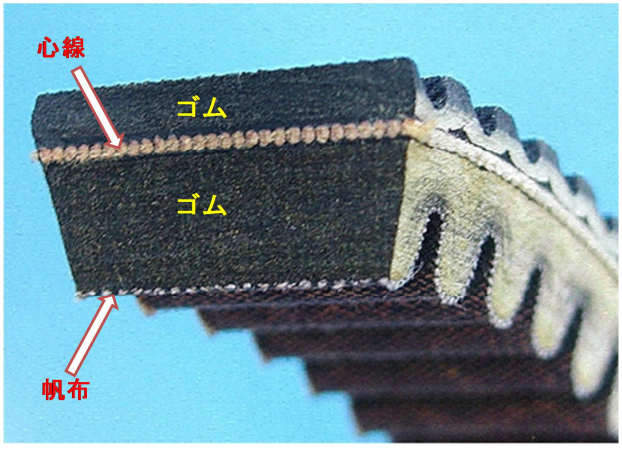


図1　産業用ゴムベルトの構造

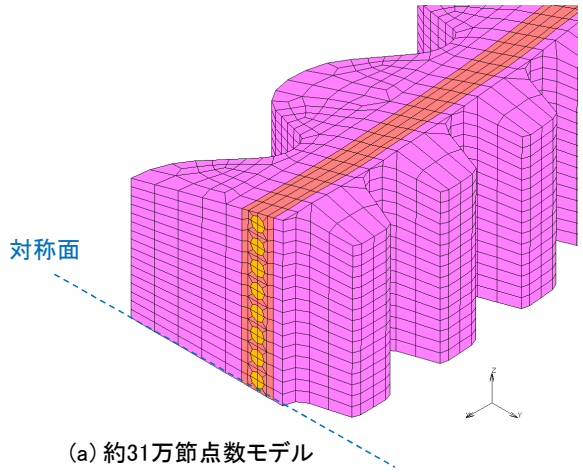
そこで本利用課題では、高精細な要素分割による解析モデルの大規模化と高速化の両立に関して、

TSUBAME における CPU 並列化と GPU 適用による解決を試み、それらが有効である事を確認した。

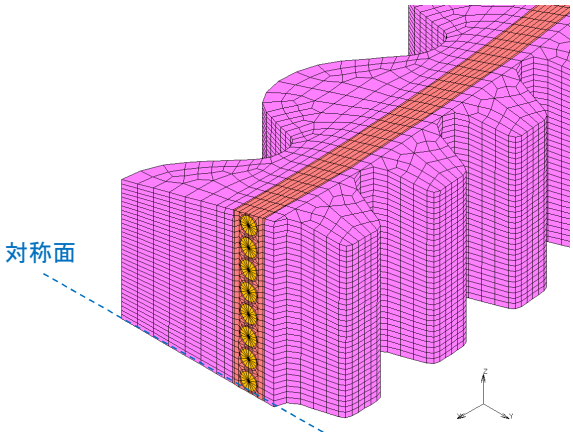
2. 概要

図 2 に、ゴムベルトの有限要素分割を示す。

図 2(a)は、要素数 279,040、節点数 311,682 の有限要素モデルである（以下、“31 万節点数モデル”とする）。



(a) 約31万節点数モデル



(b) 約118万節点数モデル

図 2　有限要素分割

図 1 でわかるようにベルトは幅方向に対称形状であるので、対称面から半分のみをモデル化している。このモデルでは分割はまだ粗く、特に心線周囲のゴム部分の応力評価に関しては不十分である。図 2(b)は要素分割を更に細かくして、要素数 1,177,600、節点数 1,185,602 とした有限要素モデルである（以下、“118 万節点数モデル”とする）。これは心線周囲の要素分割も十分精細であるが、規模が大きく、メモリ容量や計算速度の点から、社内の計算機で計算することは困難である。

