

GPUによる並列有限要素法の最適化

Optimization of Parallel Finite Element Method by GPU

指導教員 奥田洋司 教授

学籍番号 080909 遊佐泰紀

1 序論

計算力学は、試作品の製作コストがかかる実験に代わる手法として、今日最も注目されている学問領域の1つである。しかし、計算力学が扱える問題は比較的単純なものに限定され、大規模で複雑な問題に対しては、現実的な時間内に解が求まらないことも多い。したがって、より大規模で複雑な問題を現実的な時間内で解くために、高速な解析手法の開発が求められる。

本研究の目的は、メニーコア型プロセッサ GPU において高速な計算が期待できるブロックフォーマットの疎行列・ベクトル積を実装し、その性能評価を行うこと、そして、その疎行列・ベクトル積の有限要素解析への適用可能性を検討することの2つである。

2 有限要素法とクリロフ部分空間法

2.1 有限要素法

有限要素法¹⁾とは、連続体力学において最も使用されている計算力学手法の1つである。有限要素法において、最も計算時間を占めるのは式(1)のような連立一次方程式の求解である。

$$Ax = b \quad (1)$$

2.2 クリロフ部分空間法

クリロフ部分空間法²⁾とは、連立一次方程式の代表的な解法の1つであり、特に係数行列が大規模疎行列のときに高速な求解を期待できる。有限要素法などの計算力学手法が生成する係数行列は、一般に大規模疎行列である。

3 疎行列・ベクトル積

3.1 BLAS

BLAS (Basic Linear Algebra Subprograms) とは、行列とベクトルに関する線形代数の基本演算を行うサブプログラムである。BLAS1, BLAS2, BLAS3 の3種類に分かれる。

3.1.1 BLAS1

BLAS1 はベクトル同士の演算を担当する。クリロフ部分空間法では、内積(式(2)), ベクトルのスカラー倍と和(式(3), (4))を利用する。

$$\text{DOT}(x, y) = x^T y \quad (2)$$

$$\text{AXPY}(x, y): y \leftarrow \alpha x + y \quad (3)$$

$$\text{AYPX}(x, y): y \leftarrow \alpha y + x \quad (4)$$

3.1.2 BLAS2

BLAS2 は行列・ベクトル演算を担当する。クリロフ部分空間法では、式(5)の SPMV (SParse-Matrix Vector multiplication)を利用する。

$$\text{SPMV}(A, x, y): y \leftarrow \alpha Ax + \beta y \quad (5)$$

3.1.3 BLAS3

BLAS3 は行列同士の演算を担当する。クリロフ部分空間法では利用しない。

3.2 疎行列格納フォーマット

疎行列をメモリに格納するとき、多くの場合はゼロ要素を削除してメモリの有効利用を図る³⁾。本研究では、代表的な疎行列格納フォーマットである CSR (Compressed Sparse Row) と ELLPACK のブロックフォーマット(BCSR と BELLPACK)の SPMV を実装した。ブロックフォーマットとは、行列をブロックに分け、非ゼロ要素が存在するブロックを密行列としてメモリに格納するフォーマットのことである。

4 GPU

GPU (Graphics Processing Unit)は、コンピュータの画像処理を担当する演算装置である。コンピュータの総合的な処理を担当する CPU (Central Processing Unit)が数個の比較的複雑なプロセッサで構成されるマルチコア型プロセッサなのに対して、GPU は数百個の比較的単純なプロセッサで構成されるメニーコア型プロセッサである。

5 チューニングと高速化手法

5.1 スレッド数のチューニング

GPU における実行スレッド数は、プログラマーが任意に決定することができるが、スレッド数のチューニングは非常に難しい。本研究では、レジスタに注目してこのチューニングを行った。1 スレッドが消費するレジスタ数とスレッド数との積が、GPU の仕様によって決まる最大レジスタ数を超えないような条件の下でスレッド数を最大化した。

5.2 ブロックフォーマットのゼロパディング

SPMV の高速化のためにはメモリアクセスの最適化が重要である。本研究では、キャッシュヒット率向上とメモリアライメントのために、各ブロックをゼロパディングしてメモリに格納するブロックフォーマット(以下、BCSR4, BELLPACK4 と呼ぶ)を開発した。

6 疎行列・ベクトル積のベンチマーク実験

SPMV のベンチマークテストを行った。高速化が確認できた BCSR4 フォーマットの結果を図 1 に示す。横軸はテスト行列のサイズ、縦軸はパフォーマンスである。本研究で使用した GPU では、 2×2 の BCSR4 フォーマットにおける高速化を確認できた。また、現行の GPU においては、行列が少なくとも 10 万程度のサイズがないと、十分なパフォーマンスを得られないことがわかった。

