

等方乱流シミュレーションコードの性能評価

横川 三津夫 蕪木 英雄
日本原子力研究所

近年スーパーコンピュータの発展とともに、ナビエ・ストークス方程式の直接数値シミュレーション (DNS) が行われている。しかし、高レイノルズ数の乱流現象シミュレーションには多大な計算時間とメモリが必要となるため、流れ場の初期条件として対称性を考慮した数値シミュレーションが行われている。

本報告では、初期条件として対称性を仮定しない等方乱流数値シミュレーションを行うために作成したスペクトル法によるプログラムを、スーパーコンピュータ VP2600/10 上で性能評価した結果について述べる。コードの計算時間の大部分を占める 3 次元高速フーリエ変換では、項数が小さいところで、ベクトル計算時間が $N^2 \log_2 N$ に比例することが分かった。また、項数 128^3 の乱流シミュレーションコードをベクトル計算した時に、約 12.2 倍の対スカラ速度向上比が得られた。

Performance Evaluation of a Numerical Simulation Code on Isotropic Turbulence

Mitsuo Yokokawa and Hideo Kaburaki

Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki 319-11, Japan

With the advent of supercomputers, direct numerical simulations of the Navier-Stokes equations for higher Reynolds number flows have been carried out. However, a large amount of CPU time and memories are required for the simulation. Therefore, numerical simulations are performed by assuming the symmetry of the initial flow fields.

In this report, results of performance evaluation of a simulation code, which is used for full spectral simulations of isotropic turbulence, have been presented. It is found that the execution time of a three dimensional fast Fourier transform is proportional to $N^2 \log_2 N$ in vector execution on the supercomputer VP2600/10. In a calculation of the code with 128^3 Fourier modes, a speedup of 12.2 times in vector execution is achieved compared with scalar execution.

1 はじめに

乱流は、大小さまざまな渦が不規則に入り混った極めて複雑な流体の運動である。近年スーパーコンピュータのめざましい発展とともに、ナビエ・ストークス方程式の直接シミュレーションが可能となり、乱流を数値シミュレーションによって解明する研究が進展している。

乱流を解明する上では、大きな渦のエネルギーが、どのように小さな渦へ伝達するかを知ることが重要であり、高波数領域におけるエネルギースペクトルの形状は、大域的な流れの振舞いにはあまり影響されず、等方的で一様な流れになっていることが期待される。この仮定の下では、周期境界条件を課した立方領域内で流れをシミュレーションすれば十分であり、スペクトル法が容易に適用できる。

しかし、乱流の微細構造においては、乱流場の間欠的構造などがあるため、コルモゴロフの仮定が満たされず、より高レイノルズ数領域の数値シミュレーションによる乱流構造の解明が必要である。高レイノルズ数の乱流シミュレーションでは、流れ場の初期条件として対称性を仮定して、高レイノルズ数乱流を達成する研究が報告されている [1]。

本報告では、初期条件として流れ場の対称性を仮定しない等方乱流を数値シミュレーションするために作成したスペクトル法のプログラムを、スーパーコンピュータ上で性能評価した結果について述べる。

2 基礎方程式の離散化

3次元立方領域 $\Omega = [0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$ を考える。流体の運動は、次のナビエ・ストークス方程式と連続の式で表される。

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

ここで、 $\mathbf{u} = (u_1, u_2, u_3)$ は速度である。速度場の初期条件は、周期境界条件を満足しているものとする。

式(1)を渦度 $\boldsymbol{\omega} = \text{rot} \mathbf{u} = (\omega_1, \omega_2, \omega_3)$ を用いて表現し、さらに速度、渦度ともに3次元フーリエ展開を行うと、フーリエ係数に関する次の発展方程式が得られる [1]。

$$\frac{d}{dt} \tilde{\omega}_j(\mathbf{k}) = \epsilon_{jkl} k_k k_m \tilde{u}_l \tilde{u}_m(\mathbf{k}) - \nu k^2 \tilde{\omega}_j(\mathbf{k}), \quad j = 1, 2, 3. \quad (3)$$

$$k_j \tilde{u}_j(\mathbf{k}) = 0. \quad (4)$$

$$\tilde{\omega}_j(\mathbf{k}) = -\epsilon_{jkl} k_k \tilde{u}_l(\mathbf{k}) \quad (5)$$

