

## 領域分割法による航空機形状まわりの遷音速流の並列計算

伊藤 良三\* 高梨 進\*\*

Parallel Computation of Transonic Flows  
About an Aircraft Configuration Using Multi-Block Structured Grids

by

Ryozo ITO\* Susumu TAKANASHI\*\*

## ABSTRACT

Three dimensional Navier-Stokes computation of transonic flows about ONBRA-M5 has been carried out using the NAL's parallel vector computer called "NWT". The computational space around ONBRA-M5 is decomposed into 14 blocks. The code is parallelized in two methods. The first one is sharing the computation for each block to each PE. The second one is combining the first one and the parallelization in each block. The performance of the parallelization is estimated by measuring the cpu-time for solving the Navier-Stokes equations.

Validation of the computational results has also been done.

## 1. はじめに

NALに1993年2月に設置された分散主記憶型並列計算機(NWT)は、140台のPE(要素計算機)を持ち、各々のPEのピーク性能は約1.7GigaFLOPSである。筆者らはVP400用に書かれたマルチ・ブロックCFDコード<sup>1)</sup>をNWT用に並列化し、その計算に要するCPU時間を大幅に短縮できることを確認した。

## 2. CFDコード

コードはReynolds平均薄層近似Navier-Stokes方程式を有限体積法で解くものである。対流項の計算には三次精度のTVDスキームを適用している。時間積分においては局所時間刻みを用い、定常解を求めている。以下で扱っている計算ではいずれも流れ場に対称条件を課して、片側の流れ場だけを解いている。また、乱流モデルとして、Baldwin-Lomax<sup>2)</sup>のものを用いている。詳細は参考文献<sup>1)</sup>を参照のこと。コードの概略的な流れ図を図1に示す。この図で、"PARALLEL REGION"と記された範囲の処理を並列化した。

## 3. 計算格子

計算領域をブロック状に分割する。三次元空間を一般曲線座標で表し、一様流方向を $x$ 、これに直交する方向を $y$ 、物体面かwake面から離れる方向を $z$ とする。各ブロックの底面( $z=0$ )は、物体の表面か、wake面と一致し、その反対側の面は、遠方境界面と一致している。隣合うブロックの境界面上の格子点の座標は互いに完全に一致している。

## 4. 並列化の方法

基本的には、一つのブロック内のCFD計算を一台のPEで行う。即ちブロック内の計算は従来の逐次型計算機によるものと同じである。具体的には、配列の次元のうち、ブロック番号に対応する次元を分割することによって並列化する。例えば、物理量 $Q(I, J, K, 5, IB)$ のうちの最後の添字がこれに相当する。各PEの計算負荷を等しくすれば並列化の効率を高めることができる。これは、各ブロックに含まれるセルの数を揃えることに相当する。負荷を調整するために複数のブロックを一台のPEに割り当てることもある。このコードでは、あるブロックの境界部分の物理量の値を計算するために、隣接したブロック内の値が必要であり、もしこれらの二つのブロックの計算が別々のPEに割り当てられているときは、図2に示す様にPE間のデータ転送が発生する。PE間のデータ転送は並列化に特有の処理なのでこの処理を最小限に食い止めることが重要である。尚、この並列化の改修により、元のプログラムと比較して計算に要する主記憶の量は約12%増加した。

## 5. 計算結果

## 1) 球の周りの流れの計算

完全に各PEの計算負荷が揃っているときに、並列化による計算の高速化がどの程度達成されるかを評価するために行った。球の様に単純な形状の周りの流れを計算する場合、わざわざマルチ・ブロック格子を用いる必要はない。格子のトポロジーはO-O型のため

\* 大興電子通信株式会社 DAIKO Ltd.

