

数値計算における疎密のある物理空間に対応する マッピング：ローリングマッピング

村田 淳[†] 天野 英 晴^{††}

偏微分方程式を差分法（陽解法）で計算する場合には、離散化された物理空間の格子点に疎密をつけることにより、部分的に精度を上げしかも全体の計算量を少なくすることができる。このように離散化された物理空間を「疎密のある物理空間」と呼ぶ。均一に離散化された物理空間に対して、複数の計算要素（Processing Unit: PU）を相互に格子状に接続した PU アレイを持つ格子状接続並列計算機で計算すると、効率のよい計算が可能である。本論文では、格子状接続並列計算機で、疎密のある物理空間を解析する場合の効率的なマッピング（計算機の PU と物理空間の格子点の対応関係）方法であるローリングマッピングを提案する。この方法を用いることにより、一定の条件で物理空間を離散化すれば、疎密のある物理空間の格子点に対して、PU アレイの各 PU が均一の負荷で計算することができるようになった。また一般的に用いられるモジュラマッピングと比較して、PU アレイにサイクリック構造を要求しないことから、PU 間の通信量を少なくすることができた。さらにローリングマッピングでは、一部に遠隔 PU 間でのデータ交換が必要となるが、このデータ交換を高速に行うための、リンクの付加方法を提案した。

Rolling Mapping of Physical Regions for Nearest Neighbor Connected Machines

ATSUSHI MURATA[†] and HIDEHARU AMANO^{††}

For solving Partial Differential Equations (PDEs), it is necessary to divide the physical region into a certain size of grid. In order to achieve a high degree of accuracy with a reasonable computation cost, the density of grid is often partially changed. We call this type grid the grid with various density. For this type of calculation, parallel machines in which Processing Units (PUs) are connected with a nearest neighbor mesh (NNM) structure are suitable. Since the number of grid is much greater than the number of PUs, processes corresponding to grid points are mapped into a PU. In this paper, we propose a new mapping algorithm called the Rolling Mapping which reduces the communication for the grid with various density without cyclic links. The scheduling method for overlapping data exchange with calculating is also proposed and evaluated. Finally, a new network topology called rolling mesh dedicated for the rolling mapping is discussed.

1. ま え が き

偏微分方程式を差分法で離散化し、陽解法により数値計算する場合、一般的には対象とする物理空間を格子状に分割し、各格子点で代表される物理量を差分式に代入して計算する。

この場合、格子点間隔が小さいと、高い計算精度が得られるが、計算量は大きくなる。逆に計算量を減らすために格子点間隔を大きくすると、精度が犠牲にな

る。したがって、通常は必要な精度が維持できる程度に、格子点間隔を一様に小さくして計算する。

しかし、この場合、一部の領域についてのみ高い精度が要求される場合でも、全体にわたって小さい格子点間隔で均一に離散化するため、物理空間中の格子点数が過大になり、計算量が膨大になる。そこで、物理空間全体を均一に離散化するのではなく、注目したい部分や精度が要求される部分では格子点間隔を小さくして計算精度を上げ、計算精度が比較的要求されない部分は格子点間隔を大きくすることで、全体の計算精度と計算速度のバランスをとる方法が用いられている²⁾。

図 1 はこの方法を用いて離散化された物理空間の一例である。基本的な格子（格子点間隔が大）の一部分に、間隔を小さくした格子（格子点間隔が小）が内挿

[†] 京都府立大学生活科学部
Faculty of Living Science, Kyoto Prefectural University

^{††} 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

されている。このように正方格子の形を保ちながらも部分的に格子点間隔を変化させている離散化された物理空間を、本論文では「疎密のある物理空間」と呼ぶ。

離散化された物理空間（正方格子点空間）での計算には、複数の計算要素（Processing Unit, 以下PUと呼ぶ）を相互に格子状に接続したPUアレイを持つ格子状接続並列計算機で計算すると、効率よく高速に計算できることが知られている¹⁾。一般的に、対象となる物理空間の格子点数はPUアレイのPU数よりはるかに多いため、PUと物理空間の格子点を対応づけるマッピングが必要となる。この場合、最も簡単な方法は直接マッピングである²⁾。直接マッピングは、物理空間中の互いに隣接したいくつかの格子点の集合を1つのPUに対応させる。しかし直接マッピングを図1のような疎密のある物理空間に用いると、1つのPUが受け持つ格子点数に差が出て、各PUの計算負荷にばらつきが生じる。

この問題を解決するために、モジュラマッピング³⁾が提案された。しかし、この方法は計算負荷にはばらつきは生じないが、PUアレイにサイクリックな構造が要求される上、疎密境界部分での遠隔PU間交信が増大する問題点がある。