ここで、 ϵ_{ijk} は交代テンソル、 $\mathbf{k} = (k_1, k_2, k_3)$ である。また、 $\tilde{u}_j(\mathbf{k})$ 、 $\tilde{\omega}_j(\mathbf{k})$ はそれぞれ速度 $u_j(\mathbf{x})$ と渦度 $\omega_j(\mathbf{x})$ のフーリエ係数であり、以下のように定義する。

$$\mathbf{u}(\mathbf{x}) = i \sum_{k_1=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k_2=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k_3=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{\mathbf{u}}(\mathbf{k}) \exp[i\mathbf{k} \cdot \mathbf{x}] \quad (6)$$

$$\boldsymbol{\omega}(\mathbf{x}) = \sum_{k_1=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k_2=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k_3=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{\boldsymbol{\omega}}(\mathbf{k}) \exp[i\mathbf{k} \cdot \mathbf{x}] \quad (7)$$

また、式(3)右辺第1項、第2項は、それぞれナビエ・ストークス方程式の移流項と拡散項である。移流項の $\tilde{u}_l \tilde{u}_m(\mathbf{k})$ は、 $u_l(\mathbf{x}) u_m(\mathbf{x})$ のフーリエ係数を表す。

$$u_l(\mathbf{x}) u_m(\mathbf{x}) = \sum_{k_1=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k_2=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{k_3=-\frac{N}{2}}^{\frac{N}{2}-1} \tilde{u}_l \tilde{u}_m(\mathbf{k}) \exp[i\mathbf{k} \cdot \mathbf{x}] \quad (8)$$

以下では、 \sim でフーリエ係数を表す。

式(3)の時間積分には4段4次のルンゲ-クッタ法を用いた。

3 数値アルゴリズム

等方乱流シミュレーションコードの計算の流れは、以下の通りである。

1. 時間ステップ Δt , 動粘度 ν , フーリエモード数 N を与える。
2. 実空間の初期速度場をフーリエ変換し, (5)式を用いて初期渦度場のフーリエ係数 $\tilde{\omega}$ を求める。
3. ルンゲ-クッタ法による時間発展のループ

- (a) $\tilde{\omega}_{old} = \tilde{\omega}$.
- (b) $\tau = RHS(\tilde{\omega}), \quad \tilde{\omega} = \tilde{\omega}_{old} + \frac{\Delta t}{6}\tau, \quad \omega' = \tilde{\omega}_{old} + \frac{\Delta t}{2}\tau.$
- (c) $\tau = RHS(\omega'), \quad \tilde{\omega} = \tilde{\omega} + \frac{\Delta t}{3}\tau, \quad \omega' = \tilde{\omega}_{old} + \frac{\Delta t}{2}\tau.$
- (d) $\tau = RHS(\omega'), \quad \tilde{\omega} = \tilde{\omega} + \frac{\Delta t}{3}\tau, \quad \omega' = \tilde{\omega}_{old} + \Delta t\tau.$
- (e) $\tau = RHS(\omega'), \quad \tilde{\omega} = \tilde{\omega} + \frac{\Delta t}{6}\tau.$

ここで, $RHS(\tilde{\omega})$ は, 式(3)の右辺の計算をする部分であり, この中で最も時間がかかる部分は, 非線形項 $\widehat{u_1 u_1}, \widehat{u_1 u_2}, \widehat{u_1 u_3}, \widehat{u_2 u_2}, \widehat{u_2 u_3}, \widehat{u_3 u_3}$ の計算部分である. これを直接にフーリエ係数を用いて計算すると3次元で計算量が $O(N^4)$ になる [2].