\*\*航空宇宙技術研究所 National Aerospace Laboratory

wakeから立ち上がるブロックは無く、ブロック数は16で、各ブロック内のセル数(ξ\*η\*ζ)は16\*16\*64で総セル数は約26万。図3に表面格子とブロックの配置を示す。これをPE台数1、2、4、8、16で実行し、CFDの繰り返し計算(三次精度TVD, 1000回)にかかるCPU時間を測定したところ、表1、図4のようになった。CPU時間は、図5の様に関数CLOCKVを繰り返し計算の始まりと終わりの二箇所に挿入したときの、二箇所の時間差とした。これを以下「最長-CPU時間」と呼ぶ。PEが16台のときの最長-CPU時間は、PEが1台のときの1/15.4になった。1台のPEで計算したときの最長-CPU時間をT<sub>1</sub>とし、同じ計算をN台のPEで並列計算したときの最長-CPU時間をT<sub>N</sub>とし(T<sub>1</sub>/T<sub>N</sub>)/Nを、N台のPEでの台数効果とすると、PE台数16個のときの台数効果は約96%になる。PE台数16台のときを例にして、台数効果が100%にならない原因を考えてみる。まず、4.の並列化の方法のところでもふれた様に、各ブロックの流れ場を別々のPEで計算するために、境界条件の値を他のPEから転送してやらなければならない。これに要するCPU時間は1000回当たり2.5秒で、逐次計算のときと比べてこの分だけ確実に遅くなる。また、図6に示す方法で、各PE内でCFDの計算のみに要するCPU時間を測定して、先に求めた最長-CPU時間の内、CFD計算でもPE間転送でもない時間を求めたところ、各PEで約6秒であった。これは最長-CPU時間の約3%に相当する。この時間は、SPRBD DOにかかる時間、PEの同期にかかる時間の合計と考えられ、逐次計算のときはかからない時間である。これらの時間を差し引くと、CFD計算のみに要する時間は約176秒になる。

## 2) ONERA-M5周りの遷音速流の計算 (CASE-1, 2)

計算格子は参考文献2)の「CFDワークショップ」で用いたものと同じで、セルの総数は約56万である(図7)。流れ場の条件は、一様流マッハ数0.84、迎角-1.0°、Reynolds数10<sup>6</sup>である。以下の二通りのブロックの切り方で結果を比較した。CASE-1: ブロック数14(図8-a)で、各ブロックがPEと1:1に対応している場合、CASE-2: 並列化の台数効果を上げるためにブロック数を23個にして各PEが担当するブロックのセル数を揃えた場合(図8-b)。使用したPE台数はそれぞれ14、16である。各ブロックのセル数、また、ブロックとPEの対応関係は表2、表3の通りである。CASE-2ではブロックとPEが1:1に対応していない。並列計算では、各PEの同期をとるため、最長-CPU時間は最も計算量の多いPEで決定される。

計算量はほぼ各PEの担当するブロックに含まれるセル数に比例すると考えられる。計算量が最大になると推定されるPEは、CASE-1とCASE-2とでそれぞれPE-4とPE-3であり、そのセル数の比は約3.25:1である(表2, 3)。即ち、CASE-2の計算は、CASE-1よりも3.25倍高速化される筈である。しかし、表4でわかる様に実際には2.63倍しか速くなっていない。この原因を調べるために、球のときと同様に、CFD計算のみに要する時間、PE間データ転送に要する時間のいずれでもない時間を測定してみると、CASE-1では最長-CPU時間の20%、CASE-2では30%に達していることがわかった。球の計算では3%に過ぎなかったことを考えると、各PEの計算量が揃っていないときや、ブロックとPEが1:1対応していないときは並列計算のためのオーバーヘッドが増加してしまうものと思われる。並列化の台数効果をあげるには、各PEの計算量を揃えることと、PEとブロックの関係を1:1にすることが重要である。