そこで、本論文では、PUアレイにサイクリックな構造を要求せず、疎密境界領域上での遠隔PU間交信も少ないマッピング法、ローリングマッピングを提案する。次に、ローリングマッピングを用いた場合に必要となる遠隔PU間交信を、計算とオーバーラップさせるためのスケジュール法を提案し、評価する。最後に、ローリングマッピング用の付加リンクをつけた結合網であるローリングメッシュについて述べる。

2. ローリングマッピング

2.1 モジュラマッピングとその問題点

格子状接続並列計算機のPUに物理空間の格子点座

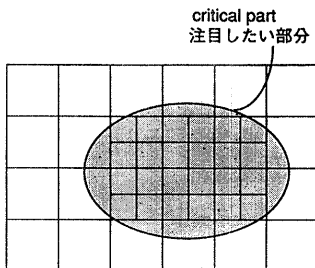


図1 疎密のある物理空間
Fig. 1 Physical region with double density.

標を割り当てる最も簡単な方法は直接マッピングである。この方法は、物理空間の格子点座標 (X, Y) に対して、PUアレイ（サイズ $m \times m$ ）のPU番号 (p, q) を

$$\begin{aligned} p &= \lfloor X/m \rfloor \\ q &= \lfloor Y/m \rfloor \end{aligned} \quad (1)$$

のように割り付ける方法である。この方法は単純であるため、均一の格子ではよく用いられるが、疎密のある格子に対して用いると、密な格子に割り当てられたPUの負荷が大きくなり、効率が低下する。

これに対し、モジュラマッピング³⁾は、物理空間の格子点座標 (X, Y) とPUアレイ（サイズ $m \times m$ ）のPU番号 (p, q) の対応を

$$\begin{aligned} p &= \text{mod}(X, m) \\ q &= \text{mod}(Y, m) \end{aligned} \quad (2)$$

で関係づける方法である。格子点数 8×8 の物理空間に対してサイズ 4×4 のPUアレイをモジュラマッピングした様子を図2に示す。

モジュラマッピングは、物理空間中でPUアレイをちょうどスタンプを押すように隙間なく並べてマップする。図2中の三角形記号はPUアレイの「向き」（配置方向）を表しているが、モジュラマッピングでは図のようにPUアレイが向きを変えることなく平行移動して物理空間に並んでいる。

この方法では、以下の条件の下に、疎密のある物理空間でも各PUの負荷が均一になるようにPUを割り付けることができる^{2),6)}。

1. 物理空間に疎密をつけるときには、疎の部分も密の部分も正方格子状に離散化する。
2. 疎の部分に対して密の部分の格子点密度は4倍（格子点間隔が1/2）とする。

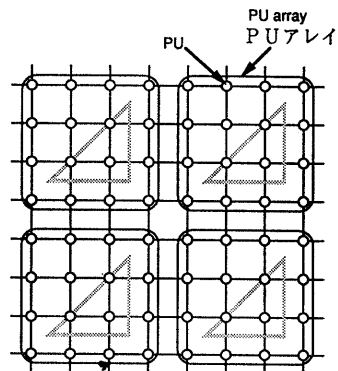


図2 モジュラマッピング
Fig. 2 Modular mapping.

なお、ここでは簡単のために格子点密度は疎の部分（以下、単密という）と密の部分（以下、倍密という）の2種類とする。実際には再帰的に密度を高くしていくことができるが、ここでは隣合う2つの部分空間の密度の比は4倍とする。

3. 疎密の基本単位を PU アレイサイズ (= $m \times m$) と同一とする。

これらの条件の下で、図3(a)で示した物理空間に疎密をつけると図3(b)のようになる。図3(a)の左上4分の1が倍密になっていて4つのPUアレイが担当している。

モジュラマッピングの問題点は、通信コストの増大を招くことである。まず、均一な物理空間に適用した場合でも、PUアレイの両端を接続しサイクリックにする必要がある。このような構造の格子はトーラスと呼ばれ、かつてはPAX³⁾をはじめとする多くの格子状接続計算機で用いられていた。しかし、最近のリンク

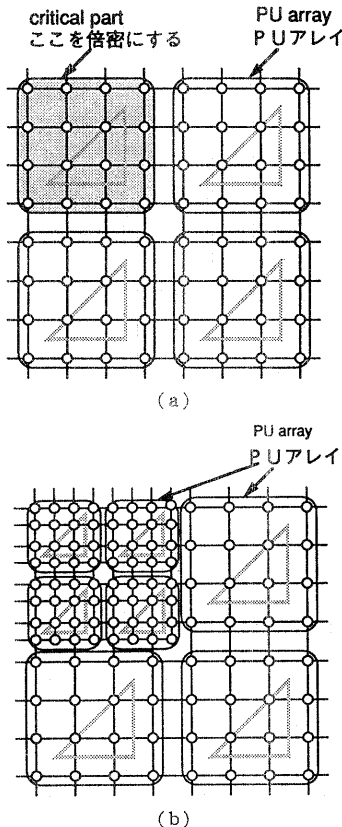


図3 疎密のある物理空間に対するモジュラマッピング
Fig. 3 Modular mapping with critical part.