ところが, この畳み込み積分について, フーリエ係数を実空間に変換してから, 実空間でそれぞれの積を計算し, またフーリエ空間に戻すことを考える. こうすれば, 高速フーリエ変換 (FFT) を用いることができ, 計算量が $O(N^3 \log_2 N)$ となるため, フーリエ空間で畳み込みを行うよりも計算時間が節約できる. このような方法は, 擬スペクトル法と呼ばれる [2].

ただし, 擬スペクトル法では aliasing error が生じるので, 3/2 則と呼ばれる方法で aliasing error を除く必要がある. いま, 高速フーリエ変換する項数 $N (= 2^l)$ を, 有効モード数 M を用いて $N = 3M/2$ とする. ここで, 有効モード数とは, 実際に(3)式を計算する項数である. 3/2 則では, 有効モード数よりも大きいモードに対して, 式(11)のように0を代入してフーリエ変換を行う [3].

$$x_j = 2\pi j/N \quad (9)$$

$$u_j = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} \hat{u}_k e^{ikx_j} \quad (10)$$

$$\hat{u}_k = \begin{cases} \tilde{u}_k & -M/2 \leq k \leq M/2 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

擬スペクトル法を用いた非線形項の計算では, まず, フーリエ空間の速度3成分を物理空間の速度に逆フーリエ変換するために3次元FFTが3回用いられる. 次に, 物理空間で $u_1 u_1, u_1 u_2, u_1 u_3, u_2 u_2, u_2 u_3, u_3 u_3$ の各項の積を求める. 最後に, これらの積のフーリエ係数を求めるために, 3次元FFTが6回必要となる. したがって, 非線形項の計算には, 計9回の3次元FFTルーチンが呼ばれる.

1回の時間ステップでは, ルンゲ-クッタ法の各ステージで非線形項の計算が行われるので, 36回の3次元FFTが実行される. 1回の3次元FFTでは, 3次元の各方向にすべて1次元FFTを行うので, 1次元FFTが $3N^2$ 回用いられる. したがって, 全体では1次元FFTが $108N^2$ 回実行される. $N=64$ の場合では, 1時間ステップ内で1次元FFTが442368回実行される.

本コードはFortranで書かれており, コード全体の実数変数は $17N^3$ 個用いた. これは, 単精度変数を用いるとして, $N=128$ で約14MB, $N=512$ で約9GBの主記憶を必要とする. ちなみに, $N=128$ の時の有効モード数は85となる.

4 数値実験結果

前節のアルゴリズムで、本コードの実行時間の大部分は3次元FFTであることが分かる。実際、 $N = 32$ と $N=64$ の場合に、実行時間の中で3次元FFTの占める割合は、それぞれ約82.5%、87.2%である。したがって、3次元FFTの評価を行っておけば、コード全体の計算時間が予測できる。

以下では、1次元及び3次元FFTの評価結果と、シミュレーションコード全体の評価結果について述べる。評価に用いた主な計算機は、Fujitsu VP2600/10である。また、以下では単精度計算のみの結果を示している。

4.1 高速フーリエ変換の評価

1次元FFTをベクトル計算機で高速に計算するためには、リストベクトルを用いたIsogeometric型(Peaceのアルゴリズム)又は、Self-Sorting型の方法が用いられている。Isogeometric型のFFTは、変換の全ステージを通してデータの入力及び出力転送パターンがすべて同一であるが、一番最後のステージではビット逆順になる。Self-Sorting型は、全ステージを通して入力の転送パターンが同じであるが、出力パターンは各ステージで規則的に変化するものである。

