

<正方格子を用いた適応格子法の並列計算に関する考察>

<A Parallel Method for Adaptive Refinement of a Cartesian Mesh Solver>

- 古山 彰一, 富山商船高専, 富山県新湊市海老江練合 1-2, E-mail : shoichi@toyama-cmt.ac.jp
- 松澤 照男, 北陸先端大, 石川県能美郡辰口旭台 1-1, E-mail : matuzawa@jaist.ac.jp
- Sho-ichi FURUYAMA, Toyama NCMT, 1-2, Neriya, Ebie, Shinminato, Toyama, Japan
- Teruo MATSUZAWA, JAIST,1-1, Asahidai, Tatsunokuchi, Nomi, Ishikawa, Japan

<Abstract> We investigate a parallel procedure for the Adaptive Mesh Refinement (AMR) method. When we choose the AMR method to analyze a flow-field on parallel computer, we have to include a load balancing method to equal a load of each PEs. To realize the effective load balancing, we had to consider about a quantity of transferring data and an ability of environments of calculations. In this paper, to consider above topics, we indicate an orientation of construction for the parallel AMR.

1. はじめに

数値流体工学分野では、計算機能力向上に伴い 3 次元での流れ場解析が一般化している。この背景に伴い、複雑形状物体に対する格子生成問題が多く取り上げられるようになってきている。この問題に対するシンプルでかつ効果的な解決策として、姫野らの直交座標系を用いたボクセル法¹や、著者らの適応格子法²といった計算手法が挙げられる。本研究では特に適応格子法に対してスポットをあてる。

適応格子法は、1980 年代初頭に 2 次元の流れ場で衝撃波をシャープに捕らえるために Berger³らによって提案された手法である。計算資源の効果的利用、計算時間の大幅な削減、さらに、精度が良い手法ということで、現在でも頻繁に話題にされる手法である。特に近年では、計算機能力向上により 3 次元の大規模な解析に際し適応格子法が話題にされることが多く、太陽風観測データを用いた現象予測のリアルタイムシミュレータとして Groth⁴らによって採用されるなど、実用的なレベルで利用される手法となってきた。さらにこの手法は、ここ 1, 2 年の間に、その範囲を移動境界問題や、非圧縮性流体解析の分野へと広げてきている⁵。これらの分野では、これまで形状が複雑であることから、計算コスト（メモリ量、計算時間）、格子生成の面で非常に多くのコストがかかっていた問題に対する解決策として適応格子法を位置付けることができる。

また一方で、PC クラスタリングを用いた並列計算手法に関する研究が大きなシェアを占めるようになってきた。これは手軽にギガフlops程度の能力を持つパーソナルユースの並列計算環境の構築が容易に可能であることから、ここ数年の間に急速に普及した。しかしながら、このシステムは、計算機能力の向上は容易であるが、通信性能については 100 BASE-TX 等の汎用ネットワークを利用するため限度がある。そのため、この様な計算環境の特性を踏まえたハイパフォーマンスな並列計算手法の確立が今後重要になると考えられる。

本研究では、以上の背景を考慮して、適応格子法により、PC クラスタ計算環境の下で流体解析を行う場合の並列計算手法について検討を行う事を目的とする。

2. 非並列計算部

ここでは、並列化を行う前段階の手法について述べる。本研究をすすめるにあたり、これらの非並列計算コードを作成してからそれを並列計算用コードに拡張する流れで行う。

2.1 計算ソルバ

解析する対象の流体は、圧縮性非粘性流体とし、有限体積形式の以下の Euler 方程式を解く。

$$\frac{d\mathbf{U}}{dt} = -\frac{1}{A} \sum_{faces} (\mathbf{F}\Delta y - \mathbf{G}\Delta x), \quad (1)$$

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad (2)$$

$$\mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho v \\ \rho E \end{pmatrix}, \quad (3)$$

$$\mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v H \end{pmatrix}. \quad (4)$$