結合型の並列計算機は、リンクの転送速度を上げるため、遠隔通信が必要なトーラス構造を避け、単なる二次元、三次元の格子状接続をとる傾向にある。Paragon³⁾, J-machine⁴⁾はこの例である。このようなマシン上ではモジュラマッピングは不利である。

さらに、モジュラマッピングを疎密のある物理空間に適用すると遠隔PU間でのデータ交換が必要となる。図4に示した疎密のある物理空間の疎密境界を部分的に拡大すると図5(上)の図になる。この図の中で←→で示された隣接格子点間のデータ交換は、実際のPUアレイでは図5の下の図に示すように、サイクリックリンクを用いた通信とPUアレイの同じ辺上に位置する離れたPU間の通信の合計2パスの通信によって行われる必要がある。

2.2 ローリングマッピング

以上の問題点を解決するため、われわれはモジュラマッピングと同様に疎密のある物理空間に適用した場合に負荷の不均一を生ぜず、しかもPUアレイにサイクリック構造を必要としないマッピング方法を考案した。この方法は、物理空間の格子点座標 (X, Y) とPU

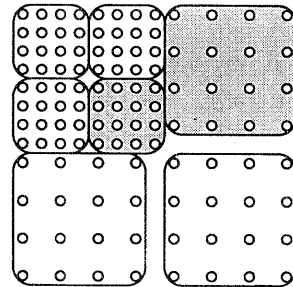


図4 疎密のある格子点
Fig. 4 Lattice points with double density.

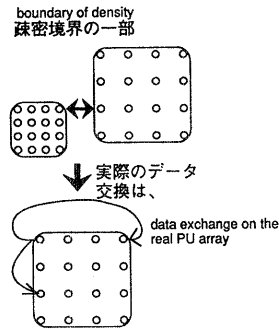


図5 疎密境界上のデータ交換
Fig. 5 Data exchange on the boundary of different density.

アレイ (サイズ $m \times m$) の PU 番号 (p, q) の対応を以下のように定義したものである。

$$\begin{aligned}
 P &= \text{mod}(X, 2m) \\
 p &= P & (P < m) \\
 p &= -P + 2m - 1 & (P \geq m) \\
 Q &= \text{mod}(Y, 2m) \\
 q &= Q & (Q < m) \\
 q &= -Q + 2m - 1 & (Q \geq m)
 \end{aligned} \tag{3}$$

このマッピング法では、モジュラマッピングと同様、個々の PU が担当する複数の格子点が物理空間中で分散されるため、疎密のある物理空間に適用した場合でも負荷の不均衡を生じない。

格子点数 8×8 の物理空間に対してサイズ 4×4 の PU アレイをマッピングした例を図 6 に示す。モジュラマッピングが、物理空間中を PU アレイが平行移動して埋めていくイメージであるのに対して、このマッピング法は、図 6 のように PU アレイを“くるりと”裏返して物理空間を埋めていくイメージである。このイメージからわれわれは式 (3) に定義したマッピング法をローリングマッピングと呼ぶことにする。

モジュラマッピングと比べ、ローリングマッピングは PU アレイをひっくり返すときの線対称軸に相対する格子点は同じ PU が担当することから、PU アレイ構造をサイクリックにする必要がない。

2.3 疎密のある物理空間に対するローリングマッピング

ローリングマッピングでも、モジュラマッピングと同様、一定の条件の下で疎密のある物理空間に PU を均一の負荷となるようにマップすることができるはずである。モジュラマッピングの場合は、式 (2) ($p =$

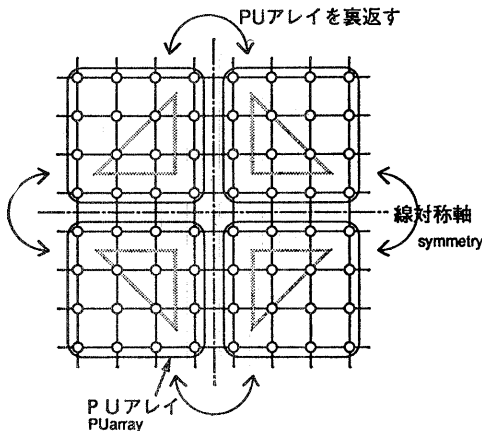


図 6 ローリングマッピング
Fig. 6 Rolling mapping.

