GPGPUコンピューティングによる
数値風況予測技術RIAM-COMPACT®の高速化

九州大学応用力学研究所
新エネルギー力学部門　風工学分野
内田　孝紀, 大屋　裕二
連絡先：takanori@riam.kyushu-u.ac.jp, 092-583-7776

1. はじめに

現在，世界では空前の勢いで風力発電産業が成長を遂げている．これは再生可能エネルギーの中で風力発電が最も脱化石燃料，CO2削減に対してコストパフォーマンスが高いからである．日本においても風力発電が最も有力な再生可能エネルギーであることは間違いなく，風力発電をより一層普及させることが，地球温暖化の克服「グリーン・イノベーション」に世界的な規模で貢献すると確信する．

風力発電分野において，今後解決すべき技術課題の一つは，風車に対する局地的な風況を正確に把握し，風車に対する局所的なウインドリスク(地形乱流)を特定できる数値風況予測技術を確立することである[1-3]．

我々の研究グループが開発を進める数値風況予測技術RIAM-COMPACT®は，これらの諸問題を一挙に解決する潜在的可能性を秘めている[4]．RIAM-COMPACT®(Research Institute for Applied Mechanics, Kyushu University, COMputational Prediction of Airflow over Complex Terrain；リアムコンパクト)に関して，そのコア技術は九州大学応用力学研究所で開発が続けられており，2006年に著者らが起業した九州大学発ベンチャー企業の(株)リアムコンパクト(http://www.riam-compact.com/)が(株)産学連携機構九州(九大TLO)から独占的ライセンス使用許諾を受けている(2006年にRIAM-COMPACT®の商標と実用新案を取得)．現在では，九州電力グループの西日本技術開発(株)，(株)環境GIS研究所，(株)FSコンサルティングと開発コンソーシアムを作り，「実地形版RIAM-COMPACT®ソフトウエア」と名付け，業界標準モデルの一つとして広く普及に努めている．現在では，国内の風力事業者最大手の(株)ユーラスエナジージャパン，電源開発(株)，日本風力開発(株)エコ・パワー(株)を含め，多数の導入実績を有する．

非定常な乱流シミュレーションに主眼を置いたRIAM-COMPACT®では，計算時間の問題が懸念されてきた．現行の流体計算ソルバーは，Intel Core i7などのマルチコアCPU(Central Processing Unit)に対応しており，計算時間は劇的に短縮され，実用面での利用において特段の問題は無くなってきた．

さらに，現在ではGPGPUコンピューティングへの対応にも成功した．GPGPU(General Purpose computing on GPU：GPUによる汎用計算)のコンセプトとは，グラフィック・レンダリングのみならず，GPU(Graphics Processing Unit)が有する浮動小数点演算能力を，他の数値演算にも幅広く利用することである．

本報では，マルチコアCPUおよびシングルGPUを利用した数値風況予測技術RIAM-COMPACT®の高速化の結果について報告する．

2. 実地形版RIAM-COMPACT®ソフトウエアの概要

本研究では，数値不安定を回避し，複雑地形上の局所的な風の流れを高精度に数値予測するため，一般曲線座標系のコロケート格子に基づいた実地形版RIAM-COMPACT®ソフトウエアを用いた．ここでコロケート格子とは，計算格子のセル中心に物理速度成分と圧力を定義し，セル界面に反変速度成分にヤコビアンを乗じた変数を定義する格子系である．数値計算法は差分法(FDM；Finite-Difference Method)に基づき，乱流モデルにはLES(Large-Eddy Simulation)を採用する．LESでは流れ場に空間フィルタを施し，大小様々なスケールの乱流渦を，計算格子よりも大きなGS(Grid Scale)成分の渦と，それよりも小さなSGS(Sub-Grid Scale)成分の渦に分離する．GS成分の大規模渦はモデルに頼らず直接数値シミュレーションを行う．一方で，SGS成分の小規模渦が担う，主としてエネルギー消散作用はSGS応力を物理的考察に基づいてモデル化される．

流れの支配方程式は，空間フィルタ操作を施された非圧縮流体の連続の式(式(1))とナビエ・ストークス方程式(式(2))である．本研究では，平均風速6m/s以上の強風を対象にしているので，大気が有する高度

