ハイパフォーマンス
コンピューティング 82－4
（2000．8．3）

科学技術計算専用ロジック組み込み型プラットフォーム・アーキテクチャの開発

－ GSMAC有限要素法専用ロジックの検討 －

溝口大介§ 荒木健悟§ 石橋政一§ 佐々木徹§ 長嶋雲兵† 棚橋隆彦‡

§アプリオリ・マイクロシステムズ E-mail: {mizo, araki, ishibashi, sasaki}@a-priori.co.jp

†産業技術融合領域研究所 E-mail : umpei@nair.go.jp

‡慶應義塾大学 理工学部 機械工学科 E-mail : taka@mech.keio.ac.jp

概要

近年，自動車周りの流れ解析などに計算機を用いた数値解析が広く用いられるようになってきている．今後はさらに数値流体解析の重要性が増すと考えられ，計算コストが大きいという欠点を解決することが求められている．我々は科学技術計算用プラットフォームを利用し，数値流体解法の一つであるGSMAC有限要素法に基づく価格/性能比に優れた流体解析システムの作成を計画している．本稿ではGSMAC有限要素法のコアループについて，演算の並列性やデータ転送を調べ，高速化のための専用ロジックの基礎検討を行った．

A Platform Architecture for Embedded High-Performance Computing

－ Custom Logic for GSMAC-FEM －

Mizoguchi Daisuke§ Araki Kengo§ Ishibashi Masaichi§ Sasaki Tohru§

Nagashima Umpei† Tanahashi Takahiko‡

§A Priori Microsystems, Inc.

E-mail: {mizo, araki, ishibashi, sasaki}@a-priori.co.jp

†Electorotechnical Laboratry E-mail : umpei@nair.go.jp

‡ Department of Mechanical Engineering, Faculty of Science And Technology, Keio University E-mail : taka@mech.keio.ac.jp

Abstruct

Custom Logic for GSMAC-FEM(Generalized Simplified Marker And Cell Finite Element Method) is discussed. GSMAC is a kind of finite element method for solving unsteady incompressible Navier-Stokes equations. Generally, unsteady flow solver needs enormous computing time. Most time comsuming step in GSMAC is to solve Poisson equation $\nabla^2\phi = -\nabla\cdot v$. Thus, we first investigated operation-level parallelism and dataflow of this performance-critical step, and proposed custom logic to speed calculation.

1 はじめに

近年，計算機の高速化にともない数値流体解析が盛んに行なわれるようになってきている．流体の支配方程式の離散化手法としては有限差分法，有限要素法等がある．非構造格子に基いた有限要素法は複雑形状を容易に表現する事が可能で汎用性に優れているが，記憶容量や計算量の面で負荷が大きいという問題を抱えている．これを解決するために提案された非圧縮性流体の解法の一つとしてGSMAC FEM([1],[2])がある．この手法は各要素において離散化ナブラ演算子を定義し，要素係数行列の記憶量を低減している．しかし実際の応用現場においては益々大規模な計算を必要とするようになっており，さらなる計算の高速化が望ま

れている．そこで本稿では，GSMAC FEMを高速実行するための専用の計算機を提案する．

一般的に専用計算機の開発には多くの時間と労力を必要とする．そこで我々はコモディティ技術を最大限に利用し，最も計算負荷が大きい処理の高速化に注力できるようにするためのプラットフォーム([3]) の開発を目指している。本稿ではこのプラットフォームに組み込むGSMAC FEM専用ロジックの検討を行なう．

2 GSMAC FEMの概要

2.1 非圧縮性流体の支配方程式

非圧縮性流体の無次元化した支配方程式は次のように与えられる．

$$\nabla \cdot \boldsymbol{v} = 0 \quad (1)$$

$$\frac{\partial \boldsymbol{v}}{\partial t} = -\nabla H + \boldsymbol{v} \times \boldsymbol{\omega} + Re^{-1}\nabla^2 \boldsymbol{v} \quad (2)$$

ここでtは時間，$\boldsymbol{v}$は速度，$\boldsymbol{\omega}$は渦度，$\boldsymbol{H}$はベルヌーイ関数,Reはレイノルズ数である．式(1)は連続の式，式(2)はNavier-Stokes方程式と呼ばれる．