$\text{mod}(X, m)$) で示したように大きさ $m \times m$ の繰り返しパターンで物理空間をマップしている。したがって疎密の基本単位は、モジュラマッピングの条件 3. (2.1 節参照) で記したように $m \times m$ であった。

ここで仮にモジュラマッピングのときの条件で、ローリングマッピングを疎密のある物理空間に適用してみる。図 7 (a) で示した物理空間の中で (条件 3. で疎密の基本単位を PU アレイサイズと同じ大きさとしたから) 図中の網かけの部分に倍密にできるはずである。これを実際に行ったものが図 7 (b) であるが、この図を見てもわかるとおり、疎密境界に相対する格子点を担当する PU がアレイの異なる辺 (アレイの反対側の辺) 上に位置することになる。この状態をここでは「不整合」と呼ぶ。不整合が起これるとローリングマッピングの利点である“PU アレイにサイクリック構造を必要としない”が成立しない。

そこで、不整合を避けるために、条件をモジュラマッピングの場合よりやや厳しくする必要がある。すなわち、疎密の基本単位を PU アレイサイズの 4 倍に定める。なぜならローリングマッピングの場合は、式 (3) ($P = \text{mod}(X, 2m)$) で示したように大きさ $2m \times 2m$ の繰り返しパターンで物理空間をマップしているためである。これで不整合は起こらなくなる。

以上の条件で疎密のある物理空間にローリングマッピングを適用した例を図 8 に示す。図 8 ではサイズ 4

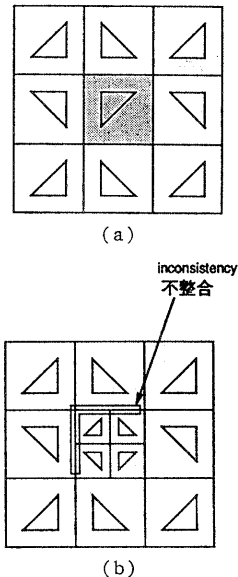


図 7 ローリングマッピングにおける不整合
Fig. 7 Rolling mapping with inconsistency.

×4のPUアレイでマッピングしている。図8の格子点を省略して描くと図9のようになる。さらに図9の疎密境界上のPUの様子を拡大して描くと図10のようになる。図10を見ると、疎密境界に相対する格子点は同じPUが担当するのではないが、PUアレイの同じ辺上に位置する離れたPUが担当することがわかる。したがって、ローリングマッピングを疎密のある物理空間に適用する場合には、PUアレイの同じ辺上に位置する離れたPU間で互いにデータ交換する必要がある。この辺は、PUアレイの4辺のうちの2辺だけである(図11参照)。

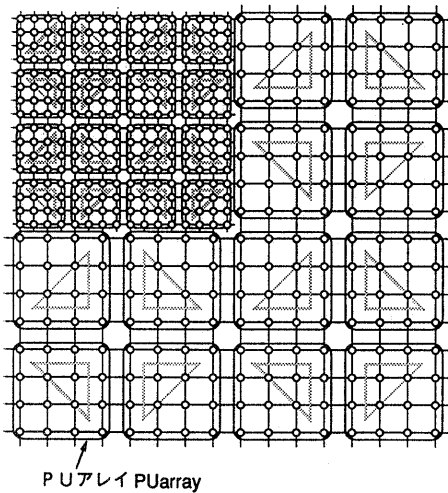


図8 疎密のある物理空間に対するローリングマッピング

Fig. 8 Rolling mapping on physical region with double density.

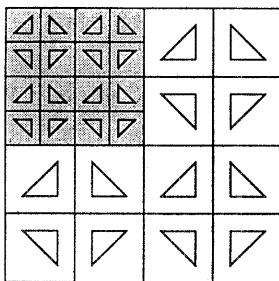


図9 空間ステップに差のある物理空間に対するローリングマッピング

Fig. 9 The direction of the PU array.

3. 疎密境界上のデータ交換のスケジューリング

3.1 スケジューリングの必要性

ローリングマッピング、モジュラマッピングの双方について、疎密境界上でのデータ交換のためには、PUアレイの同じ辺上の離れたPU間でのデータ交換が必要になる。離れたPUへのデータ交換は、ルーティングを伴うため一定の時間を要する。しかし最近のリンク結合型の並列計算機の多くは専用のルータを持つため、データのルーティングをPUでの計算と並行して行うことができる。このことを利用し、PUの受け持つ格子点に関する計算の順番を工夫することにより、遠隔PU間のデータ交換のルーティングを、その他の計算とオーバーラップさせ、ルーティングの時間を低減または隠蔽することができる。

以下、計算の順序とルーティングのオーバーヘッドが隠蔽できる条件について考察する。

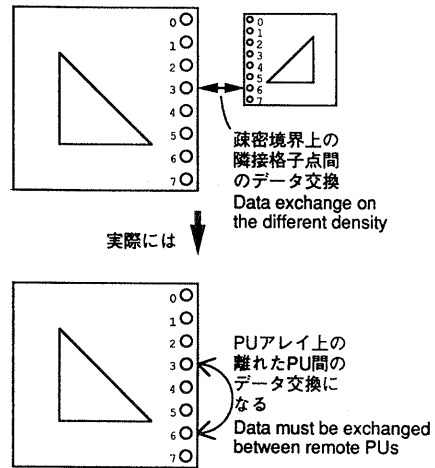


図10 疎密境界上のデータ交換

Fig. 10 Data exchange on the boundary of different density.