## 3) ブロック内を更に並列化した場合(CASE-3)

上の2)で用いた方法では、並列化の効果を上げるためには分割するブロックの数を増やさなければならないが、当然のことながら、ブロック境界では計算精度が落ちることが予想されるので、あまり細かく分割することは好ましくない。また、格子生成作業との兼ね合いから、ブロック内のセル数をあまり揃えることができないこともありうる。そこで、ブロック内の計算に現れるDO-LOOP(例えばξ方向の対流項の計算の中のどのDO-LOOP)を一つ一つ並列化する方法を試みた(図9)。並列化の制御行(IXOCLで始まる行)については文献3)を参照されたい。尚、並列化し易くするためにηのDO-LOOPとどのDO-LOOPの入れ替えを行ったことにより計算の順序が変わっているところもあるが、理論上の精度には影響を与えない範囲に留めた。この方法では、ブロックの分割数を増やすときの様なブロック境界の計算精度への悪影響を避けることができる反面、ある方向に分割された配列の値を別の方向に分割された配列の領域に転送する処理(例えばη→ξ、図10)が頻繁に発生するため、使うPE台数の割りには計算は高速化されない。また、一種類の配列を二種類の添字で分割するため、CASE-1, 2の方法よりも約2倍の主記憶を要する。ここでは、CASE-1の計算格子の各ブロックの内部のDO-LOOPを、ηあるいはξの方向に6分割して並列計算してみた。使用したPEの台数は14x6=84台である。最長-CPU時間を表7に示す。CASE-1に比べて4.7倍速くなっている。6倍速くならないのは、PE間のデータ転送が増加したからである。

6. 計算結果の妥当性

CASB-1~3の計算結果についてその妥当性を確認した。繰り返し回数 25000回までの計算結果に基づく主翼表面のCp分布を図10に示す。S1、S2、S3断面の分布は三つの計算ともに、VP2600での計算結果<sup>2)</sup>と一致した。一方、翼端のS4断面では、かなり異なっている(図11)。翼端の流れは一般に剝離し易く不安定であり、定常計算と言えども解が完全に収束していないことが考えられる。更に繰り返し計算を進めたところ、どのCASBにおいても翼端のCp分布は一定の振幅で周期的に変動していることがわかった。途中の計算結果からCp分布を描いてみると、繰り返し計算の中のある時点では、CASB-2,3の計算においてもCASB-1と似た分布が出現することがわかった(図12)。CASB-1~CASB-3で、全機の空力係数の相違は約0.3%以内であった。

7. 結び

逐次処理計算機用のマルチブロックCFDコードをNALの並列ベクトル計算機NWT用に二つの方法で並列化し、計算時間が短縮されること、また、並列化が計算結果へ大きな影響を与えないことを確認した。それぞれの方法の長所・短所を挙げると以下の様になる。

① 単純にブロックをPEに対応させる方法

長所:

- ・コーディングが簡単で、改修がやり易い。
- ・主記憶の無駄が少ない。
- ・並列化の台数効果が高い。

短所:

- ・ブロックの切り方によっては解に悪影響を与えることがある。
- ・計算できる格子のセル数は各PE一台当たりの主記憶の量で決まる。

② ①に加え、ブロック内も並列化する方法

長所:

- ・ブロックの切り方をあまり工夫しなくても良い。
- ・解に悪影響を与えない。
- ・主記憶容量の不足に依り一つのPEで計算できないときはブロック内の並列度を上げれば良い。

短所:

- ・コーディングは①に比べて複雑で、改修が①に比べてやり難い。
- ・主記憶の無駄が多い。
- ・PE間のデータ転送が増えるため、並列化の台数効果は低い。

また、①、②の両方で言えることだが、ブロックとPEが1:1で対応していないときや各PEの受け持つ計算量が不均一なときは並列化のためのオーバーヘッドが増加してしまうことがわかった。更なる原因究明が必要と思われる。

参考文献

- 1) 橋、高梨: 複雑な形状の航空機まわりの流れの数値計算, NAL SP-10, 1989.
- 2) 高梨、伊藤、橋: ONBRA M5 全機形態まわりの流れ解析, NAL SP-20, 1993.
- 3) 吉田、中村、福田、岡田、中村: NWT向け並列Fortran プログラミングについて, NAL SP-19, 1992.