方向の温度成層(密度成層)の効果は省略した．また，地表面粗度の影響は地形表面の凹凸を高解像度に再現することで取り入れた．

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \qquad -(1)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} \qquad -(2)$$

$$\tau_{ij} \approx \overline{u_i' u_j'} \approx \frac{1}{3}\overline{u_k' u_k'}\delta_{ij} - 2\nu_{SGS}\bar{S}_{ij} \qquad -(3)$$

$$\nu_{SGS} = \left(C_s f_s \Delta\right)^2 \left|\bar{S}\right| \qquad -(4)$$

$$\left|\bar{S}\right| = \left(2\bar{S}_{ij}\bar{S}_{ij}\right)^{1/2} \qquad -(5)$$

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \qquad -(6)$$

$$f_s = 1 - \exp\left(-z^+ / 25\right) \qquad -(7)$$

$$\Delta = \left(h_x h_y h_z\right)^{1/3} \qquad -(8)$$

計算アルゴリズムは部分段階法(F-S法)[5]に準じ，時間進行法はオイラー陽解法に基づく．圧力に関するポアッソン方程式は逐次過緩和法(SOR法)により解く．空間項の離散化は式(2)の対流項を除いて全て2次精度中心差分とし，対流項は3次精度風上差分とする．ここで，対流項を構成する4次精度中心差分は，梶島による4点差分と4点補間に基づいた補間法[6]を用いる．3次精度風上差分の数値拡散項の重みは，通常使用される河村-桑原スキーム[7]タイプの α =3に対して，α =0.5とし，その影響は十分に小さくする．LESのサブグリッドスケールモデルには標準スマゴリンスキーモデル[8]を用いる．壁面減衰関数を併用し，モデル係数は0.1とした．

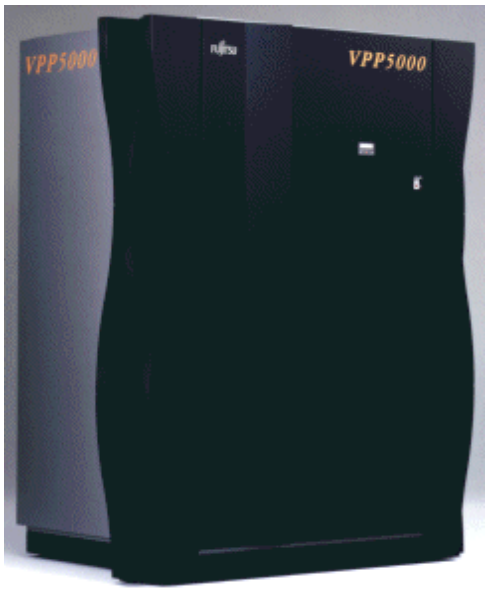
3. 本研究で使用した計算機環境の概要

ここでは，本研究で使用した計算機環境について説明する．図1および図2には，GPUの検証に用いた計算機環境などを示す．この機種のGPUには，図2に示すように，「NVIDIA® Tesla™ C2050 3GB」が搭載されている．また，Windows上で起動するGPU用の実行バイナリを作成するため，PGI社とNVIDIA社が共同開発したPGI CUDA Fortranコンパイラ (v.11.4)をインストールした．さらに，マルチコアCPUを使った性能を評価するために，Intel Composer XE 2011 (Update 3)も同じ機種にインストールした．

【構成】
CPU : Intel Xeon X5680 (3.33GHz, 6コア) × 2 (合計12コア)
GPU : NVIDIA Tesla C2050 x 1 (3GB)
Mem : 48GB DDR-3 SDRAM
OS : Windows7 64bit版
GPU用Compiler : PGI Accelerator Fortran/C/C++ WS
for Windows 32/64bit (v.11.4)
CPU用Compiler : Intel Composer XE 2011 (Update 3)
CUDA (Compute Unified Device Architecture) : CUDA 3.2

図1 本研究で使用したHP製Z800 Workstation

図2 本研究で使用したGPU,
NVIDIA Tesla C2050 (3GBメモリ)