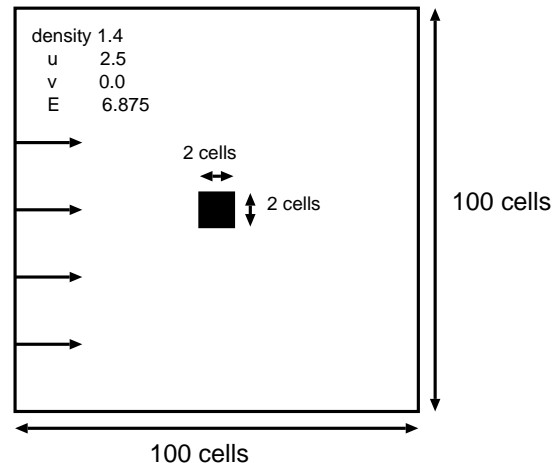


Fig. 1: Initial Condition

ここで、 A はセルの面積を表し、 Δx , Δy はセルの辺長を表す。また、周積分方向は半時計周りとする。

解析手法は、Zeeuw⁶らに従い、linear reconstruction method, 近似リーマン解法, multi-stage time-stepping scheme を利用した。multi-stage time-stepping scheme については 5 段階のものを利用し、CFL 数を 1.1508 で固定した。また、時間積分については、local time-stepping を用いている。

計算対象は、Fig.1の様に空間内に正方形の物体を置き、風を流し込む場合を考えた。初期セル数は 100×100 ($\Delta x = 0.5, \Delta y = 0.5$) とし、物体の大きさを 1×1 (4セルで構成) とし、計算空間の中心に置いた。物理量の初期値は、 $\rho = 1.4, u = 2.5, v = 0, E = 6.875$ とし、計算を行っている間の流入口の境界条件も同じとした。

2.2 適応格子法

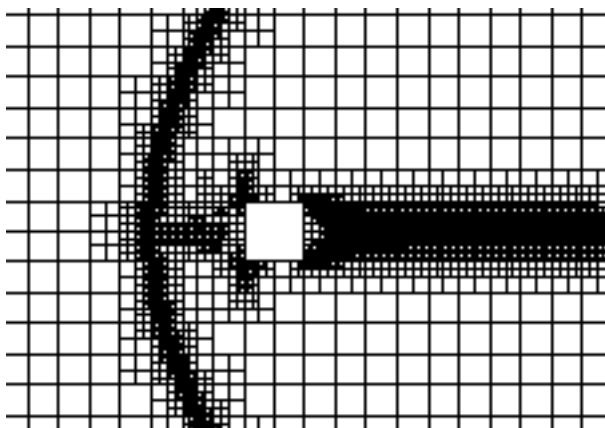


Fig. 2: Square initial mesh: 100×100 cells, level 5 mesh

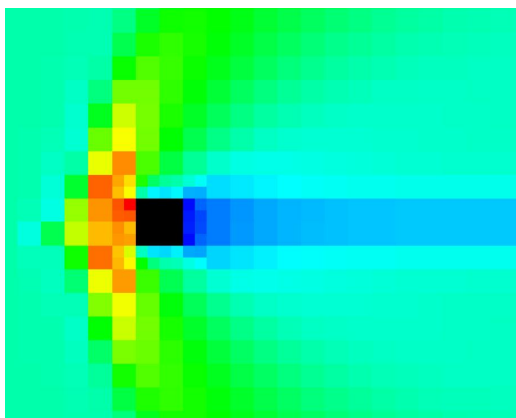


Fig. 3: Square initial mesh: 10032 cells, level 1 mesh

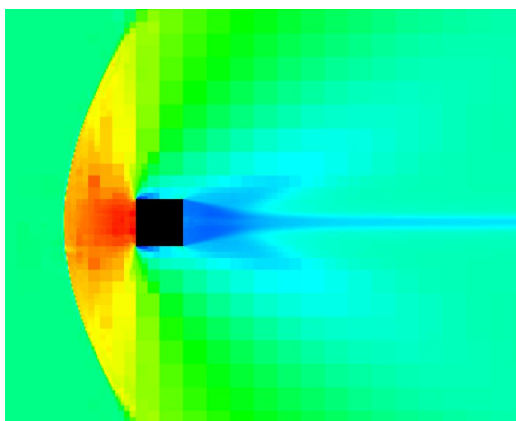


Fig. 4: Square mesh: 26787 cells, level 5 mesh

適応格子法を実現するに際して、1セルあたり以下の合計49個(284byte)の情報を格納することにした。具体的には以下の情報である。()内の数字は、その項目に対して必要なデータ数を示す。