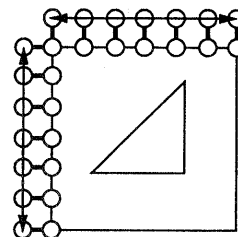


図11 辺上の離れたPU間のデータ交換が必要
Fig. 11 Data exchange between remote PUs.

3.2 PU アレイの計算順番と条件式

図 12 は、疎密境界の最小単位である。この2つの単密側部分空間と4つの倍密側部分空間のパターンを一単位として疎密境界が続いていく。この一単位の疎密境界の長さを「単位疎密境界長」と呼ぶ。ここで、図12で示した疎密境界のデータ交換に要する時間を、単密から倍密方向を C_{SD} 、倍密から単密方向を C_{DS} とする。この C_{SD} と C_{DS} の実際の時間は、データ交換でどのようなルーティングを行うかにより異なってくるが、一般的にはPU アレイサイズに比例した時間になると考えられる。

ここで、PU アレイサイズと同じ大きさの部分空間を計算する時間、すなわち各PU が格子点1つ分の計算を行う時間を T_0 とする。 T_0 に比べ C_{SD} や C_{DS} が小さければ、各PU が1つの格子点に関する計算を行っている間に疎密境界のデータ交換を実現することができるため、全く問題は生じない。しかしPU の計算速度の向上から、現実的には、 C_{SD} 、 C_{DS} は T_0 に比べ大きくなる。そこで、疎密境界に関係しない格子点を各PU が計算している間に、疎密境界のデータ交換を行うことができるように、以下の順に計算を行う。

1. 最初に、疎密境界に接する単密部分を PU アレイが計算し始める。
2. 計算が終わった単密の部分空間から「単密→倍密」のデータ交換を始める。
3. 次に、疎密境界に接する倍密部分を PU アレイが計算し始める。
4. 「単密→倍密」のデータ交換がすべて終了した後、計算が終了した倍密の部分空間から「倍密→単密」のデータ交換を始める。
5. 最後に疎密境界に接していない単密と倍密の部分空間を計算する。

これをタイミングチャートで示すと図 13 のようになる。

3.3 データ交換に許される時間

上記の1.~5. の計算順番と図 13 のタイミングチャートより、PU アレイが部分空間を計算している間に疎密境界のデータ交換を完了する条件を考える。

1つの物理空間では、疎密境界のデータ交換にかかる時間は、疎密境界長に比例する。単位疎密境界には、単密側に2個、倍密側に4個で計6個の単位部分空間がある。この6個の単位部分空間をPU アレイが計算している間にデー

タ交換を終わらせるには以下の条件を満足する必要がある。

[1つの疎密境界のデータ交換時間は $C_{SD}+C_{DS}$ 、部分空間1つをPU アレイが計算する時間は T_0 だから]

$$C_{SD}+C_{DS} \leq 2 \times T_0 + 4 \times T_0$$

より

$$C_{SD}+C_{DS} \leq 6 \times T_0 \tag{4}$$

つまり

$$1 \text{つのデータ交換時間} \leq 6 \times 1 \text{つの部分空間計算時間} \tag{5}$$

となる。

実際には、データ交換と“疎密境界に関係ない部分空間”の計算は、同時並行的に行うことができる。したがって式(4)より

$$C_{SD}+C_{DS} \leq 6 \times T_0 + \frac{k_N \times T_0}{\text{全疎密境界長}} \tag{6}$$

k_N : 疎密境界に接しない部分空間数
 $k_N \times T_0$: 疎密境界に関係ない部分空間をPU アレ

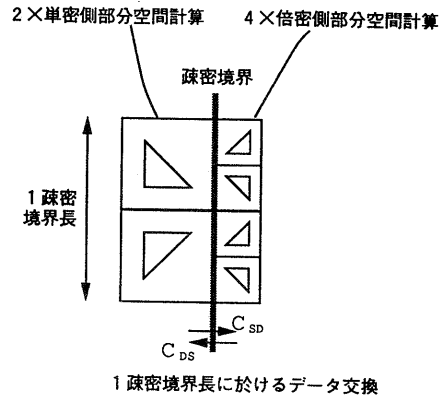
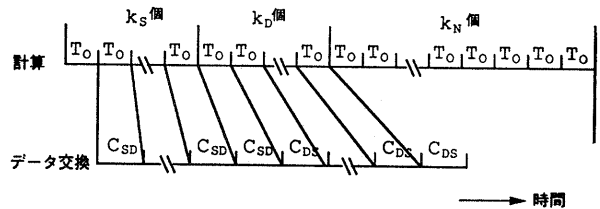


図 12 データ交換に要する時間

Fig. 12 The time required for data exchange.



k_S : 境界に接する単密の部分空間数
 k_D : 境界に接する倍密の部分空間数
 k_N : 境界に接しない部分空間数
 T_0 : PUアレイが1つの部分空間を計算する時間

図 13 タイミングチャート

Fig. 13 Timing chart.