【構成】
PE数 ：1PE (9.6GFLOPS)
Mem : 4GB

図3 本研究で使用したFUJITSU製VPP5000U
（ベクトル計算機，1999年リリース）

【構成】
CPU数 ：6CPU (92.16GFLOPS/単体×6=552.96GFLOPS)
Mem : 256GB
外付けディスク装置 ：iStorage D3-10 4TB (RAID5)

図4 本研究で使用したNEC製SX-9F
（ベクトル計算機，2007年リリース）

図3および図4には，本研究で使用したベクトル計算機の概要を示す．いずれも，九州大学応用力学研究所がこれまでに所有していた汎用機である．現在は，NEC製SX-9Fのみが稼動中である．

4．本研究で対象とした流れ場と計算条件

本研究では，3次元の孤立峰を過ぎる流れ場[4]を対象として，計算時間の比較を行った．孤立峰の形状は下記の関数で表現される．

$$z(r)=0.5h\times\{1+\cos(\pi r/a)\},\ r=(x^2+y^2)^{1/2},\ a=2h \quad -(9)$$

計算領域と座標系を図5に示す．主流方向(x)，主流直交方向(y)，鉛直方向(z)に40h(±20h)×9h×10hの空間領域を有する．ここで，hは孤立峰の高さである．格子点数はx，y，z方向に260×121×71点(合計約223万点，使用するメモリ容量は約800MB)である．孤立峰の近傍における計算格子を図6に示す．x方向の格子幅は不等間隔に(0.04～1)×h，y方向の格子幅は不等間隔に(0.05～0.5)×h，z方向の格子幅は不等間隔に(0.0035～0.5)×hである．

境界条件について説明する(図5および図7を参照)．流入境界面には，建設省告示1454号に示された地表面粗度区分Ⅲに従う速度プロファイルを与えた．

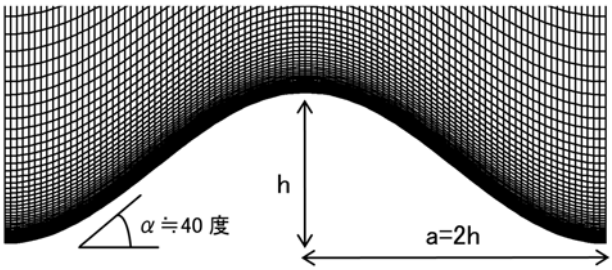


図6 孤立峰近傍の計算格子，
主流直交方向(y)の中央面(y=0)