図1 CFDコードの流れ図

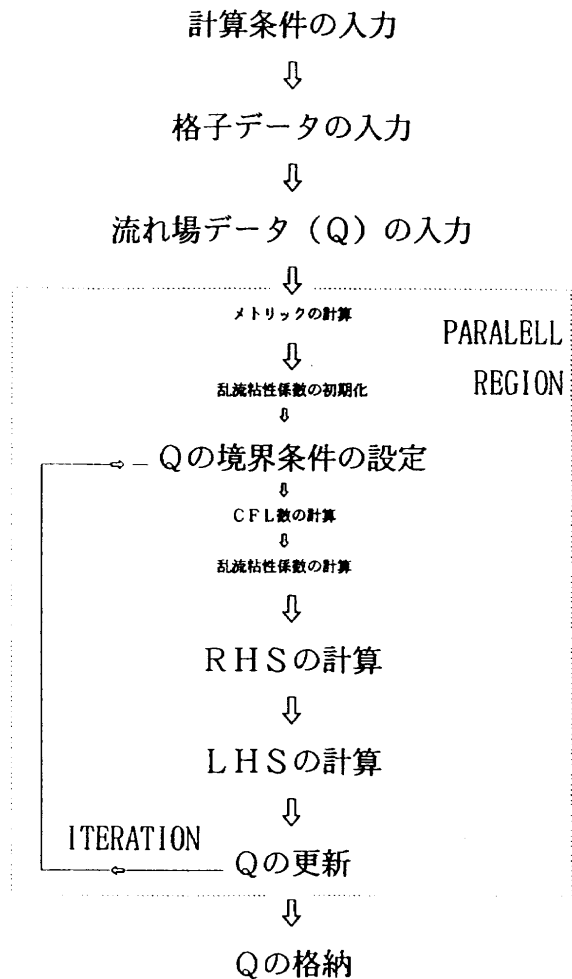
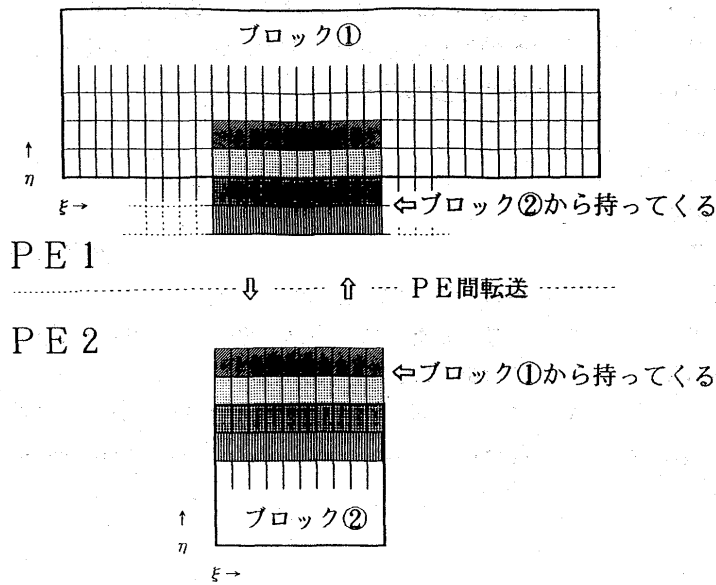


図2 PE間のデータ転送



(ξ = 一定の面を上から見たところ)

