

チャンネル乱流場における低通信スペクトル法 DNS コードの開発とその性能評価

山本 義暢

山梨大学大学院総合研究部

1 はじめに

乱流は依然として古典物理学における未解決の問題の一つであり、その解明が困難なのは、非線形かつ散逸的な力学系に内在する膨大な自由度に起因している。その極めて高い次元性とランダム性にもかかわらず、普遍的な法則やスケーリング関係の存在が理論的に予測されてきた。これらの独特な性質は流体運動の顕著な特徴であり、気象学や工学における抵抗、ならびに熱・物質輸送を予測するための極めて重要なメカニズム的洞察を提供している。このような乱流理論を検証する上で、第一原理に基づいてナビエ・ストークス方程式を数值的に解く直接数値シミュレーション (DNS) が、最も有力な手法であることが実証されている。工学的に重要な壁面境界付き乱流である乱流チャンネル流 (Turbulent Channel Flow 以下、TCF と呼ぶ) の DNS は、1987 年に達成された [1]。それ以来計算リソースの進歩により、扱えるレイノルズ数が継続的に拡大し、DNS は乱流研究にとってますます強力なツールとしての地位を確立している。

高レイノルズ数の DNS を実現する上での主な課題は、計算スケールの拡大に伴う計算効率の著しい低下にある。具体的には、TCF の DNS における自由度の数は Re_τ^3 に比例してメモリを必要とする一方、計算コストは Re_τ^4 に比例する。ここで、 Re_τ は、流れ領域 h のサイズ (例えば、図 1 に示すような流路の半高さ) によって決まる最大の渦と、粘性長さスケール $l_\tau (= \nu/u_\tau)$ によって決まる最小の渦によって定義される: $Re_\tau = h/l_\tau$ 、ここで、 ν は動粘性係数、 u_τ は摩擦速度、である。

DNS の計算効率はスーパーコンピュータシステ

ムのバイセクション帯域幅に依存するが、最近のスーパーコンピュータシステムのバイセクション帯域幅は、計算性能の向上に対して著しく遅れをとっている。その結果、高レイノルズ数 DNS の計算効率は低下しており、その実現が妨げられている。この課題を克服するためには、特に計算能力は高いが帯域幅が乏しい近年のスーパーコンピュータシステムにおいて、バイセクションバンド幅への依存度を低減する高精度なスキームと効率的な並列計算手法を確立することが不可欠である。

本研究では、スペクトル精度を維持しつつ、メモリおよび通信要件を大幅に削減するハイブリッドなスペクトル・有限差分 DNS コードを開発することで、これらの課題に取り組んでいる。

2 従来 DNS コードの概要 [2], [3]

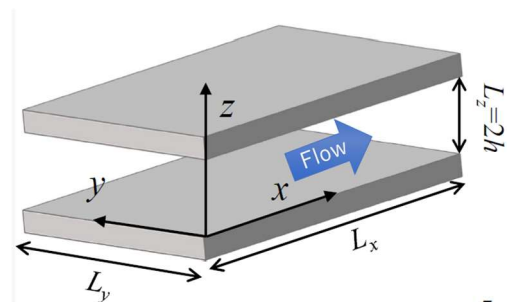


図1 チャンネル流と座標系

本研究で対象とする流動場は図1に示すチャンネル流であり、主流方向(x)への一定の圧力勾配により駆動される流れ場である。流れ場の支配方程式は非圧縮性流体の Navier-Stokes 式と連続式となる。数値計算手法としては、周期境界条件が適用できる壁水平方向(x, y)にフーリエ・スペクトル法、壁垂直方向

(z)方向に2次精度中心差分法を適用する。境界条件は主流(x)およびスパン方向(y)に周期境界条件、壁面($z=0, 2h$)でno-slip条件を課す。時間積分は圧力項にEuler陰解法、その他の項に2次精度Adams-Bashforth法を適用し、Fractional step法により連続式とカップリングする。圧力のポワソン方程式は、波数空間(k_x, k_y)において、壁垂直方向への3重対角行列をTDMA(Tri-Diagonal Matrix Algorithm) [2]により直接解く。

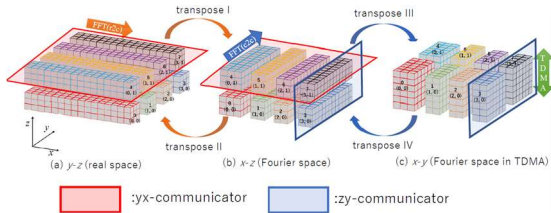


図2 2次元領域分割

並列化手法は図2(a)に示す y および z 方向への2次元領域分割を適用する。フーリエ・スペクトル法における x, y 方向への2次元フーリエ変換は高速フーリエ変換(Fast Fourier Transform, FFT)を用いるが、FFTは並列性を持たないために領域分割方向の計算において並列性を確保するために、領域分割軸の変更(y 軸 $\rightarrow x$ 軸, 図2(b)参照)を行う。この軸変更はコミュニケータ内のAll-to-all通信(A2A)となり、通信負荷が極めて高く、その性能はシステムのバイセクションバンド幅に依存する。

