

# 並列計算の特性と評価

## Performance Evaluation and Characterization of Parallel Programings

中村孝\*、吉田正廣\*、山崎裕之\*  
Takashi Nakamura\*, Masahiro Yoshida\*, Hiroyuki Yamazaki\*

### ABSTRACT

We had already reported about parallel performance with CFD programs on the NWT and the PARAGON installed National Aerospace Laboratory.

In this paper, we analyze the characteristics of Laplace equation solver programs, number of operations, amount of memory and sequence of operations. And also we analyze the effective performance of execution, memory access latency and network bandwidth of these parallel computers.

We report the suitability of parallel programs and parallel computers with their characteristics of operations and network communications.

### 1. はじめに

筆者らはこれまでに、CFDアプリケーションプログラムおよびそのカーネルを用いて、航技研に設置されている数値風洞(NWT)、パラゴン上での並列化を行い、転送性能や演算性能、並列化効率など、プログラムおよび計算機の性能評価を行ってきた<sup>1),2)</sup>。

本報告では、ソフトウェア(プログラム: 解法、並列化)およびハードウェア(計算機、ネットワーク)のそれぞれの特徴を抽出することにより、対象とするプログラムと計算機の適合性について考察する。

ソフトウェアの解析には、CFDプログラムで特徴的に現われるラプラス方程式を例にとり、そのプログラムの構成や解法の違いによる特性、離散化や並列化の手法の違いによる特徴を解析する。

ハードウェアとしてはNWTおよびPARAGONを対象とし、その演算性能、メモリアクセス性能、データ転送性能等並列計算機の特徴を解析する。

以上の結果を用いて、ソフトウェアとハードウェア—プログラムと計算機—の適合性について論ずる。

### 2. 方程式

ここではCFDプログラムでよく現われ、また議論を理解しやすいものとするために、二次元ラプラス方程式

$$\nabla^2 \phi = 0 \quad (1)$$

を用いる。計算領域は等間隔直角構造格子

とし、2次精度の差分法

$$(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}) / \Delta x^2 + (\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}) / \Delta y^2 = 0 \quad (2)$$

により離散化する。

### 3. 数値解法と並列化

ラプラス方程式の陰的反復数値解法にはいくつかがあるが、本報告ではヤコビ法とADI法の二つを採用した。

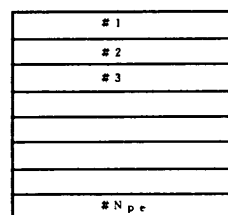
(1) ヤコビ法

ヤコビ法の計算手順は、

$$\phi_{i,j}^{n+1} = ((\phi_{i+1,j}^n + \phi_{i-1,j}^n) / \Delta x^2 + (\phi_{i,j+1}^n + \phi_{i,j-1}^n) / \Delta y^2) * \Delta x^2 * \Delta y^2 / 2 \quad (3)$$

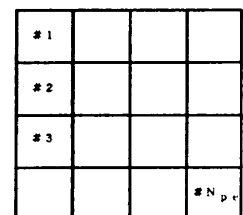
と表される。この解法では全ての次元方向に並列性があり、ベクトル化、並列化が容易である。しかし、他の方法に比べて収束速度が遅いという短所がある。式からもわかるように、 $n+1$ ステップのデータを計

一次元分割



演算量:  $N^2 / N_{pe} \times F$   
転送量:  $N \times 2$   
( $N_{pe}$ に無依存)

二次元分割



演算量:  $N^2 / N_{pe} \times F$   
転送量:  $N / \sqrt{N_{pe}} \times 4$

ただし一辺をNとする正方形領域

図1 ヤコビ法における領域分割

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

算するには  $n$  ステップでの上下左右方向のデータ ( $\phi_{i\pm 1, j\pm 1}$ ) を必要とする。

ヤコビ法の並列化は領域の1次元分割、2次元分割の2通りについて行った。図1の左側に示す1次元分割の場合、各プロセッサ (PE) は  $N$  個のデータを上下に隣接する PE に転送する必要がある。従ってこの場合の転送量は、 $N$  (データ数)  $\times 2$  (転送の相手方の数) となり、並列計算に使用する PE 数には依存しない。