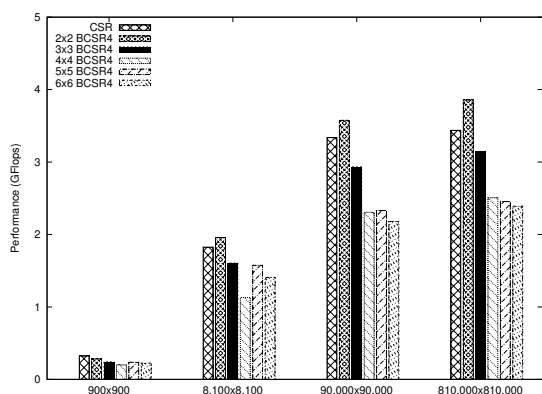


図 1 BCSR4 の SPMV ベンチマーク

7 有限要素解析への適用

7.1 有限要素解析

本研究で開発した疎行列・ベクトル積を用いた有限要素解析を行った。有限要素法が生成する剛性行列は、節点を持つ自由度の影響から、非ゼロ要素の分布がブロックフォーマットと相性の良いものになる。そのため、SPMV には 6 章で高速計算を確認した BCSR4 フォーマットのものを用いた。また、比較のために CPU においても同条件で解析を行った。

7.2 解析対象モデル

v6engine, frame, drill の 3 つのモデル(図 2)に対して応力解析を試みた。

7.3 解析結果と考察

v6engine では、CPU と同程度の反復回数で、CPU の約 4 倍の速度での収束を確認できた。

frame は、BCSR4 フォーマットにおいてメモリの容量不足により解析ができず、BCSR フォーマットでの解析を行った。また、frame の剛性行列は v6engine のと比べて条件が悪く、前処理なしでは収束を確認できなかった。そこで、比較的単純な前処理であるヤコビ前処理を使用して解析を行ったところ、v6engine 同様の結果を得た。しかし、CPU において、強力な前処理を使用して同条件の解析を行ったところ、反復回数が少なく、GPU に迫る解析速度であった。

drill は、frame と同じくメモリ容量不足により解析ができなかった。また、BCSR フォーマットにおいても同様に解析ができなかった。

以上から、GPU においても CPU と同等の解析が行えることがわかった。その計算速度は、CPU の約 4 倍となり十分な高速化を確認できた。しかし、メモリ容量不足により解析ができない問題が存在し、解析できる問題の規模は GPU のメインメモリ容量と、使用する疎行列格納フォーマットに大きく依存することがわかった。具体的に、現行の GPU において解析できる問題の規模の上限は百数十万程度であった。



図 2 3 つの解析対象モデル

8 結論と今後の展望

GPU において、BCSR フォーマット、特に 2×2 の BCSR4 フォーマットにおける疎行列・ベクトル積の十分な高速化を確認することができた。しかし、BELLPACK フォーマットにおいては十分な高速化を確認できず、更なる高速化手法の導入が求められる。また、メニーコア環境において本手法の性能を十分に発揮させるためには、現行の GPU において、問題の自由度が少なくとも 10 万程度必要であることがわかった。

本研究で開発した疎行列・ベクトル積を用いた有限要素解析では、CPU と同程度の反復回数で、約 4 倍の高速解析に成功し、本手法が有限要素解析に十分適用可能であることを確認した。しかし、GPU で解析可能な問題の規模は、GPU のメインメモリ容量に大きく依存し、現行の GPU における上限は百数十万自由度であった。したがって、現行の GPU 1 枚で解析するのに適している問題の自由度は、10 万程度から百数十万程度に限定される。それ以上の規模の問題を解析するためには、GPU を複数利用した分散メモリ環境を用いる必要があることがわかった。また、強力な前処理のメニーコア並列実装は困難であるため、GPU のようなメニーコア環境は、悪条件問題に対して脆弱であることがわかってきた。したがって、メニーコア環境でのロバストな解析を行うために、今後は、既存の前処理手法のメニーコア並列実装や、効果的な新規前処理アルゴリズムの考案が求められる。

参考文献

- 1) 吉村忍, 矢川元基: 有限要素法, 計算力学と CAE シリーズ 1, 培風館, 1991.
- 2) Y. Saad: *Iterative Methods for Sparse Linear Systems*, SIAM, 2000.
- 3) R. W. Vuduc: *Automatic Performance Tuning of Sparse Matrix Kernels*, PhD Thesis, UC Berkeley, 2003.