図3 球の表面格子とブロック番号

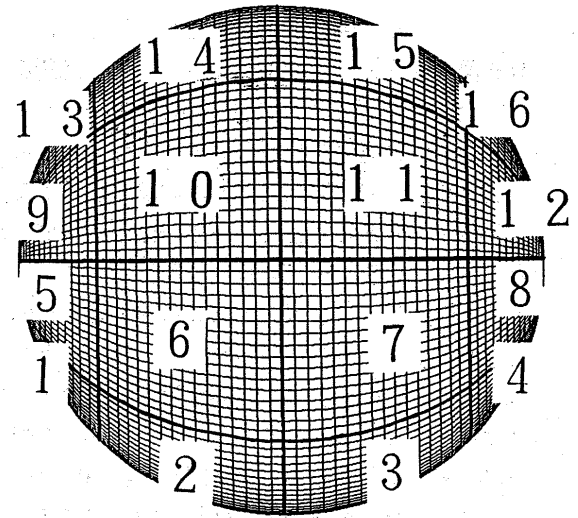
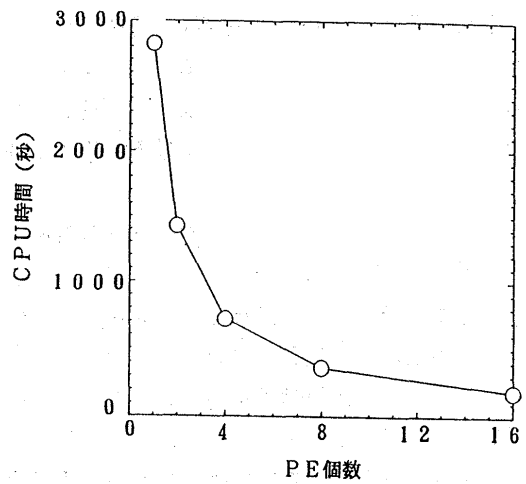


図4 PE個数とCPU時間 球



```

PROGRAM MAIN
  ファイル入力etc.
  .
!XOCL PARALLEL REGION
  .
  メトリックの計算etc.
  .
  CALL CLOCKV(VPU1, CPU1, 2, 2)
  繰り返し計算の開始
  .
  .
  RHS, LHS の計算etc.
  .
  .
  繰り返し計算の終了
  CALL CLOCKV(VPU2, CPU2, 2, 2)
  .
!XOCL END PARALLEL
  .
  ファイル出力etc.
  .
  STOP
  END
    
```

図5 繰り返し計算に要したCPU 時間の測定方法  
CPU 時間 = CPU2 - CPU1

```

SUBROUTINE SUB1
!XOCL SPREAD DO /INDP
  DO 1000 IB = 1, IBN
    CALL CLOCKV(VPU1, CPU1, 2, 2)
    .
    計算
    .
    CALL CLOCKV(VPU2, CPU2, 2, 2)
  1000 CONTINUE
!XOCL END SPREAD
  RETURN
  END
    
```

図6 各サブルーチンでの各ブロック毎のCPU 時間の測定方法  
CPU 時間 = CPU2 - CPU1

図7 ONERA-M5の表面格子 (wakeを含む)