右側の図で示される二次元分割の場合の転送量は  $N/\sqrt{N_{pe}}$  (データ数)  $\times 4$  (転送の相手方の数) と表される。この場合には使用する PE 数の増加に応じて、1 台の PE が転送すべき転送量は減少する。

## (2) ADI 法

次に、

$$\begin{aligned} & (-\phi_{i+1,j}^{n+1} + (2+a)\phi_{i,j}^{n+1} - \phi_{i-1,j}^{n+1}) / \Delta x^2 \\ & = (\phi_{i,j+1}^n - (2-a)\phi_{i,j}^n + \phi_{i,j-1}^n) / \Delta y^2 \\ & (-\phi_{i,j+1}^{n+2} + (2+a)\phi_{i,j}^{n+2} - \phi_{i,j-1}^{n+2}) / \Delta y^2 \\ & = (\phi_{i+1,j}^{n+1} - (2-a)\phi_{i,j}^{n+1} + \phi_{i-1,j}^{n+1}) / \Delta x^2 \quad (4) \end{aligned}$$

で表される ADI 法は収束速度がヤコビ法に比べて速いものの、常にどれか1方向の次元に逐次性があり、この方向にはベクトル化、並列化は不可能である。残りの方向に対しては並列性があるが、2次元の場合、ベクトル化と並列化の双方を一つの方で同時に行わなければならない。このため、NWTなどのベクトル型の並列計算機ではベクトル長の低下すなわち性能の低下は免れない。

$n+1$  ステップの計算には  $i$  方向に依存性があり  $i$  方向の全てのデータを必要とし、 $n$  ステップの  $j$  方向の  $j \pm 1$  のデータを必要とする。 $n+2$  ステップでは  $j$  方向に依存性があり、 $j$  方向の全てのデータを必要とし、 $n+1$  ステップの  $i \pm 1$  のデータを必要とする。

ADI 法に対しては次に述べる二通りの方法で並列化を行う。一つは領域を一次的に分割し、 $n \rightarrow n+1 \rightarrow n+2$  というステップ毎に、すなわち依存方向が異なる度に配列の転置を行う方法、もう一つは領域を2次的に分割し、各領域を PE に循環的に割り当てることにより、全ての PE で同時に実行開始を可能とする循環ウェーブ

フロント法である。

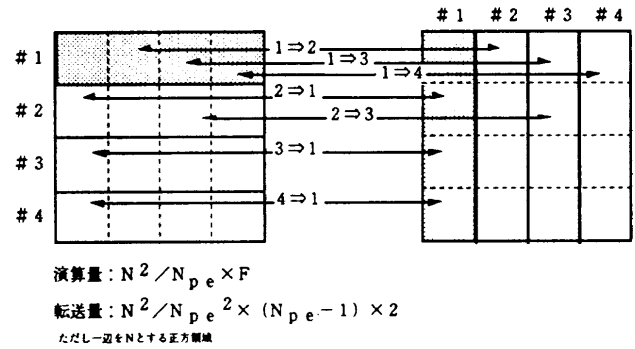


図2 転置型法

転置型の ADI 法では図2に示すように、各 PE に割り当てられた領域をさらに  $N_{pe}$  個のブロックに分け、その内の  $(N_{pe} - 1)$  個のブロックをそれぞれ別の PE に転送する必要がある。この場合、一つの PE が他の一つの PE に送る必要があるデータ量は  $N^2 / N_{pe}^2$  となる。これだけのデータ量を  $(N_{pe} - 1)$  台の PE に対して転送する必要がある。

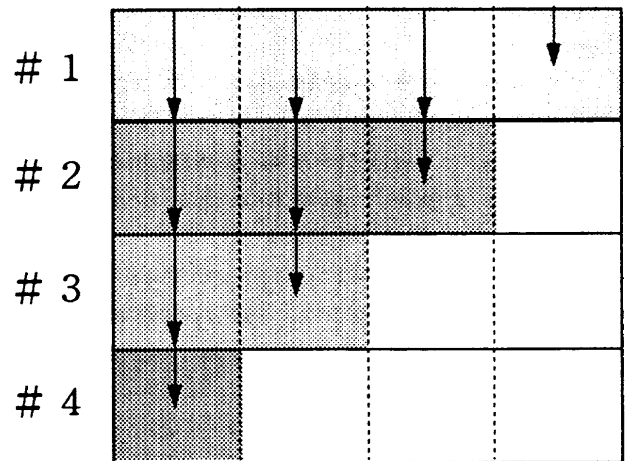


図3 ウェーブフロント法

通常ウェーブフロント法では、分割されていない、PE 内に閉じている方向について逐次性のあるループはそれぞれ並列に実効可能である。これは転置型と同様である。分割した方向に逐次性のある場合は、演算可能な PE から逐次実行開始し、転送を行いながら演算を進める。