イが計算する時間

以上の条件が満足されれば、データ転送時間は、計算時間とオーバーラップされ、性能の低下は生じない。

3.4 時間ステップが空間ステップと共に変化する場合のスケジューリング

前節までは、物理空間中の単密部分も倍密部分も時間方向の計算ステップ（時間ステップの大きさ=Δt）は同一であるとした。しかし、場合によっては、空間ステップの大きさによって時間ステップの大きさも変えることも考えられる。例えば、拡散方程式を差分法（陽解法）で解析する場合には、

時間ステップ/(空間ステップ)² が一定値以下という条件（安定条件）が要求される⁷⁾。この場合、疎密のある物理空間で空間ステップの大きさが変われば、それに従って時間ステップの大きさも変える必要がある場合もある。具体的には、単密部分が1時間ステップ進むごとに倍密部分が4時間ステップ進む、という方法が一般的である。これを図示すると図14のようになる。ただし、このように倍密部分の時間ステップを単密部分の時間ステップより多くすると、当然物理空間全体を計算する時間（総計算時間）は長くなる。単密部分1ステップに対し、倍密部分4ステップだとすると、同じ面積に対する計算時間の比率は、単密：倍密が1：16になる。しかし、物理空間全体を倍密にするよりは、はるかに少ない計算時間で済むことになる。

ここでは、図14のように

単密部分の計算ステップ：

$$\text{倍密部分の計算ステップ} = 1 : 4 \quad (7)$$

である場合を考える。また格子点の計算は陽解法を想定している。

この場合「倍密→単密」のデータ交換には、図15の1)と2)のように、最低2種類が考えられる⁵⁾。図

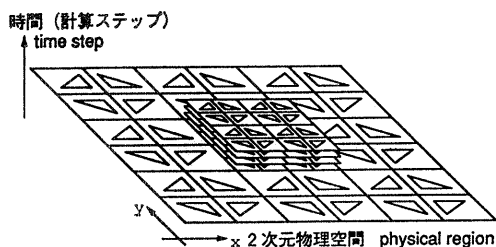


図14 単密の計算ステップ：倍密の計算ステップ=1：4の場合

Fig. 14 Time step on single density: Time step on double density=1: 4.

の1)は、単密側の格子点の計算に、倍密側の“一番古い”計算結果を用いている。一方、2)は、単密側の格子点の計算に、倍密側の“最新の”計算結果を用いている*。

PUアレイの計算のスケジューリングから考えてみると、1)の方が2)より良い。なぜなら1)は、倍密側の計算ステップ0の終了直後に「倍密→単密」のデータ交換を進めることができる。つまり【倍密→単密のデータ交換】と【1~4の計算ステップ】を独立に進めることができるからである。一方、2)は、「倍密→単密」のデータ交換を行うには、倍密側の計算ステップ3を先に終了しておく必要がある。さらにそのためには、「単密→倍密」のデータ交換を済ませた後、1, 2, 3の計算ステップの計算を済ませておく必要がある。

このような、倍密部分が4計算ステップ必要の場合でも、図15の1)の方法を採用するならば、疎密境界のデータ交換の最中に計算できるのは、倍密部分の第1ステップのみである。つまり、計算ステップが単密と倍密で同じように進む場合（3.3節の場合）と同じである。倍密部分の2, 3, 4ステップ目は、疎密境界のデータ交換が完全に済んでから計算することになる。

したがって、物理空間内で計算ステップに差がある場合でも、データ交換に許される時間は、式(6)と同

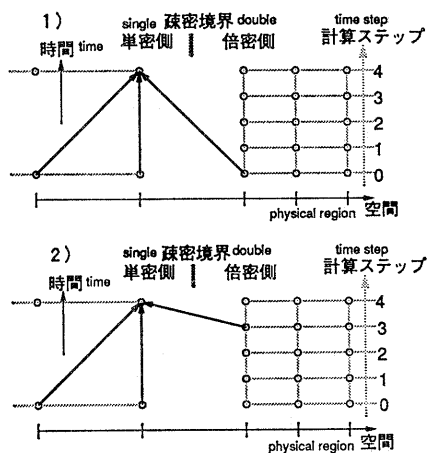


図15 疎密境界単密側格子点の隣接格子点の選び方の違い

Fig. 15 The choice of neighbor mesh point.