さらに壁垂直方向のTDMAにおいても一般には並列性を持たないため、同様に分割軸の変更(z 軸 $\rightarrow y$ 軸, 図2(c)参照)により並列性を確保する。この転置も同様にコミュニケータ内でのA2Aを伴う。従って本コードでは、1ステップあたりの時間進行において、擬スペクトル法に伴う非線形項に起因する18回のA2A(yx -communication)と圧力ポアソン解法における2回のA2A(zy -communication)とA2Aが発生し、その回数はアンバランスとなっている。そのため通信負荷を低減するうえでは、TDMA方向(zy -communicator)の並列数を多く設定することが得策となる。

3 低通信スペクトル法コードの開発

このスペクトル法における通信負荷を低減するために、本研究では以下の改良を行った。

(1) A2Aを伴わないTDMA並列手法の開発

A2Aを伴わないTDMA並列化手法は複数知られているが(例えば、[5], [6])これらは領域分割方向に十分な格子数が確保できる場合を前提としている。一方上述の通り、本DNSコードでは、TDMA方向の分割数を多くとることが得策であるため、MPIプロセスあたりの z 方向格子数は小さくなり、これらの手法を適用しても並列効果が十分に得られないと考えられる。そこで本研究ではTDMA自体は非並列(完全逐次)とし隣接シフト通信により行い、TDMAと独立な k_x, k_y 方向をオーバーラップさせることにより並列性を確保する新手法を開発した。本手法は、並列化に伴うメモリ増がほぼゼロであり、通信タイプが隣接シフト通信のみで行えることが大きな特徴である。

(2) FFT演算とA2Aのオーバーラップ手法の開発

一方FFT方向(yx -communicator)では依然としてA2Aを伴うが、このコミュニケータをノード内あるいは隣接ノード間に配置すれば、高速なA2Aが期待できる。さらにFFT演算とA2Aをオーバーラップさせることにより、A2Aコストの隠ぺいを図った。同様にこの操作に伴うメモリ増はほぼゼロである。

4 ベンチマークシステム

本開発手法の性能評価として、京都大学学術情報メディアセンターのシステムA(Camphor3)の1024ノードを使用した。その概要を以下にまとめる。

Camphor3 (Kyoto Univ.)

- Architecture: x86 CPU cluster
- Interconnect: InfiniBand, Fat-tree (non-full bisection)
- Maximum nodes used: 1,024
- Memory per node: 128 GB
- Peak performance per node: 6.80 TFLOPS

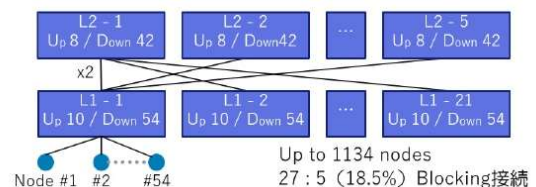


図3 システムA Fat-tree NDR 論理接続の概要

図3にシステムAにおけるインタコネクットの概要を示す。システムAのインタコネクットは2段fat-tree構成であり、第1段(L1)スイッチ1筐体あた

り 54 ノードを収容し、これが 21 筐体構成のリーフ層を形成する。第 2 段 (L2) スイッチによりこれら 21 筐体が集約され、全体として最大 1134 ノードを接続する (27:5 (18.5%) blocking の非フルバイセクション構成)。

本 DNS コードでは、A2A 通信を伴う yx-communicator (FFT) をできる限りノード内、あるいは 1 段目スイッチ内筐体 (54 ノード) に閉じ込めることが最も効率的である。しかし本ベンチマーク環境ではユーザー側でノード配置を指定できなかったため、FFT を担う yx-communicator はノード内に割り当て、TDMA を担う zy-communicator はノード間通信として扱う構成を採用した。

先述の通り、FFT は A2A を伴うがノード内通信であり高速化が期待できる。

一方、非フルバイセクション構成のノード間通信となる TDMA は隣接シフト通信のみで通信負荷が小さく、ネットワークの blocking の影響を最小限に抑えられると考えられる。

5 ベンチマーク条件

表 1 ベンチマーク条件

Re_τ	$N_x(\Delta x^+)$	$N_y(\Delta y^+)$	$N_z(\Delta z^+)$	$M[\text{TB}]$
8000	7200 (17.8)	6912 (7.4)	4096 (0.6-8.0)	24

表 1 に示す $Re_\tau \approx 8000$ の高レイノルズ数条件を対象とした。ここに $N_x(\Delta x^+)$, $N_y(\Delta y^+)$, $N_z(\Delta z^+)$ は x, y, z 方向への格子数(動粘性係数と摩擦速度で無次元化した格子分解能)を示す。また主流およびスパン方向への計算領域は $16h$ および $6.4h$ としている。

本格子数条件に必要な総メモリ量 $M \approx 24 \text{ TB}$ を満たすためには、システム A (ノード当たり 128 GB) では最低 256 ノードが必要となる。