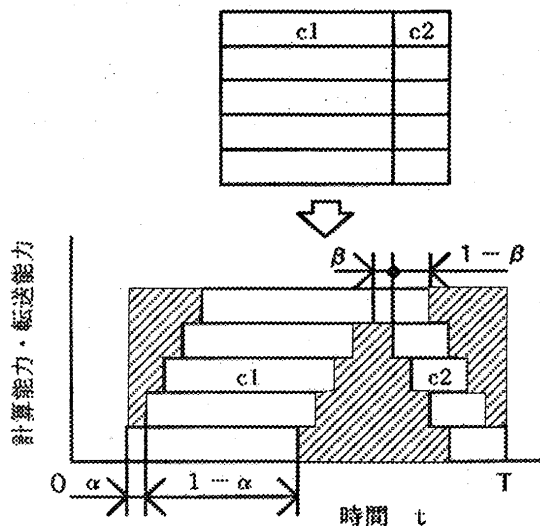


図4 ウェーブフロント法の効率

図3に通常のウェーブフロント法による演算の進め方を示す。図に示すように、前進代入過程では、一番目のPE(#1-PE)が初めのブロックを計算し、境界に達した時点で#2-PEにデータを渡し、#1-PE自身は次のブロックの計算を開始する。こうして計算が#4-PEまで到達した時点で#1-PEがまだ演算を行っていれば並列性はこの時点で100%になる。後退代入はこの逆になる。この場合の並列効率を図4に示す。この方法は格子点数および格子点当たりの演算数が十分多い場合にその並列性能を発揮できるが、今回のラプラス方程式のような2次元で比較的演算数が少ない場合は並列効果は期待できない。

3)。図中で $\alpha$ は前進代入における総時間内の転送時間を、 $\beta$ は後退代入時の総時間内の転倒時間である。図4でハッチングを施した部分がオーバーヘッドとなる。

これに対し、循環的ウェーブフロント法<sup>4)</sup>は、領域を2次元的に分割し、それぞれのブロックを図5に示すように循環的にPEに割り当ててものである。こうすることにより、全てのPEが同時に実行開始可能とと共に、依存方向が変化しても転置型のようにデータの配置替えをすることなく並列実行可能となる。しかし、どちらの方向に計算する場合でも、両方向ともに領域が分割されて異なったPEに割り当てられているため、データ転送が必要となる。

ベクトル長に間してはADI法ではいずれの場合も同様である。

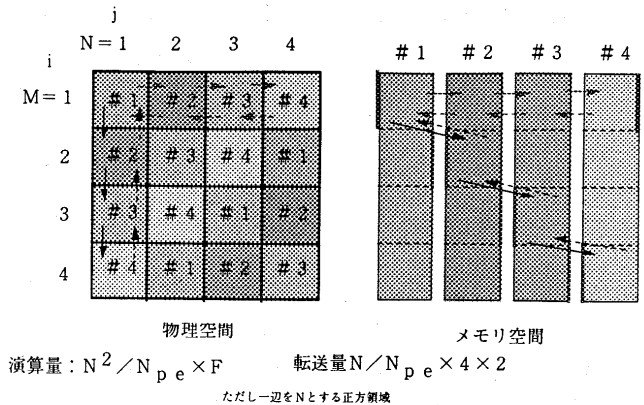


図5 循環的ウェーブフロント法のPE割付と転送量

ウェーブフロント法と転置型では格子点当たりの演算数は等しいが、転送量が異なる。ウェーブフロント法は転送先となる隣接PEは4で、1方向1回当たりのデータ転送量は $N/N_{pe} \times 4 \times 2$ を4方向となる。

