

1. 研究概要

様々なモデルに利用される方程式は最終的には連立一次方程式にたどり着くことが多い。したがって連立一次方程式の求解を高精度で速く解くことができれば、より精度の高いシミュレーション結果を得ることができる。しかし、double型よりも長い多倍長ライブラリはデータ量、計算量共に多く遅い。そこで今回は、CPU上での並列化のAPIであるOpenMPを用いたマルチスレッド演算と、線形演算ライブラリであるMAGMAの関数を用いてグラフィックボード上のGPUを利用し、多倍長混合精度反復改良法の高速化を行った。混合精度反復改良法とは反復法の一つで、連立一次方程式の近似解を計算速度の速いS桁で計算し、L桁で残差を求め、これを用いて再びS桁で計算するものである。使用した多倍長ライブラリは、double型を2つ組合せたDD、4つ組合せたQD、任意精度のMPFR/GMPである。多倍長精度浮動小数点数は倍精度では不足する悪条件問題に有効である。開発環境は以下に示す通りである。

- ・CPU: Ryzen7-1700
- ・GPU: GeForce GTX 1050Ti
- ・OS: Ubuntu 18.04

2. ベンチマーク結果

表1

threads \	MPFR	double	DD	QD
1	1	1	1	1
2	0.6	0.5	0.49	0.51
4	0.34	0.29	0.29	0.31
8	0.18	0.16	0.15	0.16

表1は500次元の係数行列AをLU分解したとき、並列化前を1とした時の計算時間の比である。並列化によって、計算時間を大幅に減らすことができた。

図1はMPFR-doubleの混合精度反復改良法のベンチマーク結果である。MPFRはCPUで、doubleはMAGMA関数を用いてGPUで実行した。次元数が少ない時はデータをCPU

からGPUへの転送、GPUからCPUへの転送に時間がかかっているためCPUの方が速い。しかし、次元数が増えると並列計算の速いGPUで計算する方が速くなる。

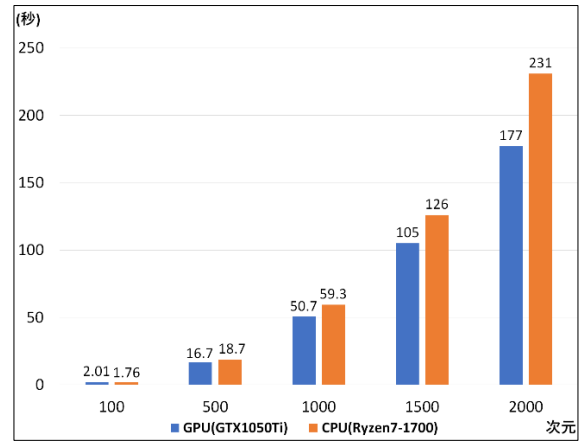


図1

3. 結論

500次元の係数行列Aの連立一次方程式を直接法MPFR、混合精度反復改良法MPFR-double、DD、QD、混合精度反復改良法MPFR-doubleのGPU利用を解の精度を高めながら解いた時のベンチマーク結果が図2である。並列化しても直接法より反復法の方が常に速い。同じMPFR-doubleの組み合わせで比較するならCPUよりGPUで計算する方が解が10e-1000以降速いが、反復回数が増え、L桁の残差計算に時間がかかるため反復回数の少ないDDやQDとの組み合わせが次第に速くなっていくことがわかる。

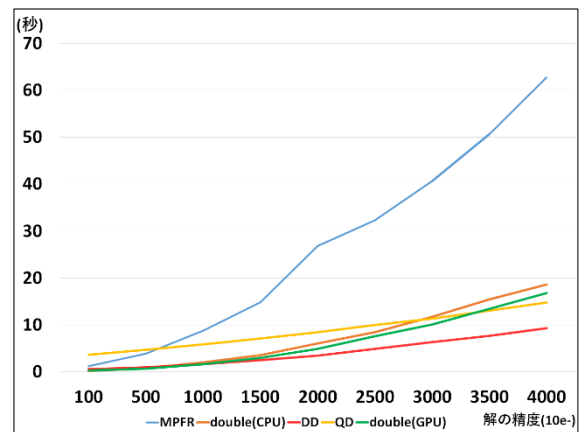


図2