本研究では、ノード内を 28 MPI $\times$ 4 OpenMP とするハイブリッド並列構成を採用し、yx-communicator (FFT) をノード内に割り当てた。一方、zy-communicator (TDMA) についてはノード間通信として扱い、256, 512, 1024 MPI による 256–1024 ノードの強スケーリングを評価した。

先述の通り、FFT は A2A 通信を伴うためノード内に閉じ込めることで高速化が期待できる。TDMA は隣接シフト通信のみで通信負荷が小さく、非フ

ルバイセクション構成のノード間通信であっても blocking の影響を最小限に抑えられる。

6 ベンチマーク結果

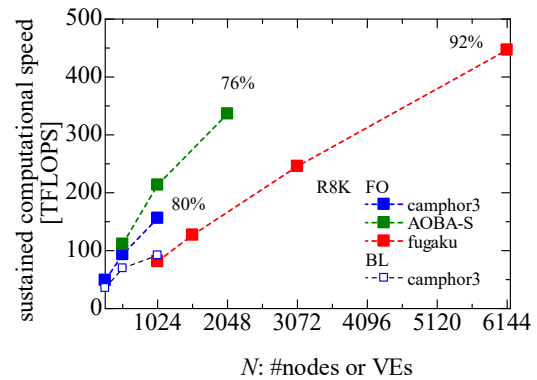


図 4 ベンチマーク結果(演算速度:Tera Floating Operation Per Second, TFLOPS)

図 4 にベンチマークで得られた実行演算速度 (TFLOPS) を示す。BL は従来手法、FO が本開発手法を意味する。また参考のため、理化学研究所・富岳および東北大学サイバーサイエンスセンタ・AOBA-S での評価結果も併記している。

システム A においては、FO (■) の実行演算速度として 1024 ノード時に 156TFLOPS(理論性能の 2.2%) を得ている。システム A 全系(1134 ノード)での HPCG ベンチマークは約 163TFLOPS であるため、本 DNS コードはこれと同等の演算速度が得られていることを意味する。また 256 ノードとの比較による並列化効率は 80% と良好な並列効果が得られている。従来手法 BL (□) の場合は、ノード数が増えると A2A コストが増加し並列効率が落ち、1024 ノード時は 92TFLOPS (FO の 58% にとどまる)

同一ノード (VE) 数で比較すると、システム A (6.8TFLOPS/ノード) は AOBA-S (4.9TFLOPS/VE) には劣るものの、富岳 (3TFLOPS/ノード) よりも高速演算が可能である。今回システム A ではノード配置の最適化が未達のため、これを行えばさらなる高速化も期待できると思われる。

7 まとめ

高レイノルズ数乱流 DNS は莫大なメモリ量を必要とし、従来は世界最高性能の計算機システムで

のみ実行が可能であった。近年のエクサスケールシステムでは演算性能の向上に比して通信性能の伸びが極めて限定的であり、通信負荷の高いスペクトル法 DNS においては対象レイノルズ数の増加が困難になりつつあった。

本研究では、スペクトル法の高精度性を維持しつつ、通信コストを大幅に低減可能な DNS コードを新たに開発した。

開発コードをシステム A の 1024 ノード上で性能評価した結果、HPCG と同等の実効演算速度を達成し、256 ノードとの強スケーリングにおいて 80% の並列化効率を確認した。本コードの適用により、高レイノルズ数 DNS における長年の課題であったレイノルズ数規模の拡大が飛躍的に進展することが期待される。

謝辞：本研究はスーパーコンピュータ共同研究制度（大規模計算支援枠）を利用して実施した。記して謝意を表する。

参考文献

- [1] J. Kim, P. Moin and R. Moser, "Turbulence statistics in fully developed channel flow at low Reynolds number," *J. Fluid Mech.*, vol. 177, pp. 133-166, 1987.
- [2] Y. Yamamoto and T. Kunugi, "MHD effects on turbulent dissipation process in channel flows with an imposed wall-normal magnetic field," *Fusion Eng. Des.*, vol. 109, no. 111, pp. 1137-1142, 2016.
- [3] Y. Kaneda and Y. Yamamoto, "Velocity gradient statistics in turbulent shear flow : an extension of Kolmogorov's local equilibrium," *J. Fluid Mech.*, vol. 929, p. A13, 2021.
- [4] L. H. Thomas, "Elliptic problems in linear difference equations over a network," Watson Sci. Comput. Lab. Rept., Columbia University, New York, 1949.
- [5] N. Mattor, T. J. Williams and D. W. Hewett, "Algorithm for solving tridiagonal matrix problems in parallel," *Parallel Computing*, vol. 21, no. 11, pp. 1769-1782, 1995.
- [6] K. H. Kim, J. H. Kang, X. Pan and J. I. Choi, "PaScaL_TDMA: A library of parallel and scalable solvers for massive tridiagonal systems," *Computer Physics Communications*, vol. 260, p. 107722, 2021.