また、心線に関しては引張剛性と曲げ剛性の差が大きく、図 2 のソリッド要素のみでは心線の力学的挙動を表現できない。そこで、心線の中心部に図 3 に示すトラス要素（軸方向剛性のみ有し、曲げ剛性を持たない）を配置し、ソリッド要素に比べて非常に大きな弾性率を設定する事で、「曲げやすく、引っ張りに強い」心線の特性を表現した。

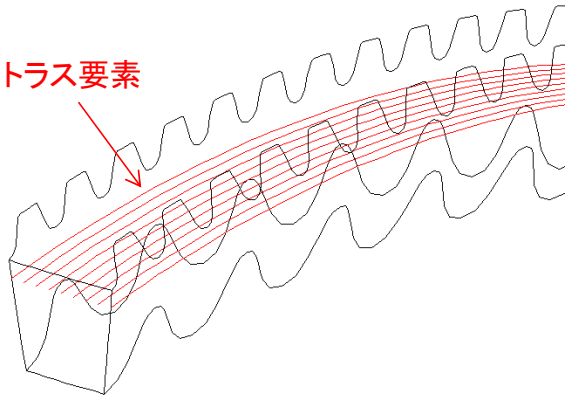


図 3　トラス要素の配置

図 4 に、解析手順を示す。

図 4(a)は初期状態である。駆動プーリおよび従動プーリは剛体面とし、ベルト近傍に配置する。図 4(b)は変形後である。従動プーリに軸荷重を与えてベルトを張り、更に負荷トルクを追加する。その状態で駆動プーリを回転させ、ベルトの運転状態を模擬する。図 4 において、巻きかけ径は駆動プーリのほうが従動プーリより小さいので、これは Low ratio の状態である。ここで、ベルトとプーリ間は摩擦力のみによって動力を伝達しているが、プーリ回転開始直後は摩擦力の状態が定常ではない。そのため、摩擦力や応力がほぼ定常になるまでプーリを回転させる必要がある。これは通常、プーリが半回転～1回転する程度は必要であり、多大な計算時間を要する。しかし本利用課題では並列数や GPU の適用有無などの条件を変え、多くの計算ケースを比較する必要が有るため、1 回の計算ケースに多くの時間を割けない。そこで、回転については、駆動プーリを回転開始から 78° 回転させた時点で打ち切る事にした。