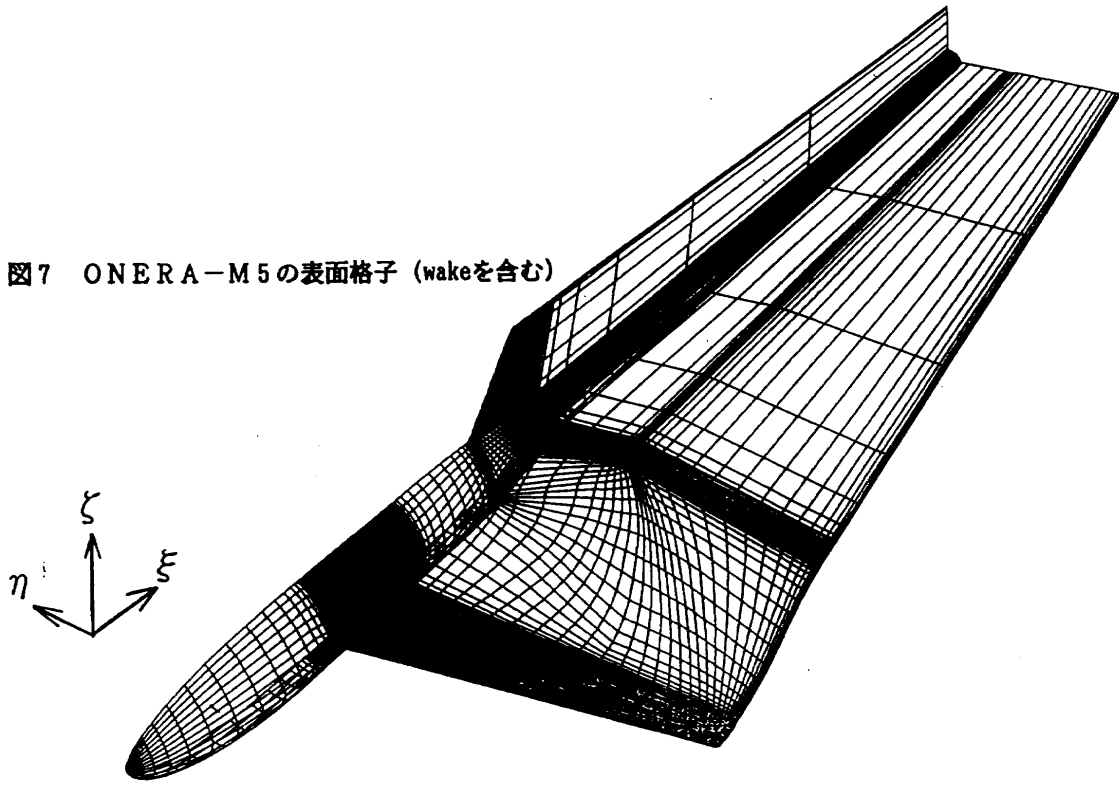
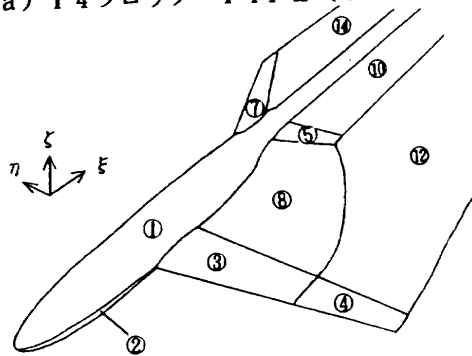
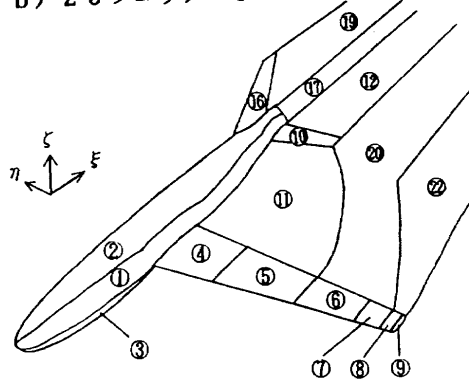


図8 ONERA-M5の格子のブロックの配置

a) 14ブロック-14PE (CASE-1)



b) 23ブロック-16PE (CASE-2)



```

SUBROUTINE SUB1
!XOCL SPREAD DO /INDP
DO 1000 IB = 1, IBN

!XOCL SPREAD DO /INDJ
DO 100 J = 1, JMAXB(IB)
DO 100 I = 1, IMAXB(IB)
DO 100 K = 1, KMAXB(IB)
.
.
計算
.
100 CONTINUE
!XOCL END SPREAD

1000 CONTINUE
!XOCL END SPREAD

RETURN
END
    
```

図9  
ブロック内のDO-LOOP を更に並列化する

```

SUBROUTINE QJTK
.
!XOCL SPREAD DO /INDP
DO 1000 IB = 1, IBN

!XOCL SPREAD NOBARRIER MOVE /INDJ
DO 100 J = -1, JMAXB(IB)+2
DO 100 N = 1, 5
DO 100 K = -1, KMAXB(IB)
DO 100 I = -1, IMAXB(IB)+2
.
.
QG(I, J, K, N, IB)=QJ(I, J, K, N, IB)
100 CONTINUE
!XOCL END SPREAD (1D)
!XOCL MOVEWAIT NOBARRIER (1D)

1000 CONTINUE
!XOCL END SPREAD

RETURN
END
    
```