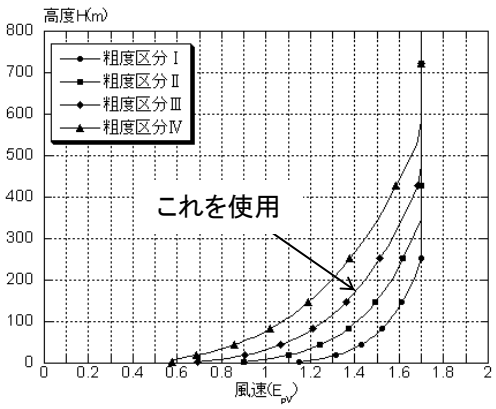


図7 本研究で使用した流入境界条件，
地表面粗度区分Ⅲを適用

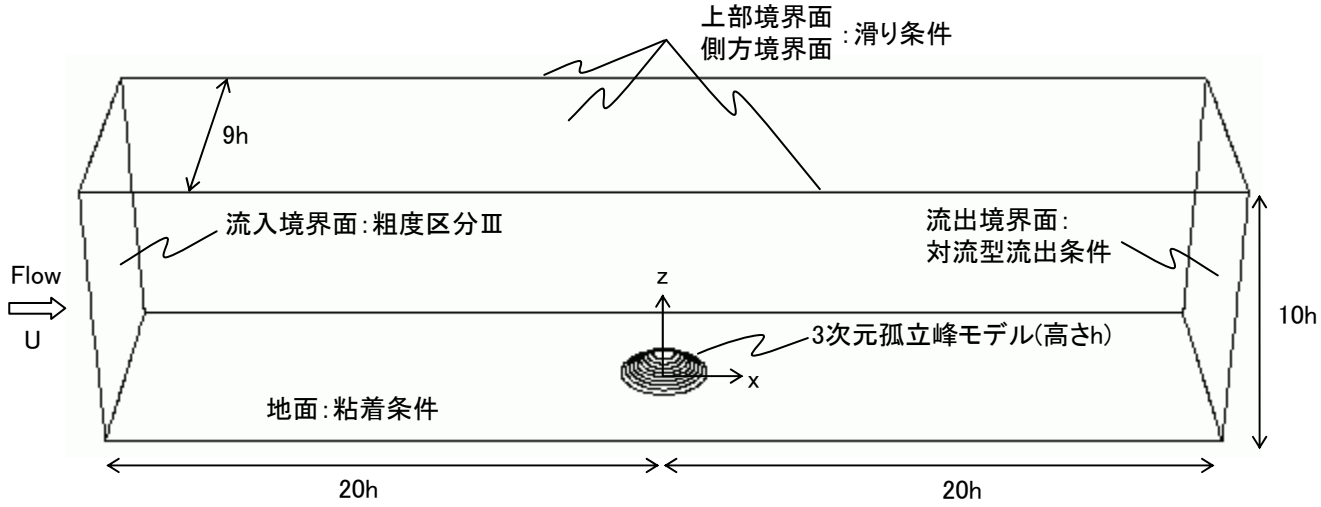
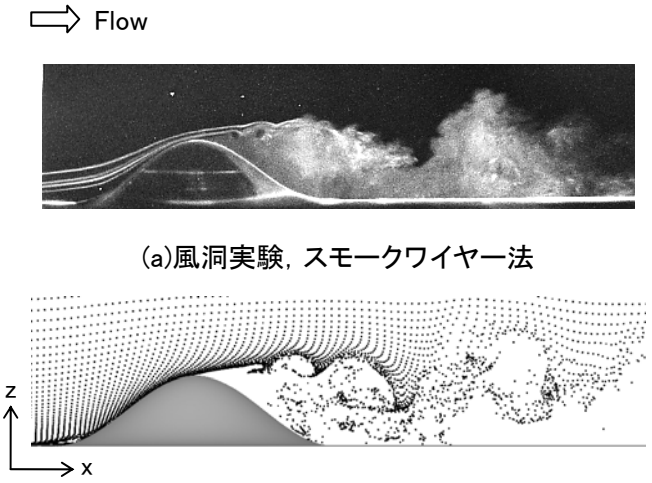


図5 計算領域と座標系



(a)風洞実験，スモークワイヤー法

(b)数値計算，パッシブ粒子追跡法

図8 孤立峰近傍における流れ場の可視化，瞬間場

側方境界面と上部境界面は滑り条件，流出境界面は対流型流出条件とした．地面には粘着条件を課した．レイノルズ数は，孤立峰の高さhと流入境界面における高さhでの風速Uに基づき，$Re(=Uh/\nu)=10^4$とした．時間刻みは$\Delta t=2\times10^{-3}h/U$とした．

本研究で対象とした孤立峰周辺に形成される流れパターンについて説明する(図8を参照)．図8(a)に示す風洞実験では，スモークワイヤー法により流れ場の可視化を行った．この方法では以下のように流れ場を視覚化する．モデルのすぐ上流で高さレベルを変えて数本のワイヤー(0.3mmのニクロム線)を平行に配線する．これに流動パラフィンとアルミ粉を混ぜたものを塗り，ワイヤーに通電して加熱し，気化した煙で流れ場を可視化する．照明装置としてスリットを付けた1kWのプロジェクターを風洞上部に3～4台設置し，これからの光でモデルの主流直交方向(y)の中央面(y=0)を可視化した．カメラによる撮影は標準レンズを用い，絞りは1.2でシャッタースピード(露出時間)は1/125sとした．風速は1.5m/sとし，実地形版RIAM-COMPACT®ソフトウエアによる数値シミュレーションと同じ条件($Re=10^4$)とした．特に孤立峰の頂部付近で剥離した境界層(剥離せん断層)の挙動に注目するため，煙が孤立峰の表面近くを流れるようにワイヤー高さを調節した．

