GPU を用いたインタラクティブ流体・構造連成シミュレータの３次元拡張

鈴木大陽[†] 西村祐介[†] 福間慎治[†]
森眞一郎[†] 山口明徳[††] 富田眞治[††]

1. はじめに

GPU に代表される高性能アクセラレータを用いた対話的な実時間シミュレーションへの期待が高まっている．このようなインタラクティブ・シミュレーションの一例として，我々はインタラクティブ流体・構造連成シミュレーションの研究を行っている．本論文では，我々が開発した管内流に対する流体と構造の 2 次元連成シミュレータ[1)2)] の 3 次元拡張と，また，インタラクティブ性を高めるための高速化手法の検討ならびに一部実装の報告を行う．

2. 流体・構造連成シミュレータ

図 1 に流体と構造の連成シミュレーションの処理の流れを示す．構造シミュレータではユーザからの入力と流体抗力 (以下，抗力) による管構造の変形計算を行い，流体シミュレータでは構造計算の結果から管内壁の位置を更新し，流体計算および抗力計算を行う．連成シミュレーションには，両方のシミュレーションを独立に行う弱連成モデルを採用する．図 1 の外部入出力は，管に強制変形を加える場合の内壁座標の入力および管壁からの反力を提示するために用いる．なお，これまでに実装した連成シミュレータでは，GPU で格子点情報の更新と流体計算を行い，CPU で抗力計算と構造計算を行っている．そのため，抗力計算時には GPU での流体計算結果を CPU へ，管内壁の更新時には CPU での構造計算結果を GPU へそれぞれ転送する必要がある．

3. シミュレーションモデルの 3 次元拡張

これまで，2 次元連成シミュレータのシミュレーションモデルとして，管壁の変形自由度に制限を加えたバネモデルで構造変形を近似してきた．このバネモデル

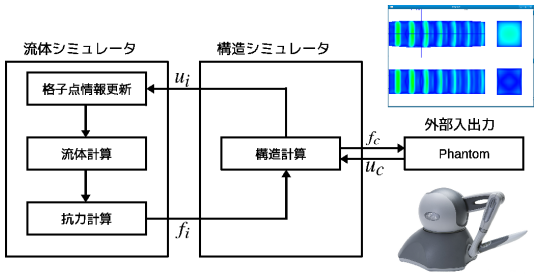
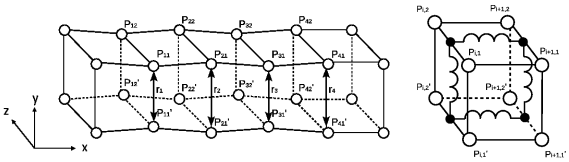


図 1 流体と構造の連成シミュレーション

に新たなバネを追加し，それを 3 次元拡張することで図 2 のような構造変形モデルを実装した．図 2(a) において，4 つのノードで構成する面で管壁を形成しており，図 2(b) のように辺と辺の間にバネがつながっていることを想定している．また，構造変形に対する制約として，$P_{i,1}$，$P_{i,2}$，$P'_{i,1}$，$P'_{i,2}$ の X 座標，Z 座標は固定とし，Y 座標のみの移動に制限するとともに，$P_{i,1}$ と $P_{i,2}$(あるいは $P'_{i,1}$ と $P'_{i,2}$) の Y 座標は同時かつ，同じ値に変化するものとする．管壁は，図 2(b) の辺に対する流体からの y 方向 (図 2 の上下方向) からの圧力を受けることでそれぞれ移動する．



(a) 壁面構造モデル (b) バネモデル

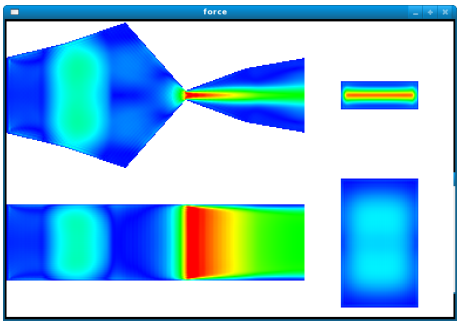
図 2 構造変形モデル

4. 3 次元シミュレータの実装と高速化

4.1 実装結果

3 次元連成シミュレータを実装し，GPU 上での処理は Cg 言語で記述した．実行環境は，CPU：Core2Duo 6300 1.86GHz，GPU：GeForce GTX であり流入条件を拍動流，流出条件を自由流出とした．なお，管形