図10  
Jで分割された配列QJから、Kで分割された配列QGへのデータ転送

図10 ONERA-M5主翼Cp分布 S1, S2, S3 断面

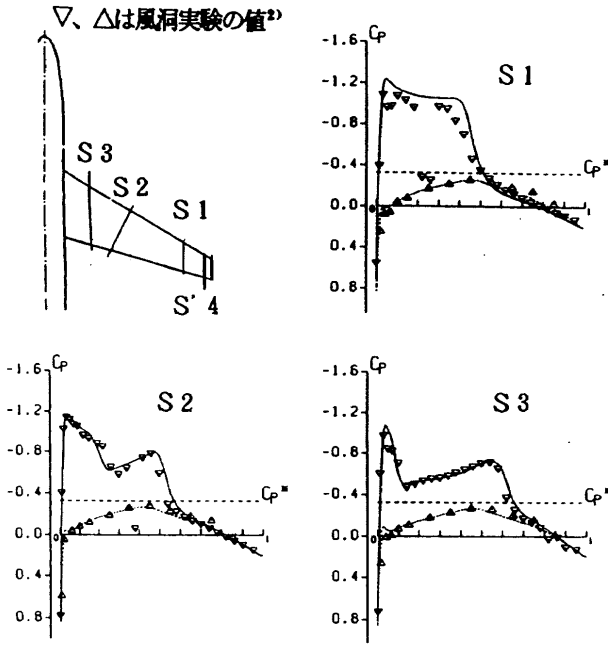


図11 ONERA-M5主翼Cp分布 S'4断面 CASB-1 CASB-2 CASB-3

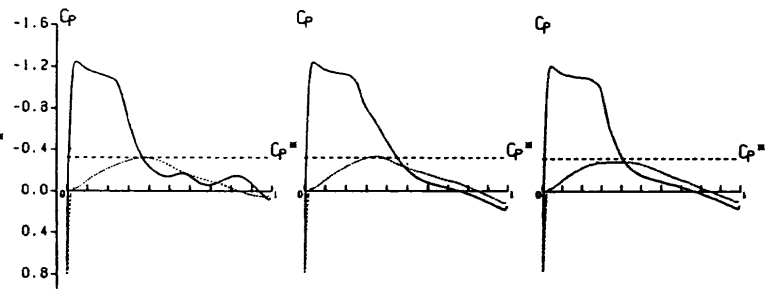


図12 ONERA-M5主翼Cp分布 S'4断面 CASB-2 (26600回) CASB-3 (26200回)

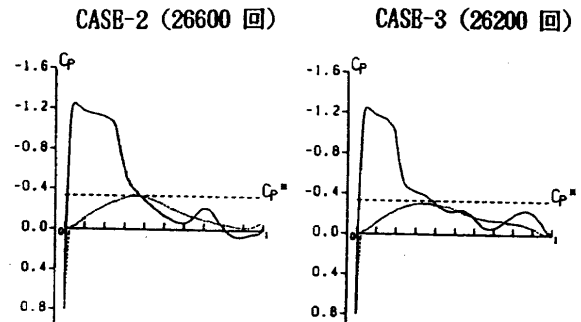


表1 PE台数とCPU時間：球の周りの流れの計算

PE台数	繰り返し計算(SBC)	PE間転送(SBC)	台数効果
1	2831.4	—	1.000
2	1429.0	1.858	0.991
4	718.7	3.703	0.985
8	362.5	2.752	0.976
16	184.2	2.495	0.961

表2 ONERA-M5のブロックとPEの対応関係 CASE-1

PE番号	ブロック番号	場所	ξ x η	セル数の比率 (%)
1	1	胴体上部(フィンガ)	160 x 10	13.9
2	2	胴体下部(フィンガ)	160 x 6	8.4
3	3	主翼(翼根側)	148 x 10	12.9
4	4	主翼(翼端側)	148 x 20	25.8
5	5	水平尾翼上面	26 x 10	2.3
6	6	水平尾翼下面	26 x 10	2.3
7	7	垂直尾翼	26 x 10	2.3
8	8*	主翼の後・内側・上	28 x 10	2.4
9	9*	主翼の後・内側・下	28 x 10	2.4
10	10*	水平尾翼の後・上	12 x 10	1.0
11	11*	水平尾翼の後・下	12 x 10	1.0
12	12*	主翼の後・外側・上	66 x 20	11.5
13	13*	主翼の後・外側・下	66 x 20	11.5
14	14*	垂直尾翼の後	26 x 10	2.3
*はwake上のブロック			11480	100.0