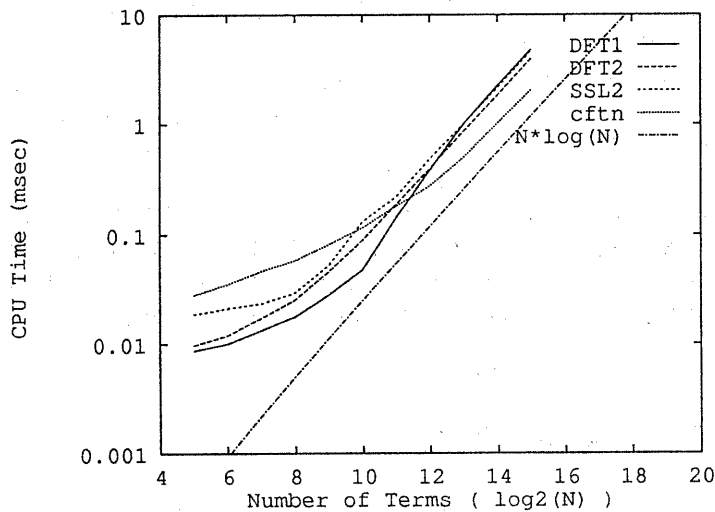


図 1. 1次元FFTの実行時間

今回評価に当たって、Isogeometric型のルーチンDFT1、Self-Sorting型のルーチンDFT2を作成し、科学技術計算パッケージSSL2のFFTルーチンVCFT1[4]、小野寺によるFFTルーチンCFTN[5]を若干修正したものと比較した。

一般に、ベンダが提供しているFFTルーチンは、引数のチェックなどのオーバーヘッドが含まれる。本シミュレーションのように、FFTが1時間ステップでかなりの回数呼び出されるような場合には、このオーバーヘッドは無視できない。本シミュレーションでは、変換の項数は前もって分かっており、引数をチェックする必要がないので、自前のルーチンの方がオーバーヘッドが削減できるはずである。一方、CFTNでは、各ステージでメモリアクセスがなるべく連続となるようにコーディングを行っている。

図1に1次元FFTの1回の変換に要するベクトル実行時間を示す。この図から分かるように、項数 N が小さい所ではリストベクトルを用いたFFTが実行時間が短い、 N が大きいの所ではリストベクトル

```

dimension xr(np,n,n), xi(np,n,n), wr(n), wi(n)
C-----
do k = 1, n                                x方向の1次元FFT
  do j = 1, n
    call FFT1D( xr(1,j,k), xi(1,j,k), ... )
C-----
do k = 1, n                                y方向の1次元FFT
  do i = 1, n
    do j = 1, n
      { wr(j) = xr(i,j,k); wi(j) = xi(i,j,k) }
      call FFT1D( wr, wi, ... )
    do j = 1, n
      { xr(i,j,k) = wr(j); xi(i,j,k) = wi(j) }
C-----
do j = 1, n                                z方向の1次元FFT
  do i = 1, n
    do k = 1, n
      { wr(k) = xr(i,j,k); wi(k) = xi(i,j,k) }
      call FFT1D( wr, wi, ... )
    do k = 1, n
      { xr(i,j,k) = wr(k); xi(i,j,k) = wi(k) }
C-----

```

図 2. 3次元FFTの計算フロー概要

ルを用いない CFTN が最も実行が早い。また、 N が大きい所ではパイプラインが飽和して、計算時間が $N \log_2 N$ に比例することが分かる。

しかし、CPU 時間やメモリの観点から、本シミュレーションの可能な範囲はせいぜい $N=256$ 程度なので、この範囲で最も実行時間が短い DFT1 (Isogeometric 型) を、以下のシミュレーションで用いることにした。

つぎに、本シミュレーションに必要な 3次元FFT の評価結果について述べる。3次元FFT では、 x 方向、 y 方向、 z 方向の各空間方向に対して、1次元FFT を適用する。3次元FFT プログラムの概要を図2に示す。ここで、 xr 、 xi はそれぞれ実数用と虚数用の 3次元配列である。また、 wr 、 wi は作業用の 1次元配列である。

