先端的計算科学研究プロジェクト報告書（ベンチマーク課題）

正方形ダクト乱流の高精度直接数値計算コードの GPU 加速

大阪大学　基礎工学研究科　物質創成専攻　化学工学領域

関本 敦

【研究目的と成果概要】

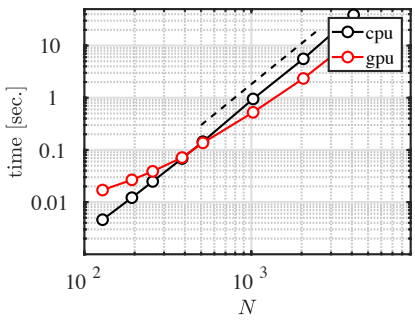
世界最大規模の正方形ダクト乱流の直接数値シミュレーション(DNS)にチャレンジし，海外の平行平板間チャネル乱流や境界層乱流データベース(https://torroja.dmt.upm.es/turbdata/index/, レイノルズ数 Re_τ は 1000 から最大 4200)と肩を並べる乱流データベースを作ることを目的とした GPU 加速 DNS コードの作成および性能評価を行う．これらの大規模乱流データベースは乱流の基礎研究ならびに乱流モデルの改良に用いられる．

計算機の性能向上がヘテロジーニアス環境（GPU などのアクセラレータを利用した環境)で維持されているという現状があり，その計算機環境に合わせた乱流の高精度 DNS コードの開発が必要である．チャネル乱流や境界層乱流の DNS は，壁面に垂直な方向が 1 方向だけで，スペクトル法およびコンパクト差分法を用いた大規模数値計算においても，ヘルムホルツ方程式のソルバー部分が 1 次元になるため，並列計算性能は大きく保てる．しかしながら，正方形ダクトの DNS では壁面に垂直な方向が 2 方向あるため，解くべきヘルムホルツ方程式が 2 次元となる．ヘルムホルツ方程式を解くためには，行列積を用いる為，ダクト断面の 1 辺の格子数を N とすると $O(N^3)$ で計算コストが増大する．このため計算規模を大きくした場合の並列性能が大きく低下する．そこで，本課題では，GPU を活用することで行列積の計算時間の大幅な減少を図り，高レイノルズ数（Re_τ=2000 程度，$N \fallingdotseq 1000$）の DNS を実現するための，GPU を活用した 2 次元ヘルムホルツソルバーの開発，およびダクトコード全体の並列性能評価を行った．

【成果の詳細】

1．ヘルムホルツ方程式ソルバーの高速化

図 1 に 2 次元ヘルムホルツ方程式ソルバーの GPU 加速の結果を示す．1 辺が N の大きさの行列積の演算を用いるため，CPU と GPU の計算のどちらでも，$O(N^3)$ で計算コストが増大しているのが分かる．GPU 計算の場合は CPU—GPU 間のデータ転送時間も全体の計算効率に大きく関わるため，図での比較では，データ転送時間も含めた公平な比較をした．N が 1000 程度で GPU での計算の方が CPU よりも高速であることがわかる．行列積の計算には cuBlas のライブラリ(cublas_dgemm)を用い，また，行列積の演算は $A^{\mathrm{T}}B$ が一番効率良く計算できたため，適切に行列の転置操作を GPU 上で行なう実装にした．Fortran コードから cuBlas を呼び出すだけで，高速化を図れるが，行列の転置操作は NVIDIA Developer Blog のコード(An Efficeint Matrix Transpose in CUDA C/C++)を参考に実装し，Coalesced Transpose Via Shared Memory が一番高速であった．



1 辺の格子点数

図 1.　2 次元 Helmholtz 方程式ソルバーの GPU 加速性能．
黒線は cpu での計算時間，赤線は GPU を用いたときの計算時間（CPU から GPU へのデータ転送時間を含む)．波線は N^3 の傾きを表す．

2．正方形ダクト乱流 DNS コード全体の性能評価

Helmholtz ソルバーを高速化できたことで，全体の計算時間に対する非線形項の割合が高まった．主な原因は FFT を流れ方向に適用する際にデータのアクセスが飛び飛びになるためであり，all-to-all 集団通信の後に配列の転置操作を加えることで，FFT 計算の効率化を図った．3072x1025x1025 の格子点数で 32 ノードを用いたときの，1 ステップの計算時間が 78.0 sec. から 12.2 sec.へと短縮できた．これは，プロダクトランのための実用的な計算時間である．チェビシェフ変換を適用するときにも同様の転置操作をすることで，更なる効率化を図ることができるが，数%程度の速度向上に止まる．

コード全体としては，次に示すように，全体に占める MPI 集団通信がボトルネックとなる． 図２は，全体最適化を行なったコードの並列性能を Weak scaling によって調べたものである．4 ノードあたりの流れ方向の計算格子数を 384 として測定すると，32 ノードまでは，並列効率を 8 割に維持できているが，64 ノード以上を用いた計算では MPI 通信が 8 割以上を占めるため，並列化効率が落ちる．これは GPU 加速が上手く実装できたことの裏返しでもあり，スペクトル法を用いた大規模 DNS 計算共通の課題である．

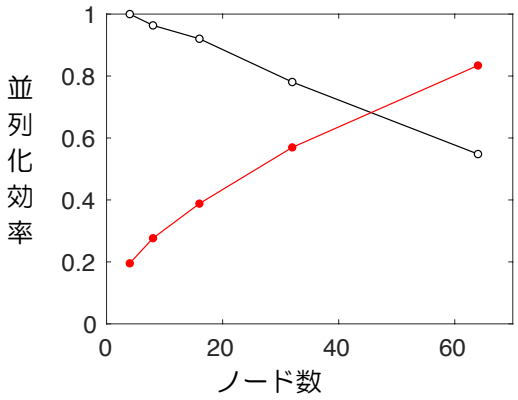


図 2.　GPU-duct DNS コードの並列性能評価（weak scaling).
物理格子点数は(*Nx*×*Ny*×*Nz*) (96*node) ×1025 (*y*)×1025(*z*).
黒線は 4 ノードの実行計算時間を基準とした並列化効率を表し，赤線は全体に対する MPI 通信時間の比率を表す．