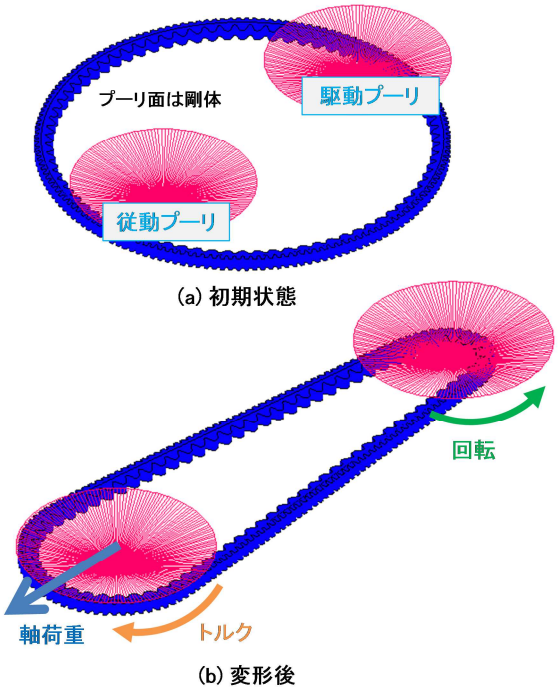


図 4　解析手順

図 5 は領域分割の一例（8 分割の例）である。

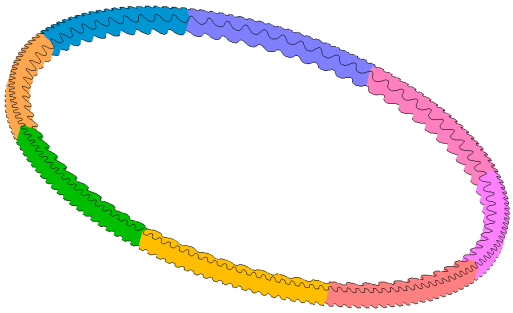


図 5　8 領域への分割例

本利用課題においては、Marc の領域分割法（DDM: Domain Decomposition Method ）を用いている。領域分割は Mentat によるプリ処理段階では行わず、Marc ソルバーにおいて自動分割しており、その結果、ベルトの長手方向に分割される。Marc の領域分割法は領域毎に剛性マトリクスを作成し解を求めるが、その後、領域間の変位と力の釣り合いを反復計算で収束させる。そのため、領域境界の節点は少ない方が良い。Marc の自動分割はそのように最適化した分割を行うため、図 4 のような分割になる。

その他、解析の設定については下記の通りである。

・ソルバー: Multifrontal（ソルバー 8）
・剛性マトリクス作成・応力計算の並列化: 無し
・摩擦モデル：Stick-Slip（摩擦係数 0.3）
・Large strain, Updated Lagrange

TSUBAME における計算は、基本的に S キュー、X キュー、または H キューを用いた。これらのキューにおいては 1 ノードが有する GPU の台数は 3 であるので、1 ノードあたりの並列数（＝領域割り当て数）は最大 3 とした。これは GPU を用いないケースにおいても同じとして、GPU の有無による速度比較を行った。

なお、並列数が少ないケースでは、1 領域あたりのマトリックスサイズが大きくなり、1 ノードのメモリ使用量が 50GB を超える場合があった。この場合は例外的に S96 キューを用いた。

3. 結果および考察

図 6 に、計算結果の応力出力例として、118 万節点数モデルの駆動プーリ内でのベルト Mises 応力分布を示す。Mises 応力で評価し、特に応力の大きい部分において、実際のベルトでも破損する頻度が多い事が分かっている。この応力値をなるべく下げる事がベルトの長寿命化に繋がる。

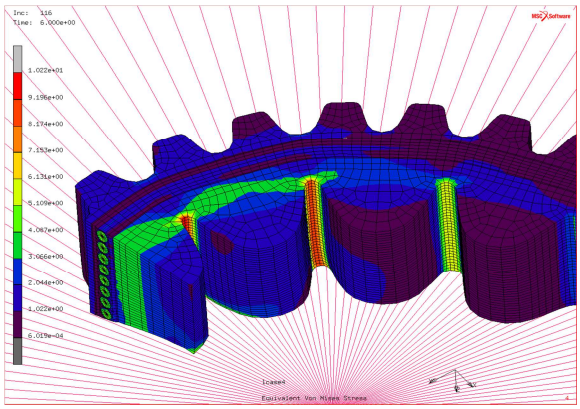


図 6　Mises 応力　出力例（118 万節点数モデル）

また、ベルト内部の応力分布も、十分に精細な要素分割数で捉えられており、内部の破損に関しても有限要

素法解析による精度良い検証が可能となる。

図 7 に、31 万節点数モデルにおける、並列数（＝領域分割数）と計算速度の関係を示す。ここで計算速度は、「並列無し（1コア）・GPU 無し」の計算速度を「1」とした相対値である。

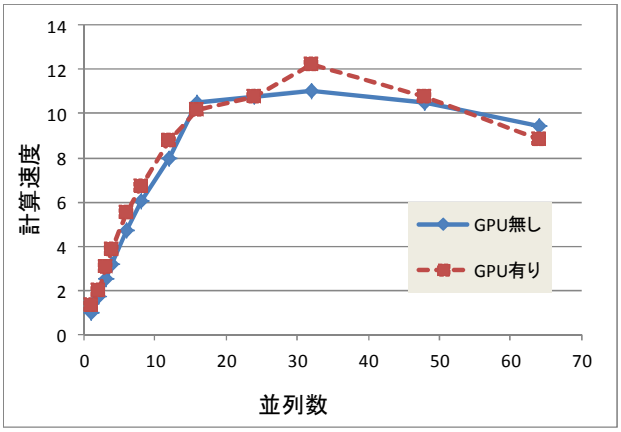


図 7　並列数と計算速度（31 万節点数モデル）

このモデルでは、並列数が 32 を超えると計算速度が低下している。これは、並列数増加による計算速度向上よりも、領域間の反復計算や通信のオーバーヘッドなどの速度低下要因のほうが上回ったためと思われる。また、GPU 適用による速度向上効果も小さい。