#### 4. プログラムの解析

次にそれぞれのプログラムの使用している配列の数、加算、乗算、除算等の数と順序、ベクトル長、ループ内の演算量等のプログラムの詳細な解析を行う。また並列プログラムの場合は、並列実行開始／終了、データ転送、同期、その他並列化のためのオーバヘッドなどを解析する。さらに、ベクトル化率、並列化率などもプログラムの特徴付けとなる。データ転送では、ハードウェアのアーキテクチャに応じて転送先までの距離が相違するかしらないか、相手先が1対1、1対多、多対多、などの転送パターンに応じた類別も必要となる。

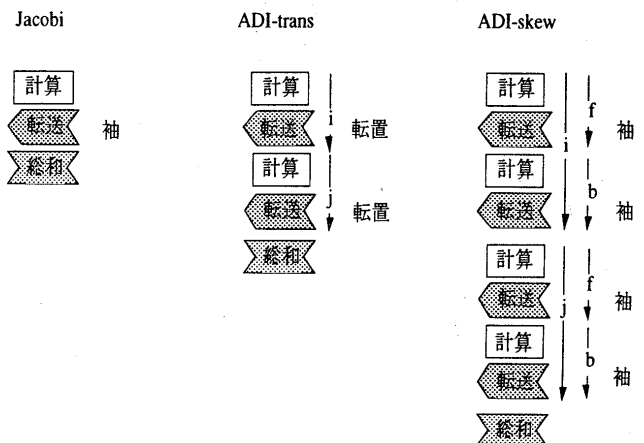


図6 プログラムフローチャート

図 6 にそれぞれの解法での転送パターンを示す。袖転送では 1 対 1、転置は多対多の転送を表す。

さらに、並列化の方法—データパラレルに基づく NWT-Fortran のコンパイラディレクティブによる並列化とメッセージパッシングライブラリによる並列化—によっても転送性能の性質が異なる。これらの違いは、呼び出されるライブラリの性能、機能の違いによる。

図 7 に並列処理性能に影響を及ぼすソフトウェア側、ハードウェア側の特徴量示す。

ソフトウェア  
 Npe: p e 台数  
 N: 格子数 (一辺が N の正方領域)  
 L: 転送データ量  
 F: 格子点当たりの演算量  
 p: 転送パターン  
 e: 並列化率  
 v: ベクトル化率  
 c: 収束性

ハードウェア  
 E: 実効演算性能  
 M: 実効転送性能 (転送パターンに依存)  
 Tm: 転送遅延時間

図 7 特徴量

一次元分割  
 $\text{演算数} = N^2 / N_{pe} \times F$   
 $\text{転送量} = N / \sqrt{N_{pe}} \times 2$   
 隣接転送、袖転送 (1 対 1)

二次元分割  
 $\text{演算数} = N^2 / N_{pe} \times F$   
 $\text{転送量} = N / \sqrt{N_{pe}} \times 4$   
 隣接転送、袖転送 (1 対 1)

図 8 ヤコビ法

図 8 にヤコビ法の特徴量を、図 9 に ADI 法の特徴量を示す。また図 10 ではそれぞれの特徴を模式的に表す。

転置型  
 $\text{演算数} = N^2 / N_{pe} \times F$   
 $\text{転送量} = N^2 / N_{pe}^2 \times (N_{pe} - 1) \times 2 + N \times 4$   
 転置転送 (多対多)、袖転送 (1 対 1)

循環ウエーブフロント法  
 $\text{演算数} = N^2 / N_{pe} \times F$   
 $\text{転送量} = N / N_{pe} \times (2 + 1) \times (N_{pe} - 1) \times 2 + N \times 4$   
 隣接転送、袖転送 (1 対 1)

図 9 ADI 法

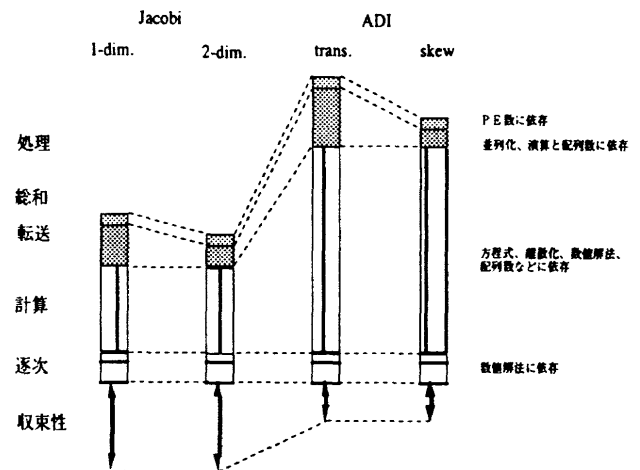


図 10 各解法による演算：転送の模式図

## 5. 計算機の解析

計算機は、実効演算性能、同時に実行可能な演算種類、実効メモリアクセス速度、スカラー/ベクトル機による違い、さらに並列機では、ネットワーク構成、実効転送遅れ、実効転送速度などにより特徴付けられる<sup>5)</sup>。