2.2 オイラー陽解法による予測子の計算

まず式(2)をオイラー陽解法で時間的に離散化し，次のように速度の予測子$\tilde{v}$を求める．

$$\tilde{v} = v^n + \Delta t(-\nabla H^n + v^n \times \omega^n + Re^{-1}\nabla^2 v^n) \quad (3)$$

ここで右肩に添えられたnは時間ステップnにおける物理量を表し，予測子の計算を行う時点で既知である．求められた予測速度は連続の式を満たすとは限らないため，速度とベルヌーイ関数に修正を施して$\nabla \cdot v^{n+1} = 0$を満たすようにしなければならない．

2.3 速度とベルヌーイ関数の同時緩和法

GSMAC FEMでは修正速度ポテンシャルϕを導入し，速度とベルヌーイ関数を同時緩和させる手法を採用している．この方法は非圧縮性流体の解法で非常によく用いられている．

$$\nabla^2 \phi = -\nabla \cdot \tilde{v} \quad (4)$$

$$v^{n+1} = \tilde{v} + \nabla\phi \quad (5)$$

$$H^{n+1} = H^n - \phi/\Delta t \quad (6)$$

よって問題は式(4)のポアソン方程式を解き，ϕを求める事に帰着する．GSMAC FEMではこのポアソン方程式を直接解く代わりに，優対角近似による逆ラプラス演算子を定義して，次のように修正速度ポテンシャルを近似的に求めて計算の簡略化を行っている．

$$\phi = (\nabla^2)^{-1}(-\nabla \cdot \tilde{v}) \quad (7)$$

対角成分だけを考慮しているので，逆ラプラス演算子を単純に乗じるだけであり，完全にelement-by-elementに処理する事が可能である．このように近似的にϕを求めて速度とベルヌーイ関数を修正する処理を，連続の式を満たす速度場が得られるまで繰り返す．このステップがGSMAC FEMにおいて最も計算負荷が大きいコアループとなる．この一連の処理をフローチャートで示すと図1のようになる．

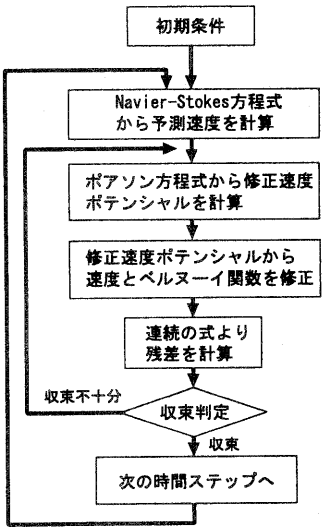

図 1: GSMAC FEMのフローチャート

2.4 離散化ナブラ演算子

空間微分のナブラ演算子∇を離散化したものが離散化ナブラ演算子であり，形状関数の勾配の要

素平均値として定義される．離散化ナブラ演算子によって $div, grad, rot$ の計算が定式化され，流体以外にも偏微分方程式に基づく様々な問題に応用する事が可能である．ここでは 6 面体 (8 節点) 双一次要素の場合についてコアループ内に現れる $div(\nabla\cdot)$ 演算を取り上げる．離散化ナブラ演算子 ∇_a は，計算空間と物理空間の間の変数変換に伴って現れる余因子ベクトル A^ξ, A^η, A^ζ から次のように計算できる．

$$
\begin{aligned}
\nabla_1 &= (-A^\xi - A^\eta - A^\zeta)/\Omega_e \\
\nabla_2 &= (+A^\xi - A^\eta - A^\zeta)/\Omega_e \\
\nabla_3 &= (+A^\xi + A^\eta - A^\zeta)/\Omega_e \\
\nabla_4 &= (-A^\xi + A^\eta - A^\zeta)/\Omega_e \\
\nabla_5 &= -\nabla_3 \\
\nabla_6 &= -\nabla_4 \\
\nabla_7 &= -\nabla_1 \\
\nabla_8 &= -\nabla_2
\end{aligned}
$$