表3 ONERA-M5のブロックとPEの対応関係 CASE-2

PE番号	ブロック番号	場所	ξ x η	セル数の比率 (%)
1	1	胴体上部主翼側	152 x 5	6.6
2	2	胴体上部対称面側	152 x 5	6.6
3	3	胴体下部	152 x 6	7.9
4	4	主翼(翼根側)	148 x 5	6.4
5	5	主翼	148 x 5	6.4
6	6	主翼	148 x 5	6.4
7	7	主翼	148 x 5	6.4
8	8	主翼	148 x 5	6.4
9	9	主翼(翼端側)	148 x 5	6.4
10	10	水平尾翼の上面	26 x 10	5.7
11	11	水平尾翼の前・上部	28 x 10	
12	12	水平尾翼の後・上部	12 x 10	
13	13	水平尾翼の下面	26 x 10	5.7
14	14	水平尾翼の前・下部	28 x 10	
15	15	水平尾翼の後・下部	12 x 10	
16	16	垂直尾翼	26 x 10	5.6
17	17	ステイニング上部	8 x 10	
18	18	ステイニング下部	8 x 6	
19	19	垂直尾翼の後	26 x 10	5.7
20	20	主翼の後・内側・上	66 x 10	
21	21	主翼の後・内側・下	66 x 10	
22	22	主翼の後・外側・上	66 x 10	
23	23	主翼の後・外側・下	66 x 10	5.7
*はwake上のブロック			11480	100.0

表4 PE台数とCPU時間(秒)：ONERA-M5の周りの流れの計算

ブロック数-PE台数	繰り返し計算	PE間転送	計算・転送以外の時間
14 - 1	7718.5*	—	—
CASB-1 14 - 14	2249.3	6.6	441.1
CASB-2 23 - 16	856.4	6.9	275.9

\*推定値(表5の各PEのCPU時間の合計)

表5 各PEのCFD計算のみのCPU時間(秒)：ONERA-M5 CASB-1

PE番号	時間	比率	セル数の比率 (%)
1	998.2	12.9	13.9
2	611.6	7.9	8.4
3	924.0	12.0	12.9
4	1808.2	23.4	25.8
5	169.3	2.2	2.3
6	169.0	2.2	2.3
7	169.4	2.2	2.3
8	217.2	2.8	2.4
9	217.2	2.8	2.4
10	88.8	1.3	1.0
11	87.6	1.3	1.0
12	977.8	12.7	11.5
13	978.3	12.7	11.5
14	282.0	3.7	2.3
7718.5		100.0	100.0

表6 各PEのCFD計算のみのCPU時間(秒)：ONERA-M5 CASB-2

PE番号	時間	比率	セル数の比率 (%)
1	489.3	6.2	6.6
2	487.9	6.2	6.6
3	580.5	7.4	7.9
4	476.0	6.1	6.4
5	476.3	6.1	6.4
6	475.5	6.0	6.4
7	474.7	6.0	6.4
8	473.3	6.0	6.4
9	470.3	6.0	6.4
10	475.7	6.0	5.7
11	473.2	6.0	5.7
12	523.9	6.7	5.6
13	498.0	6.3	5.7
14	499.8	6.4	5.7
15	495.4	6.3	5.7
16	497.3	6.3	5.7
7867.0		100.0	100.0

表7 PE台数とCPU時間(秒)：ONERA-M5の周りの流れの計算 CASB-3

ブロック数-PE台数	繰り返し計算	PE間転送(A)	PE間転送(B)
14 - 84	476.5	3.7	約120

PE間転送(A): ブロック間の転送に起因するもの  
 (B): ブロック内の並列化に起因するもの