y 方向、 z 方向の 1次元FFT では、それぞれ 3次元配列の第 2 番目と第 3 番目の添字についてアクセスしている。このため、第 1 番目の配列要素数によっては、メモリバンク・コンフリクトが生じる。特に 1次元FFT は、 n が 2 の巾乗に取られるため、この問題は大きい。そこで、第 1 番目の添字 np を n 、 $n+1$ 、 $n+2$ とした時に、3次元FFT の変換・逆変換する実行時間を計測した。表 1 に、計測結果を示す。表において、 n は項数、 np は第 1 添字の大きさ、Scalar はスカラ実行時間、Vector はベクトル実行時間、Speedup は対スカラ速度向上比である。この表から、メモリバンク・コンフリクトの影響は非常に大きいことが分かる。 $n=128$ の時、 $np=128$ と $np=129$ のベクトル実行時間を比較すると、1.5 倍の差があることが分かった。

また、 $n+1$ のケースについて、項数とスカラ実行時間、ベクトル実行時間の関係を図 3 に示す。この

図から明らかに、スカラ実行時間は3次元FFTの演算量 $N^3 \log_2 N$ に比例しているが、ベクトル実行時間は、 $N^2 \log_2 N$ に比例することが分かった。

表 1. 3次元FFTにおけるバンクコンフリクトの影響

n	np	Scalar (msec)	Vector (msec)	Speedup
16	16	23.46	13.66	1.72
	17	23.57	12.84	1.84
	18	23.52	12.97	1.81
32	32	244.29	75.38	3.24
	33	212.92	62.68	3.40
	34	213.10	62.36	3.42
64	64	2102.37	400.59	5.25
	35	1803.45	298.78	6.04
	36	1861.38	307.41	6.06
128	128	25745.00	2428.52	10.60
	129	19349.80	1610.54	12.01
	130	21868.80	1796.80	12.17

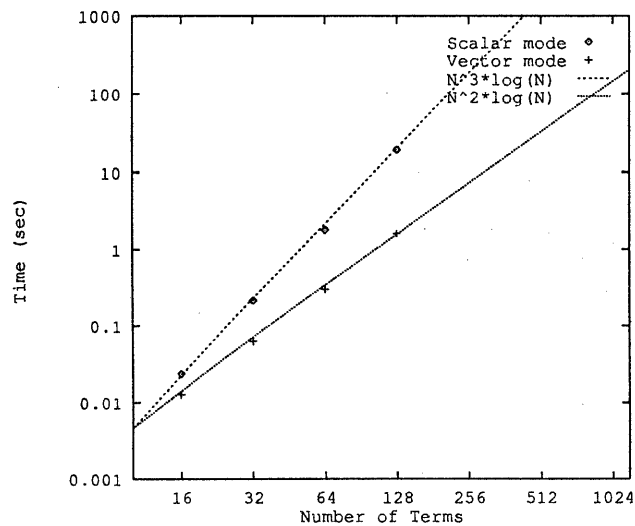


図 3. 3次元FFTの実行時間 (np = n+1)

4.2 等方乱流シミュレーション

Isogeometric 型の 1 次元 FFT, $np=n+1$ とした 3 次元配列を用いて作成したシミュレーションコードの計算精度を調べるために, 時間ステップを 0.04, 0.02, 0.01, 0.005 と変化させて, 流体の単位質量あたりの運動エネルギーを計算した. この時の動粘度は $\nu = 0.001$, 項数は $N=64$ である. この結果, ルンゲ-クッタ法の収束範囲内では良い一致が見られ, 時間ステップの大きさは結果に影響を与えないことが分かった. また, 倍精度演算結果との比較も行ったが, 運動エネルギーの値に差は見られなかった.

図 4 に $N=128$, 動粘度 $\nu = 0.001$, 時間ステップ 0.005 とした時の $t=4, t=5, t=6, t=7$ におけるエネルギースペクトルを示す. 横軸は波数成分, 縦軸はエネルギースペクトルである. 波数は, 渦の大きさに対応していると言える. この図からほぼ $-5/3$ 乗則が再現されていることが分かる.