ここで Ω_e は要素の体積である．後半の 4 つは 6 面体の対称性により求める事ができる．余因子ベクトル，及びそれらから作られる離散化ナブラ演算子は要素の形状にのみ依存するので，記憶容量と処理負荷のトレードオフにより，予め計算して記憶しておくか逐次生成するかを選択できる．この離散化ナブラ演算子を用いて速度場の div を計算すると

$$
div\ v = \sum_{a=1}^{8} \nabla_a v_a \qquad (8)
$$

のように定式化される．ここで v_a は各節点に定義された速度である．コアループ内ではこの計算を高速化する事が重要となる．

2.5 GSMAC FEM の計算例

実際の計算例として PentiumII 450Mhz の PC 上にて正方キャビティ内の強制対流計算を行った。計算条件は $Re = 100$，時間刻み $\Delta t = 0.01$，左右と下面は静止壁で，奥行き方向には周期境界条件を用いた．また上壁が急に動き出すインパルシブスタートを仮定している．メッシュは $16 \times 16 \times 8$，$32 \times 32 \times 8$，$64 \times 64 \times 8$ の 3 種類を用いた．繰り返し計算の収束判定は無次元化された $div\ v$ の最大値が 10^{-3} 以下になったときとした．図 (3) は時間ステップ 200 までのコアループの繰り返し計算回数をプロットしたものである．要素数が増えるにつれて収束までに要する繰り返し回数が増えている．また表 (1) は全体の計算時間に対するコアループの割合を示しており，計算時間の大部分を占めている事が分かる．この例では初期状態から急激に変化が起こり，その後定常状態へと至るまでの過渡状態であるため，収束までに要する繰り返し計算回数が多く，特にコアループの割合が顕著になっている．

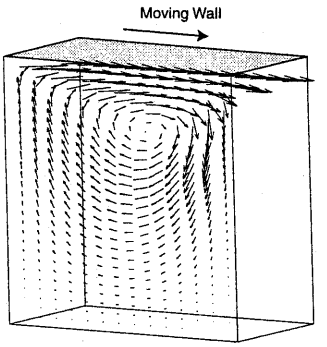

図 2: 計算結果の速度ベクトル

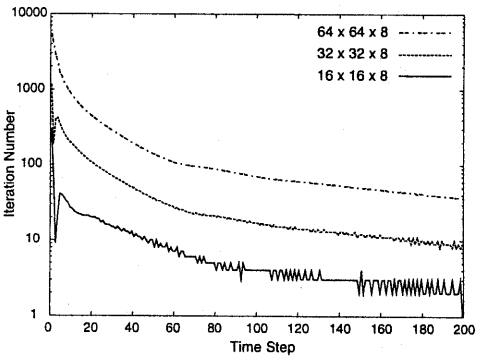

図 3: 各 TimeStep における繰り返し計算回数

要素数	時間の割合	平均ループ回数
$16 \times 16 \times 8$	65.7%	10.1
$32 \times 32 \times 8$	91.0%	47.8
$64 \times 64 \times 8$	97.7%	225.8

表 1: 正方キャビティ内の強制対流問題を解く際のコアループの占める割合

3 専用ロジックの検討

本節では「科学技術計算専用ロジック組み込み型プラットフォーム」に組み込むGSMAC FEMに特化したロジックの構成を検討する．まずこのプラットフォームについて簡単に説明する．

3.1 プラットホームについて

「科学技術計算専用ロジック組み込み型プラットフォーム」とは「各種アプリケーションに応じた専用ハードウエア・ロジックを組み込み可能とし，そして当該専用ロジックが提供する特定用途向け計算機能以外の共通基盤機能を提供する『専用システムの土台』」である．この実装は種々可能であるが，本研究では短期間・低価格で実現することを目的としているため，物理的には下記のような構成を前提としている．