図 8 は、118 万節点数モデルの並列数と計算速度の関係である。

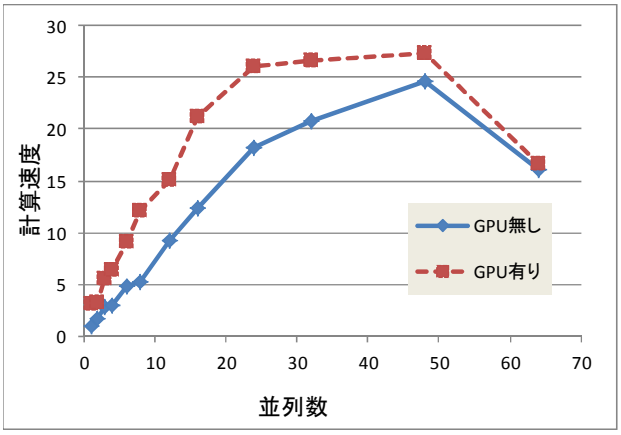


図 8　並列数と計算速度（118 万節点数モデル）

このモデルでは、31 万節点数モデルに比べて全体的に速度向上が大きく、速度低下する並列数も 48 まで伸びている。また、GPU 適用による速度向上効果も明らかである。

ここで、同モデル・同並列数において、「GPU 有りの計算速度／GPU 無しの計算速度」を、「GPU 加速率」と定義し、並列数と GPU 加速率の関係をプロットしたものを図 9 に示す（横軸を対数とした）。バラツキはあるが、モデル規模が大きく、並列数が少ないほど GPU の効果が大きい。つまり、1 領域のマトリクスサイズが大きいほど効果が大きいことがわかる。

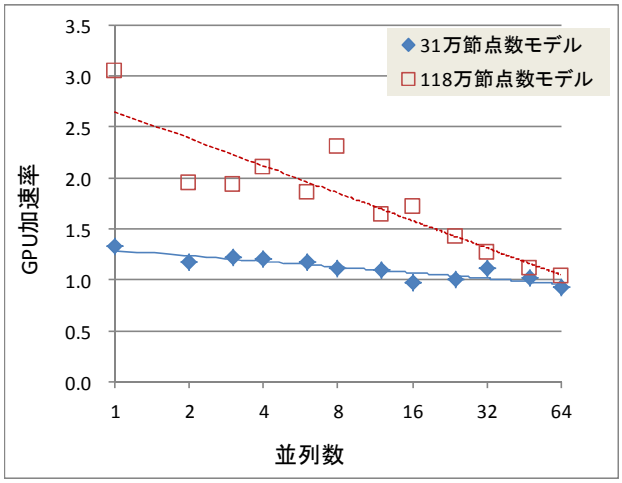


図 9　並列数と GPU 加速率

4. まとめ

(1) Marc DDM における CPU 並列化の速度向上効果は、モデル規模が大きいほど大となる。ただし、数十並列程度で頭打ちとなり、それ以上では速度低下する。

(2) GPU 適用による速度向上効果は、1 領域のマトリクスサイズが大きいほど大となる。

(3) CPU 並列化および GPU 適用により、並列無し・GPU 無しに比べ、最大 27 倍程度の速度向上効果が得られた（118 万節点数・48 並列時）。

5. 今後の課題

(1) 更に大規模なモデルにおける速度向上の検証。

(2) 剛性マトリクス作成・応力計算の領域内並列化など、今回試行していない Marc 並列オプションの効果検証。

(3) 別のベルトシステムモデルによる検証。