またデータ通信機構をハードウェアでどう実現しているかによる違いもある。すなわち、CPUで転送を制御しているかまたは独立なハードウェアを持っているかなどにより、転送性能が異なる。

ここでは、1 対 1 の転送性能および多対多の転送性能を計測する。1 対 1 は隣接転送や袖転送を、多対多は転置転送の場合を代表する。次に演算器の性能を計測するために、格子点数を変化させて非並列型プログラムの処理性能を計測する。

## 6. 適合性評価

プログラムと計算機の特徴が解析されれば、対象とするプログラムが計算機上でどのような性能を発揮するか、の適合性を推定することが可能になる。さらには、推定した性能が実測により発揮されない場合は、チューニング余地があることを示唆する場合も考えられる。

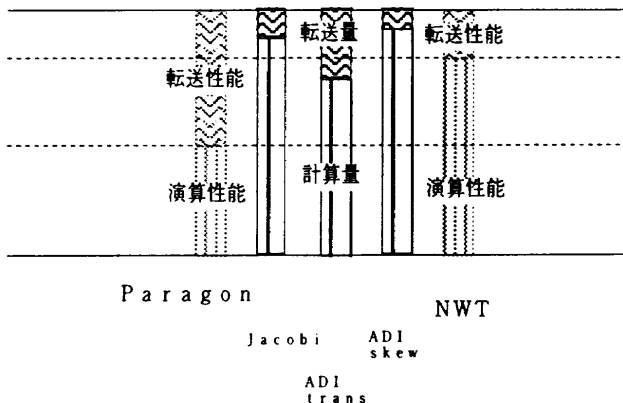


図 1.1 適合性の模式図

適合性の評価には、図 1.1 に示すようにそれぞれ計算機の持つ性能特性を無次元化する。またプログラムの持つ特徴も無次元化し、図 1.2 に示すようにそれらの比をとって表す。

$$\frac{N^2/N_{pe} \times F}{N^2/N_{pe} \times F + L} \left\{ \begin{array}{ll} < 1 & \text{演算性能過多} \\ & \text{転送性能過少} \\ = 1 & \text{適} \\ > 1 & \text{演算性能過少} \\ & \text{転送性能過多} \end{array} \right.$$

$$\frac{E}{E + M}$$

図 1.2 正規化した適合性

ここでは台数効果はスケールアップの場合のみを考える。この場合の演算数と転送量の比を表 1 に示す。表 1 で、ヤコビ法の格子点当たりの演算数は 12、ADI 法でのそれは 40 であり、全体の演算量と転送量に占める演算量の割合を下段に示す。これは図 1.2 での左辺の分子を表す。

表 1 演算と転送の比

Npe	N	Jacobi(12)		ADI(40)	
		1-dim.	2-dim.	trans.	skew
4	2,000	32,000	80,000	12,064,000	136,000
		0.997	0.993	0.768	0.997
16	4,000	64,000	80,000	15,128,000	308,000
		0.995	0.993	0.726	0.992
64	8,000	128,000	80,000	16,006,000	634,000
		0.989	0.993	0.714	0.984

表 2 にハードウェアの演算性能と転送性能の比を示す。表で、Ex は逐次版のプログラムのプログラムの主ループ部分の実効性能で表した演算性能である。単位は MFLOPS である。1:1, all:all は転送パターンの区別と転送性能を表している。単位は MB/s である。ratio は総時間に対する演算時間の比を表す。

表 2 ハードウェア演算と転送性能比

	Ex(Jac)	1:1	ratio	Ex(ADI)	all:all	ratio
NWT	1206	400	0.751	277	400	0.409
PARAGON	16	40	0.286	(16)	1.1	0.936

