

圧縮性流れの直交格子法による並列計算 Parallel Compressible Flow Computation via Cartesian Grid

城之内忠正, 四日市大, 三重県四日市市萱生町 1200, jyo@yokkaichi-u.ac.jp:
Tadamasa Jyonouchi, Yokkaichi Univ., Kayou-cho 1200 Yokkaichi-shi, 511-0912 JAPAN:

Component architecture in CFD application software is proposed. In this architecture, the flow field containing some body is divided into rectangular domain solver object. Each domain object has its own thread that enables parallel computing in symmetric multi processor machines. We developed a boundary value interpolation method in time, to synchronize each domain thread when domain needs other boundary values. This time interpolation algorithm prompts scalability of domain-decomposition computation.

1. はじめに

我々は格子生成の問題に対して、ソフトウェア部品の再利用という方向から、領域分割型部品として再利用するインターフェース⁽²⁾を提案してきた。再利用の仕組みは、予め特定のインターフェース(関数呼び出しのスタイル)を規程し、そのインターフェースを満たす関数をそれぞれの部品に実装しておく。そして、境界条件を求める等の必要に応じて、このインターフェースを呼び出すというものであった。具体的には、領域分割型のソルバーを部品として定義しておき、この領域分割内部は自律的に計算させ、境界値が必要になったときに隣接する領域のインターフェース(関数)を呼び出して、境界値を得るというものだった。このインターフェースを公開し、それぞれが実装すれば、プログラムの中味を知らなくても、相互に部品を利用できるというわけである。

この領域分割部品は、境界値の計算が容易であることから直交格子を使用していたが、有限体積法を使って、一般的な座標形状にも対応していたため、やや複雑であった。

さらに、最近直交格子法が格子生成の容易さから見直され、プログラムも単純になり、ソフトウェア教育の道具としても有望である。そこで、改めて直交格子法による領域分割型のソフト部品を開発することにした。この時、直交格子を使うために格子生成が容易となるため、既存の格子部品の再利用という視点ではなく、領域分割型部品のフレームワーク⁽³⁾として開発することにした。フレームワークというのは応用ソフトの枠組みを意味し、予め枠組みを作っておき、アプリケーション毎に変更しなければならない部分だけを、追加的に記述することによって、アプリケーションを効率的に作る方法である。

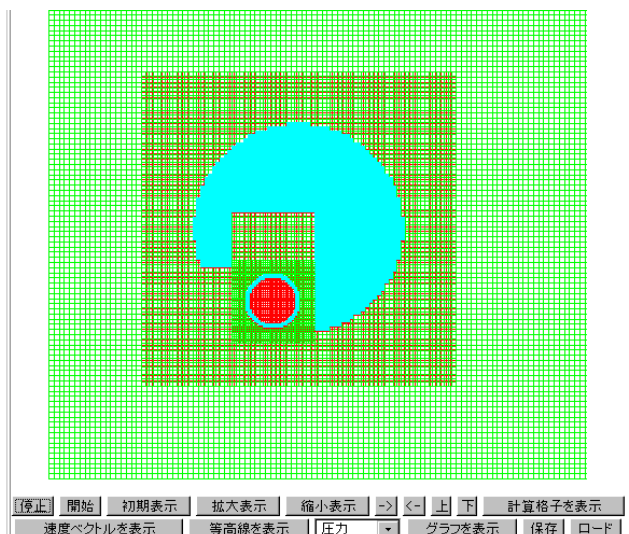
直交格子のアプローチは、計算格子にカーテシアン座標を使い、物体内の格子に印をつけることで形状を表現しようというものである。この方法の最大の長所は、格子が簡単に生成できるという点であり、形状が複雑になればなるほど直交格子は有効となる。さらに格子情報を記憶する必要がないため、プログラムが単純になるという点も見逃せない長所である。

しかし、物体を直交格子で表現するため、形状を忠実に表現することが難しいという欠点を持っている。例えば斜めの曲線を階段で表現するわけで、翼型のような流線形物体の表面圧力分布は、一般曲線座標のようになめらかに計算できない。

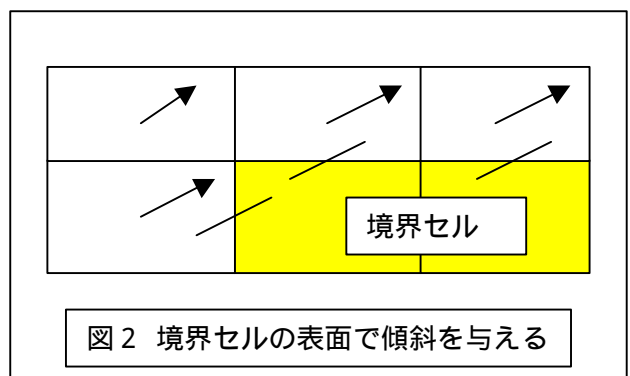
オイラー方程式においても、境界形状の階段状近似を使うと、格子をどんなに細分化しても解は収束しない(格子収束)。例えば、超音速の角をまわった後の斜面の流れを考えてみる。角を回る流れのように、代表スケールの無い問題においては、格子を細かく分割しても形状が相似形であるため意味が無い。それで、角を回る斜面について、粗い格子で考えて明らかに、この斜面は階段を過ぎる流れなのである。このため、境界条件の与え方で様々な手法が提案されているが、直交格子の単純さを維持したまま、簡単に計算できる手法が望ましい。