- 専用ロジック → 専用 LSI
- プラットフォーム → プリント基板上の汎用 MPU(マイクロプロセッサ) + 業界標準バス/ネットワーク規格

このプラットフォームは一つの基板上に複数のプロセッサと分散型のメモリを搭載し，その基板をさらに複数枚用いてシステムが構成される．各基板には最低限 1 つの汎用 MPU を置き，基板内，また基板間の通信機能や，計算の制御といった複雑な作業を担当させる．よって専用 LSI は本来の役割である各アプリケーションに特化した最も負荷の高い計算にのみ専念すれば良く，開発の負荷を大幅に低減できる．GSMAC FEM は element-by-element に処理が可能で，領域分割法に基づく並列化 ([4]) が容易であるため，メモリ分散型の計算機との親和性にも優れている．この領域分割によって生じる通信については汎用 MPU が処理するため，専用 LSI の設計にあたっては自らが担当する領域内の計算についてのみ検討すればよい．

3.2 コアループ内演算の並列性

コアループは図 4 のように 3 つのループからなる．ループ (1) は要素についてのループであり，各要素毎に並列に処理することが可能である．ループ (3) の処理に関しては境界条件を含み，個々の問題依存となるため，柔軟性が必要となる．よって境界に属する節点に対しての処理については汎用 MPU が担当し，非境界節点についての単純な処理を専用ロジックに割り当てる等の対策を検討中である．ループ (3) については，ループ間でデータの依存関係が発生し，完全に並列化を行なうにはメモリ・プロセッサ間の構成法に工夫が必要である．

本稿では GSMAC FEM 専用ロジックの最初の検討として，比較的並列化が容易であるループ (1) に着目する．また要素は 8 節点の 6 面体要素として議論を進める．

```
// ループ(1)
for(IE=0; IE< 要素数 ; IE++){
   DIV = div v ;
   Φ [IE] = - DIV ×( ∇²)⁻¹;
   // 収束判定のために |DIV/Ωe| の最大値を保存
   DIVMAX = max(DIVMAX , abs(DIV/Ωe));
}
// ループ(2)
for(IE=0; IE< 要素数 ; IE++){
   //要素IEの各節点の速度の修正分を計算
   v'[1~8] = grad Φ[IE];
   //ベルヌーイ関数を修正
   H[IE] = H[IE] + Φ [IE]/ Δt ;
}
// ループ(3)
for(I=0; I< 節点数 ; I++){
   境界条件を考慮して各節点の速度を修正
}
```

図 4: コアループ内の処理

図 5 はループ (1) の主な処理である速度の $\boldsymbol{DIV}$ の計算の模式図である．ここでは離散化ナブラ演算子と各節点の速度はレジスタに格納されている事を前提としている．節点 1 から 8 についてそれぞれ内積を求め，$\boldsymbol{DIV}$ にアキュムレートしていく．これは 3 次元のベクトル演算であり，容易に分かるように各 x, y, z 成分は独立に積和演算可能

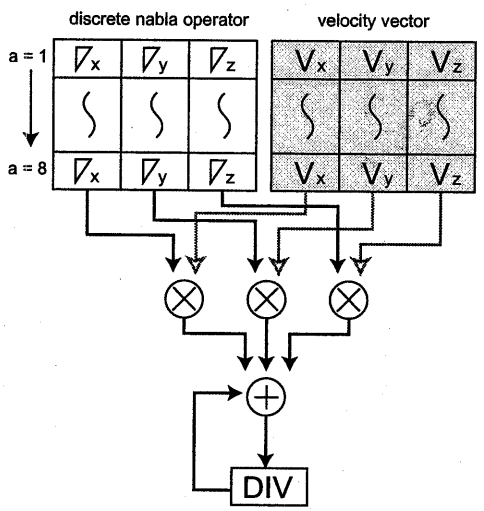

図 5: $DIV = divV$ の具体的な処理

である．よって演算能力が不足する場合は複数の演算器を用意して SIMD 演算方式による高速化を行うことができる．また分岐のない規則的な演算の繰り返しであるため，パイプライン化による高速化も容易である．

3.3 演算性能とデータ転送のバランス

2 項算術演算を 1 命令と見なした場合のループ (1) 中の総演算数は 50 であり，そのうち 47 命令が DIV を求める処理 (図 5) に費やされている．このため，DIV を求める処理の高速化が重要となる．DIV を求める処理は，離散化ナブラ演算子 24 個とそれに 1 対 1 で対応する速度をそれぞれ掛けて，その乗算結果 24 個をすべてを加算する，という処理である．