一方，図8(b)に示す実地形版RIAM-COMPACT®ソフトウエアによる数値シミュレーションでは，パッシブ粒子追跡法により流れ場の可視化を行った．パッシブ粒子の放出間隔は$\Delta t=0.1h/U$とし，合計100コマ(時間t=100～110h/U)の結果である．数値シミュレーションと風洞実験で得られた流れの定性的な挙動は，非常に類似している．すなわち，流れは孤立峰の頂部付近で剥離する．この剥離したせん断層は，孤立した渦に巻き上がる．孤立渦は合体して大規模渦を形成し，これが孤立峰の下流に放出されている．

5．マルチコアCPUおよびシングルGPUの計算速度

本研究では，孤立峰の周辺で形成された流れ場(t=100h/U)を初期値とし，そこから5,000ステップの計算(時間t=100～110h/U)を実行し，その経過時間を比較した．

表1には，我々の研究室が所有するHP製Z800 Workstationの結果を示す．表中の「Auto」とは，Intel Fortranコンパイラの自動並列化オプション「/Qparallel」を利用した並列計算(OpenMP)の結果である．一方，「Manual」とは，計算プログラムにOpenMP用の並列化指示行(directive)を事前に挿入し，Intel Fortranコンパイラの「/Qopenmp」オプションの指定により，それらを有効にした場合の結果である．まず，CPUによる並列計算の結果に注目すると，予想されるように「Auto」に比べて「Manual」の方が速度向上率は良い値を示している．両ケースともに，1コア

表1 HP製Z800 Workstationの結果，約223万格子点の場合

ハードウエア 構成	OS	CPU1コア ※1	CPU12コア		シングルGPU ※4
			Auto※2	Manual※3	
CPU : Intel Xeon X5680 (3.33GHz, 6コア) × 2 (合計12コア) 「Westmere」 GPU : NVIDIA Tesla C2050 x 1 (3GB)	Windows 7 (64bit)	3,727.93(s)	1,165.38(s)	902.61(s)	391.74(s)
		速度向上率	3.20倍(vs 1コア)	4.13倍(vs 1コア)	9.52倍(vs 1コア)
		0.56倍※5	1.79倍※5	2.31倍※5	5.33倍※5

備考

※1 Intel Fortranコンパイラの「/fast」を適用.

※2 Intel Fortranコンパイラの「/fast /Qparallel」を適用.

※3 Intel Fortranコンパイラの「/fast /Qopenmp /fpp」を適用.

※4 PGIコンパイラの「-fastsse -ta=nvidia, cuda3.2, cc2.0」(GPU使用)を適用. ECC(Error Checking and Correcting)設定=OFF

※5 FUJITSU製VPP5000Uの1PEの結果(2088.99s)との比較

表2 HPCシステムズ(株)の検証機の結果，約223万格子点の場合

ハードウエア 構成	OS	CPU1コア ※1	CPU4コア	シングルGPU ※3
			Manual※2	
CPU:Intel Core i7 2600K (3.4GHz) x 1 「Sandy Bridge」 GPU:NVIDIA GeForce GTX580 × 1 (1.5GB)	Windows 7 (64bit)	2,625(s)	1,288(s)	310(s)
		速度向上率	2.04倍(vs 1コア)	8.47倍(vs 1コア)
		0.80倍※4	1.62倍※4	6.74倍※4

備考

※1 Intel Fortranコンパイラの「/fast」を適用.

※2 Intel Fortranコンパイラの「/fast /Qopenmp /fpp」を適用.

※3 PGIコンパイラの「-fastsse -ta=nvidia, cuda3.2, cc2.0 -Mpreprocess」(GPU使用)を適用. ECC設定=なし

※4 FUJITSU製VPP5000Uの1PEの結果(2088.99s)との比較

表3 HPCシステムズ(株)の検証機の結果，約223万格子点の場合

ハードウエア 構成	OS	CPU1コア ※1	CPU4コア	シングルGPU ※3
			Manual※2	
CPU:Intel Core i7 2600K (3.4GHz) x 1 「Sandy Bridge」 ※AVX使用 Advanced Vector eXtensions SSEに続くIntel X86 CPUの新しい SIMD演算(ベクトル演算)命令セット GPU:NVIDIA GeForce GTX580 × 1 (1.5GB)	Windows 7 (64bit)	2,208(s)	1,193(s)	310(s)
		速度向上率	1.85倍(vs 1コア)	7.12倍(vs 1コア)
		0.95倍※4	1.75倍※4	6.74倍※4

備考

※1 Intel Fortranコンパイラの「/fast /QxAVX」を適用.