整数型: 親格子へのポインタ、子格子へのポインタ (4)、格子の状態 (固体、流れ場、境界など)、格子の細分化レベル、一時のデータ (2)、隣接格子へのポインタ (1 2)

実数型: 物理量 (4)、物理量一時データ (4)、各物理量の残差 (4)、時間刻み、勾配 (8、物理量 4×2 方向)、線形補間制限値、座標 (2)、セル面積

細分化対象フラグは、隣接セルとの速度差を計算し、計算領域全体での最大値速度差の80%以上のセルについて立てることとした。また、分割ルールとして、隣接セルとは細分化レベルの違いが1しか許さないこととし、実際の細分化は等面積で4分割することとした (そのため子格子へのポインタは4種類必要)。

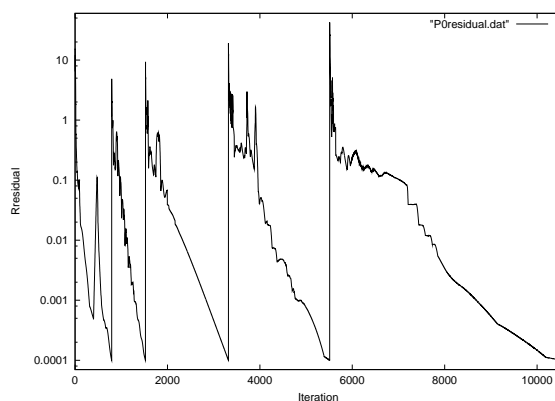


Fig. 5: Representative residual convergence history.

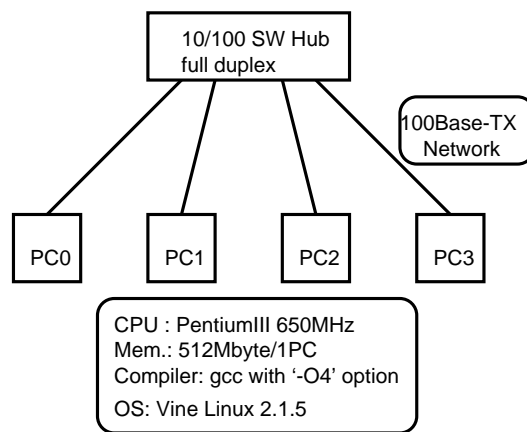


Fig. 6: Environment: PC cluster

細分化のタイミングは、計算誤差が収束した時に行なうこととした。Fig.5で、計算空間での計算誤差の最大値の収束履歴を示した。この場合、計算誤差が 10^{-4} になった時に格子細分化を行っている。

また、Fig.2では、正方物体周りの格子生成の様子を示している。初期格子の細分化レベルを0とした場合、この計算での最大細分化レベルは5である。これは、Fig.5で、5515ステップから10530ステップまでの間に計算が行われた格子の状態である。また、Fig.3では、非AMR (計算初期の段階で物体周りにだけ lev.1 の格子は張ってある) での密度を色で示したものと、Fig.3ではAMRを用いたもの (計算状態とし

ては、Fig.2と同じ)を示した。AMRを用いる事により、詳細な流れ場の様子が計算できている事がわかる。

3. 並列計算

並列計算について、計算環境、計算手法について触れ、またその結果について述べる。

3.1 計算環境

計算環境には、PCを4台利用したクラスタリングシステムを用いた。環境構成はFig.6に示した通りである。本システムの性能の目安として姫野ベンチマークテストプログラム⁷を走らせたところ、SMALLサイズの場合で約225MFlopsの結果が得られた。