* 数値計算の精度上は、1)より2)の方が良いように思えるかもしれないが、実際は、1)も2)もあまり変わりがないという実験結果もある²⁾。

じとなる。

4. ローリングメッシュ

今までは、遠隔リンクを持たない平面メッシュ構造を持つ並列計算機に対して、疎密のある物理空間のマッピングおよび、交信コストを隠蔽するためのスケジューリング法について検討してきた。しかし、PUアレイのサイズが大きくなると、相対的にPUの受け持つ部分空間が小さくなり、式(6)に示す条件が成立せず、辺上での交信コストが全体の計算時間を支配する場合が考えられる。そこで、ここでは、対象とする並列計算機が遠隔リンクを持つことができる場合に、ローリングマッピングに適したリンクを加える方法を提案する。このローリングマッピング用の遠隔リンクを持つメッシュ結合をローリングメッシュと呼ぶ。

4.1 境界上の交信パターン

図16にローリングマッピングの疎密境界の最小パターンを示す。ここで、この疎密境界上で相対する格子点を担当するPUがどのように配置されているかを考える。一例として、 8×8 のPUアレイを考え、辺上に位置するPUに0から7までの番号をつける(図17)。図17の疎密境界を拡大し、疎密境界に相対する隣接格子点を担当するPU番号を示すと図18のようになる。図18中の、矢印でつながったPUが互いにデータ交換する必要がある。ここで、単密側のPU番号を n 、倍密側のPU番号を m としたときの m と n

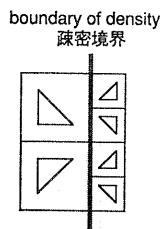


図16 疎密境界の単位

Fig. 16 Unit of the boundary of density.

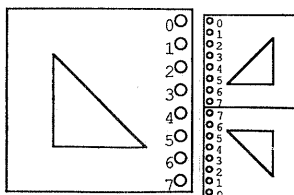


図17 疎密境界のPU番号

Fig. 17 PU numbers on the boundary.

の関係を図中に示す。

この関係式を横軸に n 、縦軸に m を取ってグラフにすると図19のようになる。この図により、縦方向、横方向共に n, m が決定すれば、それぞれ対応する m と n は2つに定まる。

図18中の矢印で示された関係に従って、PUアレイの辺上にリンクを付け加えた様子を図20に示す。図中では1つのPUから出るリンクの数は最大でも4である。これは、図19中に示すように、各 n に対応する m は2、各 m に対応する n は2であり合計すると、各 n, m に対応する m, n が4になるためである。以上はPUアレイが 8×8 の例であるが、これを容易に任意のサイズに一般化することができる。

4.2 $p \times p$ のPUアレイに対する付加リンク

PUアレイが $p \times p$ (p は自然数でPU番号はゼロオリジン $0 \sim p-1$) の場合で、 $m=f(n)$ を一般化する

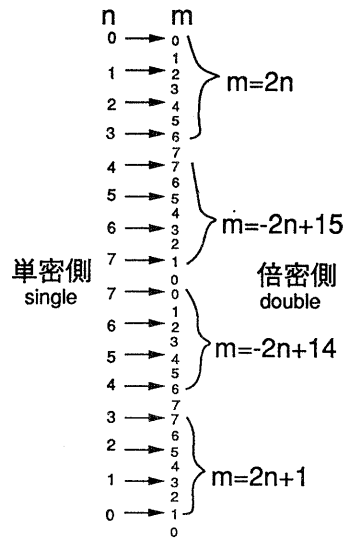


図18 疎密境界のPU番号の対応関係

Fig. 18 PU numbers of the boundary of density.

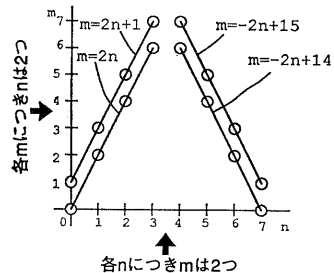


図19 各PUに要するリンクの数

Fig. 19 Links for each PU.

と

$$\begin{aligned}
 f(n) &= 2n+0 & (0 \leq n \leq \lfloor (p+1)/2 \rfloor - 1) \\
 f(n) &= -2n+(2p-1) & (\lfloor (p+1)/2 \rfloor \leq n \leq p-1) \\
 f(n) &= -2n+(2p-2) & (p-1 \geq n \geq \lfloor p/2 \rfloor) \\
 f(n) &= 2n+1 & (\lfloor p/2 - 1 \rfloor \geq n \geq 0)
 \end{aligned}
 \tag{8}$$

となる。ここで、 p が偶数の場合の $m=f(n)$ のグラフは図 19 と相似であり各 n, m に対応する m, n は 4 になる。

p が奇数の場合には、各 n に対応する m は 2つまたは 1つになり、各 m に対応する n は 2つまたは 1つである。したがって、PU のリンク数は 4 または 3 になる。

以上より、最大 4 のリンクを付加すること、疎密のある物理空間をローリングマッピングするための、必要な通信は実現することができる。

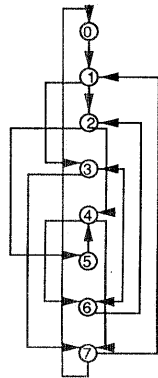


図 20 図 18 のリンクを実現するネットワーク
Fig. 20 Network for remote PUs.