現時点では専用ロジックを製造するプロセス等は未定であり，面積の制約等が決定していないため，演算器の (個数・結線等の) 最適化については言及できない．しかし，ここで言えることは，DIVMAX の値一つを求めるために，48 個の変数の値を必要とするということである．必要とするデータが多いため，メモリからプロセッサにデータを転送する時間と演算時間とのバランスをとる事が重要となる．よって，ここではデータ転送について検討する．但し離散ナブラ演算子については，逐次生成型ではなく，メモリ記憶型とする．

離散化ナブラ演算子は，要素に固有でメモリ上のまとまった領域に記憶でき，バースト転送が可能である．しかし，速度は節点において定義されており，要素を構成する節点のデータは必ずしもメモリ上に連続して記憶されてはおらず，要素から各節点へのインデックスを参照する必要がある．このため，節点 8 個分の速度データは 8 回にわけたバースト転送となる．

例として，ループ (2) を倍精度浮動小数点データ (64 ビット) を用いて処理した場合の、データ転送と演算のおおまかなスケジューリングを行なった (図 6)．メモリはパーソナルコンピュータで一般的に用いられている SDRAM(PC100, CL=2) を想定し，レジスタの数は十分にプロセッサ内に用意されているものとした．

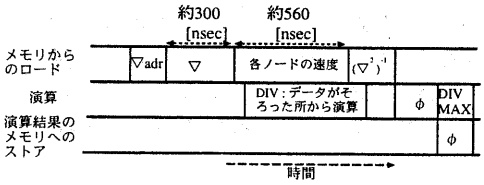

図 6: データ転送と演算のスケジューリング

この結果，節点 1 つあたりの速度ベクトルのデータの転送時間は約 70[ns] であると見積もられる．次の節点の速度ベクトルが読み込まれるまでに処理すべき演算は，速度ベクトルと離散化ナブラ演算子の乗算結果を変数 DIV に加算する処理だけである．演算数にすると，乗算 3 回と加算 3 回である．70[ns] で浮動小数点演算 6 命令実行する演算性能は約 80MFLOPS と見積もられ，現在のプロセスルールを用いて容易に実現できる性能である．

以上の検討により，ループ (1) の高速化はデータをプロセッサに転送する時間が支配的である事が分かり，次のような設計指針が得られた．

- 次の演算で使用するデータの転送を現在の演算と並行して行う．
- 次のデータが到着するまでに演算を終了させうる演算性能を用意．
- 可能な限り間断なくデータをプロセッサに供給する．

高速化を行なうためにはメモリ・プロセッサ間のデータ転送速度を高速化することが望ましい．しかし，我々は価格／性能比の良い専用計算機の開発を目標としているため，市販のメモリの利用を前提として考えている．

4 おわりに

本論文ではGSMAC FEM専用計算機を「科学技術計算専用ロジック組み込み型プラットフォーム」を利用して実現する際の，専用ロジック部分の検討を行なった．今回着目したコアループの ***div*** 計算部分には分岐がなく各方向成分の演算に並列性があり，パイプライン化やSIMD演算化による高速化が容易である．その反面，データにインデックス参照が多く使われ，データ転送の負荷が大きい．そのため，データ転送の効率化が重要であり，また演算とデータ転送の並行実行が必須である．

今後は，さらに具体的な検討と，今回取り上げなかった他のループの検討を行なっていきたい．

参考文献

[1] 棚橋隆彦,"GSMAC-FEM(流体力学の基礎とその応用)", アイピーシー (1991).

[2] 棚橋隆彦," 流れの有限要素法解析 I・II", 朝倉書店 (1997).

[3] 佐々木徹，荒木健悟，石橋政一，大谷泰昭，長嶋雲兵，溝口大介，村上和彰，" 科学技術計算専用ロジック組込み型プラットフォーム・アーキテクチャの開発 −プラットフォーム・アーキテクチャー",SWoPP 2000

[4] 川本昇一，棚橋隆彦," 分散メモリ型並列計算機を用いた GSMAC 有限要素法解析", 日本機械学会第 12 回計算力学講演会，講演論文集 99-5, (1999-11), 781-782.