3.2 手法

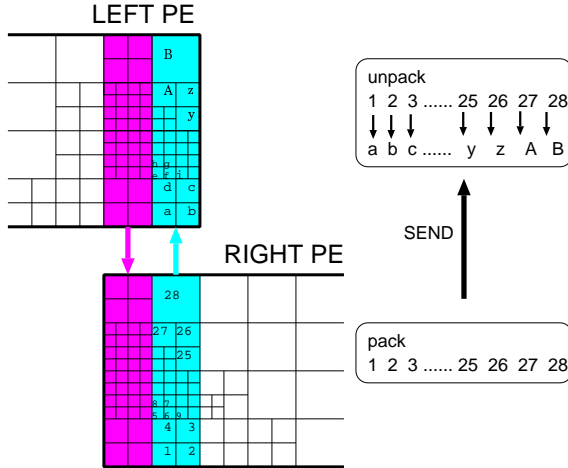


Fig. 7: Message Passing: Pack & Unpack data

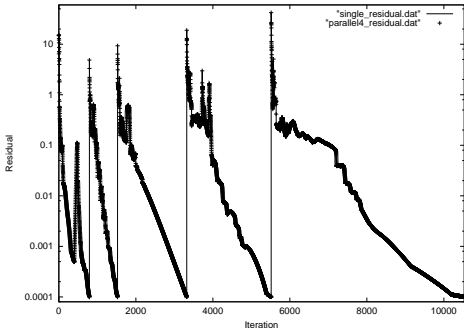


Fig. 8: Representative residual convergence history.(Parallel case)

並列計算は、領域分割法を用いることで行なう。領域分割は、level0格子の辺に沿って、 $x = const.$ のラインで分割するとする。それぞれのPEの計算領域に通信バッファ領域を設け必要データを通信しあう。バッファ領域は、左右にlev.0格子一つ分とした。送受信が必要なデータは、セル毎の構造体になっている為、まずはじめにMPI_Pack, MPI_Unpackを用いて対象セルをまとめる事を行う。その後、実際に送受信する事を行う。本手法では、2.2節で述べたデータの中で、物理量(4)、物理量一時データ(4)、時間刻み、勾配(8)、物理量 4×2 方向)、線形補間制限値、を通信を必要とするデータとして、PE間でデータ交換を行っている。

Fig.7では、具体的なデータのバックの方法を示した。LEFT PEの水色部分の格子は、RIGHT PEの水色部分と同じセル構成になっているとする。この図のRIGHT PEの水色の部分をLEFT PEの水色部分に送信する場合について考える。Packをしていく順番はlev0格子の単位で考え、その子供を再帰的にPackしていく。Packされたデータを、LEFT PE

では同様の手順でUnpackする。図中a,b,c,...には、実際のセルデータが入っている構造体へのポインタが入っており、それに従い、受信領域にあるセルデータを変更する事になる。

この手法を用いて計算を行った流れ場に関する残差の履歴をFig.8に示した。この図で、点で示したものが並列計算した場合であり、線で示したものが非並列手法で解いた場合のものを表している。両者は完全に一致している為、精度を落とす事なく、完全に並列計算が行われていると考えられる。

4. 結果

並列計算における、パフォーマンス評価を行う。ここでは、計算領域を単純にPC数で、x方向で分割した場合について検討する。Tab.1に、PC数を増やした場合の、1PCあたりの全計算時間(Whole Time)、通信時間(MP Time)と全体計算時間にしめる通信時間の割合(MP rate)を示した。この表によると、PCを増やす事により全体の計算時間が減り、並列計算の効果が現れている事がわかる。

	Whole Time	MP Time	MP rate
	Ave. (sec.)	Ave.(sec.)	Ave.(%)
1PC	38,937	-	-
2PCs	25,868	415	1.6
4PCs	14,111	536	3.8

Tab. 1: A Parallel Calc. Time

Fig.9には、Tab.1に基づいて、速度向上比をプロットした。2PCを利用した場合で、1PCの場合の1.51倍、4PCを利用した場合で2.75倍の結果が得られた。