※2 Intel Fortranコンパイラの「/fast /Qopenmp /fpp /QxAVX」を適用.

※3 PGIコンパイラの「-fastsse -ta=nvidia, cuda3.2, cc2.0 -Mpreprocess」(GPU使用)を適用. ECC設定=なし

※4 FUJITSU製VPP5000Uの1PEの結果(2088.99s)との比較

表4 HP製Z800 Workstationの結果，約400，600，800万格子点の場合

格子点数	400万点	600万点	800万点
CPU1コア	6,007(s)	9,731(s)	12,768(s)
GPU※ECC無効	759(s)	1,189(s)	1,599(s)
速度向上率	7.91倍(vs 1コア)	8.18倍(vs 1コア)	7.98倍(vs 1コア)

の結果に比べて3～4倍程度の速度向上率を示しており，数年前のスーパーコンピュータ1PE以上の性能を有することが分かる．次に，GPUの結果に注目して頂きたい．NVIDIA Tesla C2050には，ECC(Error Checking and Correcting)機能が備えられている．ECCとは，メモリに誤った値が記録されていることを検出し，正しい値に訂正する機能である．今回の性能評価において，ECCのON/OFF設定の違いで計算結果に差異が無いこと，また，ECC設定=OFFにした方が，ECC設定=ONにした場合に比べて計算速度が速いことを確認した．一枚ざしのGPUの結果であるにも関わらず，驚異的な速度向上率を示しているのが分かる．その速度向上率の値は，1コアのCPUの結果に対して約9倍である．この値は，表5に示す最新のスーパーコンピュータの4～6CPUを利用した並列計算の結果に並ぶ程である．

本研究では，HPCシステムズ(株)の協力の下，最新のCPU(Intel Core i7 2600K（3.4GHz），「Sandy Bridge」)を搭載した計算機環境での性能評価も実施した．「Sandy Bridge」には，Intel AVX(Advanced Vector eXtensions)と言う新しい拡張命令セットが導入された．これはSSE(Streaming SIMD Extensions)に続く，Intel X86 CPUの新しいSIMD演算(ベクトル演算)命令セットである．これは，浮動小数点演算を高速化する技術であり，流体シミュレーションの計算時間の高速化が期待できる．本研究では，Intel AVXの性能についても検討した．一方，GPUには，図9に示すNVIDIA GeForce GTX580を搭載している．GTX580にはECC機能が無いものの，Tesla C2050に比べて6分の1ほどの価格である．実務面での利用においては，非常に有力なGPUであると言える．表2および表3のCPUの結果に注目すると，Intel AVX技術によりCPUの計算は2割ほど向上しているのが分かる．GPUの結果に注目する．NVIDIA GeForce GTX580の結果は，NVIDIA Tesla C2050の結果に比べて，2割程度の速度向上が確認された．

図9 本研究に使用したGPU，
NVIDIA GeForce GTX580 (1.5GB メモリ)

表4には，格子規模を変化させた場合のHP製Z800 Workstationの結果を示す．今回の性能評価において，NVIDIA Tesla C2050(3GBメモリ)を使用した場合には，800万点までの計算が実行できることを確認した．しかしながら，1,000万点の計算はメモリ容量から実行することができなかった．表4に注目すると，格子規模を変化させた複数の場合においても，1コアのCPUを用いた計算結果に比べて8倍程度の速度向上率が示された．

6. おわりに

本研究では，マルチコアCPUおよびシングルGPUを利用した数値風況予測技術RIAM-COMPACT®の高速化について調査した．

その結果，NVIDIA社のTesla C2050やGeForce GTX580を利用したGPU計算では，800万点規模の数値流体シミュレーションが可能であること，また，1コアのCPUを用いた計算結果に比べて8倍程度の驚異的な速度向上率を有することが示された．特に，約223万点を用いた計算では，最新のスーパーコンピュータの4～6CPUを利用した並列計算の結果に並ぶ程の高速化が達成されていることが分かった．