表 3 適合性

		Jac-1	Jac-2	ADI-tr	ADI-skew
NWT	4	1.33	1.32	1.88	2.44
	16	1.32	1.32	1.77	2.43
	64	1.32	1.32	1.75	2.41
PARAGON	4	3.49	3.48	0.82	1.07
	16	3.48	3.48	0.78	1.06
	64	3.46	3.48	0.76	1.05
	256	3.43	3.48	0.76	1.04

次に、表 3 に表 1 のプログラムの特徴と表 2 の計算機の適合性を表す比を示す。この比で、1 に近いほど適合性が高いことを示す。図 1.2 に示すように、適合性の値が 1 より大きい場合は、プログラムの性質からみて演算性能がデータ転送に比べて高

ざることを示す。逆に値が1より低い場合は、演算性能が貧弱であることを示す。表3から、ヤコビ法、ADI転置法では1次元分割、2次元分割ともPARAGONよりはNWTの方が適しており、逆に、ADI循環ウェーブフロント法ではNWTよりPARAGONの方が適合性が高いと言える。

さらにこれらを実時間で表すと図13のようになる。当然ではあるが、適合性が高いことと実際に処理時間が短いこととは無関係である。この図からスケールアップに対してはいずれの計算機も高い台数効果を発揮していることがわかる。

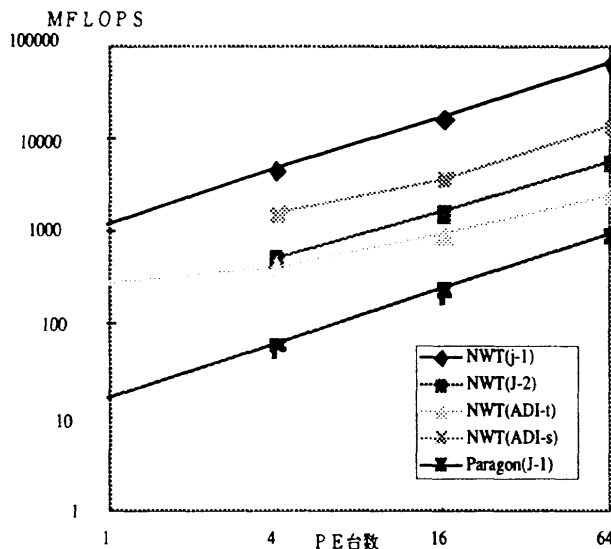


図13 実測性能

## 7. おわりに

この解析をさらに詳しく発展させることにより、CFDプログラムの詳細な評価や、モンテカルロ法の様な、データ転送パターンがこれまでの解析と異なるようなプログラムにも適用可能になる。また、新たにプログラムを開発する際には、実測や実シミュレーションを行う前に、あらかじめフローチャートにより実測やシミュレーションを行う前に適合性についての評価、実機による性能予測が可能になる。

ここで用いたプログラムはそのまま並列化における例題プログラムとして利用可能であると共に、MPIなどのメッセージパッシングライブラリの利用が可能な並列計算機上ではそのまま再利用することにより、他の計算機の上での特性を調べることも可能である。

推定部分の実測による確認、ここに取り上げた以外の解法や計算機についても適合性を調べ、一般性を引き出すこと、が今後の課題である。

## 参考文献

- [1] 中村孝、吉田正廣：非圧縮NSコードによるNWTの性能評価。第12回航空機計算空気力学シンポジウム講演論文集、1994年12月、航技研SP27
- [2] 中村孝、吉田正廣、山崎裕之：行列積における並列処理性能の評価。第14回航空機計算空気力学シンポジウム講演論文集、1997年1月、航技研SP34
- [3] 藤田直行、納富昭、上野潤一郎：NWTにおける3次元ADI並列計算法の性能比較。第14回航空機計算空気力学シンポジウム講演論文集、1997年1月、航技研SP34
- [4] Van der Wijngaart, R. F. : Efficient Implementation of a 3-dimensional ADI Method on the iPSC/860. Supercomputing '93, Portland, OR, November 1993, IEEE Press
- [5] 山崎裕之、吉田正廣、岩宮敏幸、Gregory Coussement, Bruno Troff, Juliette Ryan, Paul Sagaut : DNSコードの並列ベクトル化とその性能評価。第15回航空機計算空気力学シンポジウム論文集、to Appear