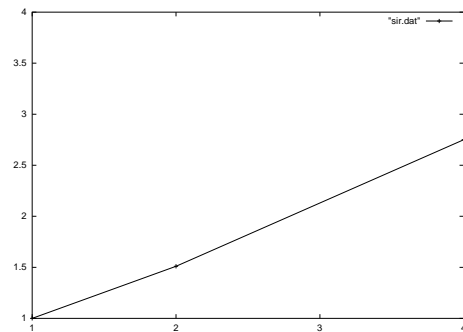


Fig. 9: Speed Improvement Ratio

	Term	Steps
T1	0 ~ 793	794
T2	794 ~ 1,524	731
T3	1,525 ~ 3,321	1,797
T4	3,322 ~ 5,514	2,193
T5	5,515 ~ 10,530	5,016

Tab. 2: A Term from refining to refining

4PC利用時には、4倍程度のパフォーマンスが得られる事が理想であるが、本手法ではそれに近い値を得る事が出来なかった。その原因として、適応格子法における格子数の局所的増加が考えられる。そこで、各PCの受け持つ格子数と計算ステップの関係をTab.3に示した。これは、各PCがT1~T5までの区間分けをした場合、その時々どの程度の格子数を各PCが担当していたかを示すものである。T1~T5は、その時点での格子数で計算を行った計算ステップ期間を示す。具体的には、Tab.2で示した通りで、T1は計算を始めてから793stepまでの794ステップ、T2は794stepから1,524の

731 ステップを計算した事を意味している。またこれは、Fig.8 での残差が既定値まで達する区間を表しているともいえる。以上のデータを再検討すると、ステップ数は、計算が進むにつれ長いものとなり、さらに、担当格子数も PC 毎に大きく異なる結果となった。具体的には、4PC を利用した場合、T5 の 5,016 ステップの間、PC0 は 2,600 の格子を計算し、PC2 は 10,973 の格子を計算していた事になる。この様に、計算のばらつきが非常に大きくなることで計算期間も長くなる為、並列パフォーマンスがそれほど上がらない事が言える。

	T1	T2	T3	T4	T5
1PC					
PC0	10,032	10,806	13,023	17,610	26,787
2PCs					
PC0	5,120	5,282	5,687	6,385	7,460
PC1	5,120	5,753	7,613	11,579	19,709
4PCs					
PC0	2,600	2,600	2,600	2,600	2,600
PC1	2,720	2,882	3,287	3,965	5,060
PC2	2,720	3,065	4,061	6,299	10,973
PC3	2,600	2,912	3,848	5,720	9,464

Tab. 3: A Number of Cells at each PC

5. おわりに

適応格子法に対して並列化を行い、その評価を行った。計算領域を単純に分割するだけでは負荷のばらつきが原因で、ある程度のパフォーマンスしか得る事が出来なかった。今後は動的な負荷分散を考慮しつつ、さらに PE 数の多いシステムで、ネットワーク性能の影響も考慮した手法を検討していく必要があるだろう。

参考文献

1. 姫野”自動車空力の数値シミュレーション”, 第 11 回数値流体力学シンポジウム, pp31-pp32(1997)
2. 古山彰一、松澤照男、”動的負荷分散を用いた適応格子法の並列計算”、情報処理学会論文誌 38 巻 9 号、平成 9 年 9 月号、pp.1869-1877
3. M.J.Berger, J.Oliger, “Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations”, J.of Comp.Phys. 53, pp484-pp512(1984)
4. C.P.T.Groth, D.L.DeZeeuw, T.I.Gombosi, “A Parallel Adaptive 3D MHD Scheme for Modeling Coronal and Solar Wind Plasma Flows”, Space Science Reviews 87, pp.193-pp.198(1999)
5. D.F.Martin, P.Collela, “A Cell-Centered Adaptive Projection Method for the Incompressible Euler Equations”, J.of Comp.Phys. 163, pp271-pp312(2000)
6. D.D.Zeeuw, K.G.Powell, ”An Adaptively Refined Cartesian Mesh Solver for the Euler Equations”, J.of Comp.Phys. 104, pp56-pp68(1993)
7. 姫野ベンチマークテスト,
<http://w3cic.riken.go.jp/HPC/HimenoBMT/index.html>