シングルGPUで使用できるVRAM(Video Random Access Memory，ビデオメモリ)の容量は，現在のところTesla C2070の6GBが最大である．今後，シングルGPUのメモリ容量が増加し，数千万点以上の大規模計算が実現することを大いに期待したい．その一方で，シングルGPUのメモリ容量に入りきらないような大規模計算を行うために，マルチGPUの利用も検討されている．但し，この場合にはMPI(Message Passing Interface)などによるノード間の並列化処理が別途必要になってくる．

Intel Core i7などのマルチコアCPUを利用した並列計算(OpenMP)においても，計算時間の大幅な短縮

表5 NEC製SX-9F(ベクトル計算機)の結果，約223万格子点の場合

ハードウエア構成	1CPU※1	2CPU※1	4CPU※1	6CPU※1
CPU数：6CPU (92.16GFLOPS)	709.28(s)	469.49(s)	344.46(s)	302.42(s)
	速度向上率	1.51倍(vs 1CPU)	2.06倍(vs 1CPU)	2.35倍(vs 1CPU)
	2.94倍※2	4.45倍※2	6.06倍※2	6.91倍※2

備考

※1 Fortranコンパイラの「-Pauto」を適用.

※2 FUJITSU製VPP5000Uの1PEの結果(2088.99s)との比較

表6 NEC製SX-9F(ベクトル計算機)の結果，約400，600，800，1,000万格子点の場合

格子点数	400万点※1	600万点※1	800万点※1	1,000万点※1
CPU数：6CPU	455.93(s)	682.00(s)	893.53(s)	1097.26(s)
Vs 約223万点の結果 302.42(s)	1.51倍	2.26倍	2.95倍	3.63倍

備考

※1 Fortranコンパイラの「-Pauto」を適用.

が確認された．また，最新のCPU(Intel Core i7 2600K (3.4GHz)，「Sandy Bridge」)に新たに導入されたIntel AVX機能による明確な速度向上も確認された．

以上のように，ハードウエアの急速な向上に伴い，風力発電産業界においても，RIAM-COMPACT®のような非定常乱流解析が主流になると予想される．

付 録

参考のため，表5および表6には，マルチコアCPUおよびシングルGPUの計算時間の比較に用いたベクトル計算機(NEC製SX-9F)の結果を示す．

謝 辞

本研究の一部は，平成22年度「戦略的国際標準化推進事業／標準化研究開発(グリーンイノベーション推進事業)／数値シミュレーション技術を用いた風車性能評価技術等の国際標準化に係る研究開発」による援助を受けました．また，HPCシステムズ(株)にはマルチコアCPUおよびシングルGPUの計算時間の評価などで多大な協力を得ました．ここに記して感謝の意を表します．

参 考 文 献

1) 村上周三，持田灯，加藤信介，木村敦子：局所風況予測システムLAWEPSの開発と検証，日本流体力学会誌「ながれ」，Vol.22，No.5，pp.375-386，2003.

2) 石原孟：非線形風況予測モデルMASCOTの開発とその実用化，日本流体力学会誌「ながれ」，Vol.22，No.5，pp.387-396，2003.

3) Sumner，J.，Watters，C.S. and Masson，C. : Review : CFD in Wind Energy : The Virtual，Multiscale Wind Tunnel，Energies，Vol.3 ，pp.989-1013，2010.

4) Uchida，T. and Ohya，Y. : Micro-siting Technique for Wind Turbine Generators by Using Large-Eddy Simulation，Journal of Wind Engineering & Industrial Aerodynamics，Vol.96，pp.2121-2138，2008.

5) Kim，J. and Moin，P. : Application of a fractional-step method to incompressible Navier-Stokes equations，J. Comput. Phys.，Vol.59，pp.308-323，1985.

6) 梶島岳夫，太田貴士，岡崎和彦，三宅裕：コロケート格子による非圧縮流れの高次差分解析，日本機械学会論文集，(B編)，63巻，614号，pp.3247-3254，1997.

7) Kawamura，T.，Takami，H. and Kuwahara，K. : Computation of high Reynolds number flow around a circular cylinder with surface roughness，Fluid Dyn. Res.，Vol.1，pp.145-162，1986.

8) Smagorinsky，J. : General circulation experiments with the primitive equations，Part 1，Basic experiments，Mon. Weather Rev.，Vol.91，pp.99-164，1963.