† 福井大学
University of Fukui
†† 京都大学
Kyoto University



左上：XY 平面 ($z = Z_{max}/2$)　　右上：YZ 平面 2 (下流側)
左下：XZ 平面 ($y = Y_{max}/2$)　　右下：YZ 平面 1 (上流側)

図 3　シミュレーションの様子（流速表示）

状を決める制御ポイントは管軸方向（X）を 5 区間に等分割して配置した．拍動流を入力として与え，この流れに対する抗力と外部入力によって管形状（管幅 y）が変化する流体・構造連成シミュレーションを行った．図 3 はシミュレーション実行中の様子であり，各タイムステップでの流速分布（XY 平面, YZ 平面、XZ 平面 [管の断面 2 個所]）を色で示したものである．流速が高い場所が赤系，低い場所が青系の色で表現されている．図では、管形状を決める 4 個所の制御ポイントのうち左から 2 番目の制御ポイントを最大管幅，3 番目の制御ポイントを最小管幅に強制変形し，残りの 2 個所は自由変動させたときの様子である．なお，流入端と流出端の管幅は固定である．

1 ステップでの各処理時間を，1000 ステップ実行時の平均時間として表 1 に示す．なお，図 1 における格子点情報の更新処理は流体計算に，構造計算は抗力計算に含めているものとする．各処理時間のうち，流体抗力の計算とそのために必要な通信にかかる時間は，シミュレーションサイズの増大による処理時間の増加が大きいため，これらの処理時間を短くすることが重要となる．

表 1　3 次元連成シミュレーションにおける各処理時間

	シミュレーションサイズ（$X \times Y \times Z$）		
各処理時間 [ms]	60×32 $\times 32$	120×64 $\times 32$	240×128 $\times 32$
流体抗力用通信	4.10	11.4	35.2
画面表示用通信	6.05	6.16	11.5
流体抗力の計算	1.21	4.36	17.6
流体計算	12.0	12.2	12.2
全体	23.7	34.6	77.0

4.2　不要なデータ転送の削減

次に，抗力関連の処理の高速化手法として，まず不要なデータ転送の削減することを考える．抗力計算には，GPU での流体計算のデータを利用するが，そのうち必要となるデータは管壁に隣接する極一部に限られる．しかしながら従来の実装では GPU 上での分岐処理を減らす目的で全領域のデータを転送していた．そこで，管壁に隣接する領域のみを抽出して通信することで，データ転送量を削減することによる高速化効果の検討を行った．その結果，最大で約 1.7 倍の高速化をシミュレーション全体で達成できた．一方で，問題サイズが小さい場合には GPU での分岐処理によるオーバヘッドのため逆に 8%程度の速度低下となった．

4.3　抗力計算および構造計算の GPU 実装

抗力計算や構造計算には分岐命令が多く，GPU 向きでないため，今まで CPU 上で実装してきた．しかしながら，そのために必要となる GPU-CPU 間の通信が性能のボトルネックとなってしまった．そこで，これらの処理を GPU に実装し，GPU の計算効率を犠牲にしつつも，データ転送をなくすことによる高速化を検討した．ただし，これらの処理と Cg 言語の親和性が良くないため，CUDA 言語を使用して実装し，この部分だけの計算時の処理時間を測定した．その結果，CPU で抗力計算を行った場合と比べて，同程度あるいは高速に実行可能であることを確認できた．また，抗力計算のための通信の必要がなくなるため，大幅な高速化が期待できることも確認した[3)]．

5. ま と め

3 次元連成シミュレータを実装し，データ転送量の削減を行うことによって高速化を達成した．また，CUDA 言語を使用した抗力計算の実装により，更なる高速化への期待が高まった．今後は，連成シミュレータの CUDA 実装を行う予定である．

謝辞　本研究の一部は日本学術振興会科学技術研究費補助金（基盤研究 (S)16100001，基盤研究 (C)22500044）の補助による．

参 考 文 献

1) 山口　明徳，”GPU を用いたインタラクティブ流体・構造連成シミュレータの構築 ”，京都大学大学院情報学研究科　修士論文 (2009)
2) 永田　佑輔，”流体・構造連成シミュレーションモデルの検討 ”，福井大学工学部情報・メディア工学科　卒業論文 (2010)
3) 西村 祐介, 鈴木 大陽, 福間 慎治, 森 眞一郎, 山口 明徳, 富田 眞治, ”GPU を用いたインタラクティブ流体・構造連成シミュレータのプロトタイプ実装,” 計算工学講演会論文集, Vol.16, F-4-2, 2011 年 5 月.