3. 階段状近似のまま境界傾斜を与える方法

境界近似を高めるために、境界セルを再分割したり、方程式の差分近似を修正したりする方法が提案されているが、オイラー方程式に対する簡便な方法として、階段近似のまま境界の傾きを満たすように速度を制御する方法を提案する。この方法は、形状を階段状に近似したまま、境界位置は変えないで、物体境界の傾斜データを記憶しておき、境界条件を与えるときに、そこでは階段なので傾斜がゼロか 90 度であるが、表面が傾斜を持つとして、境界速度をその傾き方向に流れるように設定する。



2. 直交格子法



例えて言うなら、薄翼理論の $z=0$ 面で翼の傾斜を境界条件として与える方法のような近似法である。図 1 に概念図を示すが、境界に傾斜を与えるということは、流れが境界に流入・流出するという意味であり、内部流などでは、質量保存等の問題が

生じるかもしれない。
プログラムに実装するには、境界位置と傾斜データを対応させておく必要があり、現在はセルの番号と傾斜データをハッシュ表に覚えさせている。

図3はくさびをすぎる超音速流れの計算結果で、計算格子と境界傾斜を適用しない場合と傾斜を適用した場合の等圧線が表示してある。傾斜を適用すると、斜面での圧力の凸凹が消滅し、一様に流れていることがわかる。なお計算手法の詳細は参考文献(3)と同じである。

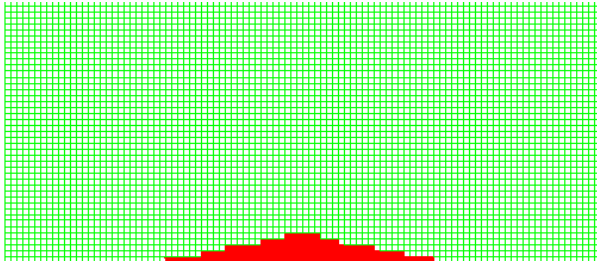


図3 楔をすぎる超音流

傾斜無し

4. 並列計算

領域分割型の計算法では、全領域に対して時間精度を合わせようとすると、領域毎の同期の問題などがあって、スケラブルに拡張することが難しい。領域分割した境界部分が陽解法の扱いとなるために、陰解法を適用することも困難な状況である。

領域部品はなるべく自律的に、あたかも独立のソルバーとして振る舞う方が計算のスケラビリティは高くなる。各領域部品はもともと独自のスレッドを持っており、自律的に計算を進めることができるが、境界値を交換するときに周囲と情報をやり取りする必要がある。そこで、お互いに境界値を時間内挿することで同期をとるアルゴリズムを考案した。

各領域部品は、自律的に時間刻み幅を計算し、流れ場の計算を進めるわけであるが、境界値のやり取りで同期を取ることによって、格子幅の大きい部分では計算量を著しく削減できるわけだ。時間内挿同期アルゴリズムは下記ようになる。(図4参照)

1. 各領域オブジェクトは時刻 n と $n+1$ の、2つの時刻の流れ場を記憶しておく

2. 各領域はそれぞれの状態に応じた時間刻み幅 t で内部を更新する
3. 親(コンテナ)領域の境界値を必要とする場合は、必要な時刻の境界値を親部品に要求する(関数呼び出し)
4. 要求された親領域は、2つの時刻の計算値から境界値を内挿出来るときは、その内挿値を返す。内挿出来る程、計算が進んでいないときは null を返す。
5. 境界値を要求した領域オブジェクトは、null 値が返ってきたらしばらく sleep する。しばらくしたら、もう一度要求を出し、境界値が内挿出来るまで繰り返す。
6. 子部品を持つ親部品は、子部品の境界値を必要とするため子部品に対して同様の境界値の要求をする
7. 時間内挿によって、各領域は同期を取りながら、時間精度が維持される

時間

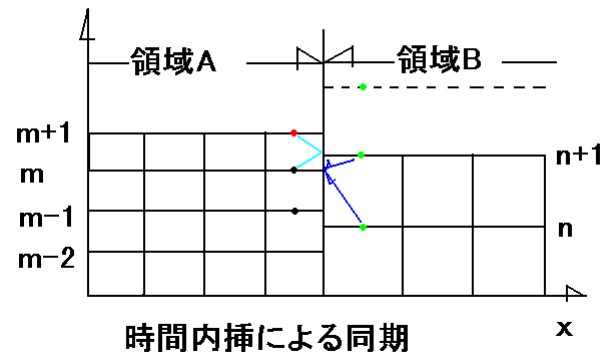


図4 時間内挿の同期アルゴリズム

5. まとめ

ながれの領域部品を組み合わせる数値実験を行う場合に、直交格子法は単純なために拡張性がある有効な手段と思われる。境界条件の精度向上に関しては、もう少し改善の余地がある。また、並列計算については原理的にはうまくいくが、2 CPU のシステムにおいて計算実績があるだけなので、規模の拡大に対する実証実験が必要である。

参考文献

- (1) 城之内忠正 "CFD における Java コンピューティング" ながれ 16-4(1997) 304-315

参考文献

- (2) 城之内忠正 "領域分割型 CFD 部品による編集と計算" 第13回数値流体力学シンポジウム講演予稿集 p.239
- (3) 城之内忠正・千葉賢 "直交格子法によるフレームワーク" 第11回計算流体シンポジウム講演予稿集 pp.369-370
- (4) 城之内忠正・千葉賢 "直交格子法によるオイラー計算" 第11回計算流体シンポジウム講演予稿集 pp.371-372