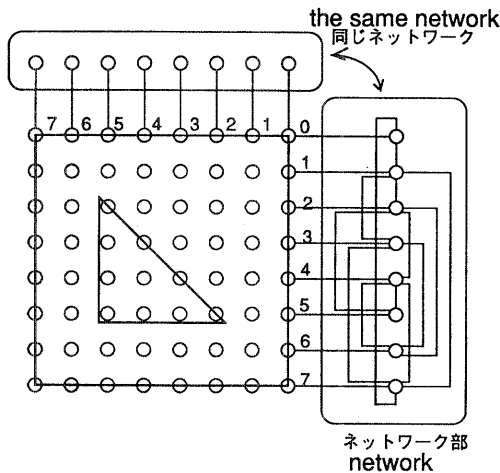


図 21 ローリングメッシュ
Fig. 21 Rolling mesh.

4.3 ローリングメッシュの構成

図 21 にローリングメッシュの概念図を示す。PU アレイの縁には、辺上の PU と同数の遠隔交信用ルータを設ける。このことにより、辺上の PU もアレイ内部と同様、4 本のリンクを持つことになる。ルータは 4 本の遠隔リンクを持ち、式(8)に従ってお互いに接続される。PU 番号 0 に直結するルータと PU 番号 7 に直結するルータ間のリンクは、PU アレイのサイクリックリンクで代用できるので、各ルータの持つ遠隔リンクは最大 4 である。ルータは 4×4 のクロスバから構成され、遠隔プロセッサの番号により、自動的にルーティングを行う。この構成により、疎密境界上の遠隔プロセッサ間の通信は、すべて 1 パスで行うことができる。

5. おわりに

本論文では、格子状接続並列計算機の計算要素と計算対象の物理空間の格子点との対応を関係づけるマッピングとして、ローリングマッピングを提案した。このマッピング法はモジュラマッピングと比べると、PU アレイにサイクリック構造を必要としない。さらに、一定の条件の下で物理空間に疎密を付けると、ローリングマッピングでは各 PU の負荷を均一にしてマップできることがわかった。また、遠隔 PU の通信を隠蔽するための計算順序と、隠蔽可能な条件について検討を行った。

最後に、疎密のある物理空間の疎密境界上のデータ交換を高速に行うためのリンクの付加方法を提案した。

今後の課題としては、遠隔 PU の通信用付加リンクの詳細な設計とシミュレーションによる評価を行う必要があると考えている。また、疎密のある物理空間によって応用的な物理問題を計算実験し、物理空間に疎密をつけた効果を評価することも必要である。

参考文献

- 1) 星野 力:PAX コンピュータ, オーム社(1985).
- 2) 村田 淳, 大沢 暁, 天野英晴, 相磯秀夫: 正方格子状接続並列計算機と疎密のある物理領域とのマッピングアルゴリズム, 第 34 回情報処理学会全国大会論文集, pp. 91-92 (1987).
- 3) Paragon XP/S Product Overview, Intel, 1991.
- 4) Dally, W. J., Chien, A., Fiske, S., Horwat, W., Kenn, J., Larivee, M., Lethin, R., Nuth, P. and Wills, S.: The J-Machine: A Fine-Grain Concurrent Computer, *Proc. IFIP 11th*

Computer Congress, pp. 1147-1153 (Aug. 1989).

- 5) 村田 淳, 天野英晴: 空間ステップと時間ステップに差をつけた格子をもつ物理空間に対する格子状接続並列計算機のマッピング, 第 43 回情報処理学会全国大会論文集 6, pp. 101-102 (1991).
- 6) Osawa, G., Murata, A., Amano, H. and Aiso, H.: Rolling Mapping for Physical Simulations, *Proc. of Second Intl. Conf. on Supercomputing* (1987).
- 7) 矢嶋信男, 野木達夫: 発展方程式の数値解析, 岩波書店 (1977).

(平成 4 年 12 月 24 日受付)

(平成 5 年 11 月 11 日採録)



村田 淳 (正会員)

1963 年 (昭和 38 年) 生. 1987 年慶應義塾大学理工学部電気工学科卒業. 1989 年同大学院電気工学専攻修士課程修了. 同年郵政省入省, 通信総合研究所および郵政研究所勤務.

1991 年郵政省辞職, 同年慶應義塾大学大学院計算機科学専攻後期博士課程入学. 1993 年京都府立大学生活科学部助手. 格子状接続並列計算機のアーキテクチャ, 並列計算機のマッピングアルゴリズムが主な研究テーマ. また, 並列計算機を使った数値計算にも興味を持つ. 日本物理学会, 日本応用数理学会各会員.



天野 英晴 (正会員)

1958 年生. 1986 年慶應義塾大学理工学部大学院博士課程修了. 工学博士. 並列計算機の研究に従事. 現在慶應義塾大学理工学部専任講師.

著書「誰にもわかるデジタル回路」(オーム社), 「並列処理機構第 5 章」(丸善).