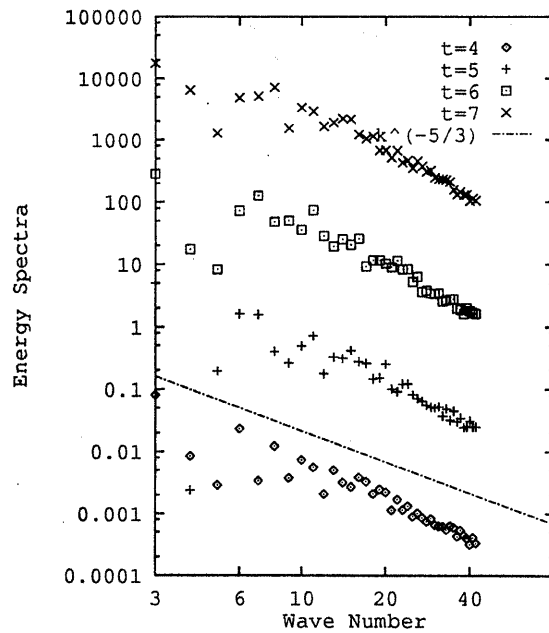


図 4. 3 次元エネルギースペクトルの時間変化

表 2. ベクトル計算による速度向上率

N	NP	Scalar (sec)	Vector (sec)	Speedup
32	33	4.18	1.16	3.60
64	65	35.38	5.46	6.48
128	129	371.12	30.31	12.24

乱流シミュレーションコードの実行時間は, 3 次元 FFT の評価によって予測が可能となった. 乱流シミュレーションを 1 時間ステップ進めるのに必要な実行時間の計測結果を, 表 2 に示す. 各項数の実

行時間は、1回あたりの3次元FFT実行時間の約36倍であり、速度向上率は3次元FFTの速度向上率とほぼ一致している。項数128のシミュレーションでは、約12.2倍の対スカラ速度向上率が得られた。

5 並列計算に向けて

本シミュレーションの様に時間発展のシミュレーションを並列計算する場合には、各時間ステップで計算の同期を取る必要がある。また、ルンゲ-クッタ法では、各ステージで前ステージの結果が必要となるから、各ステージでも同期が必要である。したがって、さらに計算粒度の小さい所での並列処理をしなければならない。

図2で示した3次元FFTのコーディングでは、外側のループに並列性が見られる。配列データについても、 x 方向、 y 方向の1次元FFTの部分では、 z 方向に分割可能である。しかし、 z 方向の1次元FFTの部分では、データを局所化出来ない。

分散メモリ型の並列計算機では、 x 方向、 y 方向の1次元FFTについて並列化を行うとすれば、3次元配列を第3番目の添字について分割して並列化できる。しかし、 z 方向については、データをプロセッサ間転送する必要が生じるので、この評価をする必要がある。

共有メモリ型の並列計算機でも同様の方法で並列化を行うとすれば、やはり最後の z 方向のデータアクセスで、どのくらいのオーバーヘッドがあるか検討する必要がある。

6 まとめ

本報告では、等方乱流シミュレーションコードをスーパーコンピュータVP2600/10上で評価した結果について示した。コードの計算時間の大部分を占める3次元高速フーリエ変換では、項数が小さい所で、ベクトル計算時間が $N^2 \log_2 N$ に比例することが分かった。また、項数128³の乱流シミュレーションをベクトル計算した時に、約12.2倍の対スカラ速度向上比が得られた。

参考文献

- [1] S. Kida and Y. Murakami, "Statistics of velocity gradients in turbulence at moderate Reynolds numbers," fluid Dynamics Research, Vol. 4 (1989).
- [2] S. A. Orszag, "Numerical Simulation of Incompressible Flows Within Simple Boundaries. I. Galerkin (Spectral) Representations," Studies in Applied Mathematics, Vol. L, No. 4, December (1971).
- [3] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag (1988).
- [4] 富士通マニュアル, SSLII 拡張機能仕様手引書 (1985).
- [5] 渡部 力, 名取 亮, 小国 力, Fortran77 による数値計算ソフトウェア, 丸善 (1989).