

CAE ポストプロセッサ IVS および適用事例

小山田 耕二

伊藤 貴之

日本アイビーエム 東京基礎研究所

本報告において、3次元数値シミュレーション結果のための可視化システム IVS 及びその適用事例について述べる。IVS では、大きく分けてデータブラウザ機能そしてデータ計測機能を提供する。データブラウザ機能は、3次元空間で定義された数値データを大局的に理解するための機能である。IVS では、ボリュームレンダリング表示、スライス・等値面のスイープ表示、そして特異点付近からの流線表示を提供する。データ計測機能は、この数値データを局所的に理解するための機能である。IVS において、ユーザは、3次元空間中の任意点に計測器を配置をして、物理量を計測する感覚で可視化処理を行なうことができる。実際の数値流体シミュレーション結果を用いて、これら機能の有用性を確かめた。

CAE post-processor IVS and its application

Koji Koyamada

Takayuki Ito

Tokyo Research Laboratory, IBM Japan Ltd.

In this paper, we describe a CAE post-processor IVS(Integrated Volume visualization System) for 3-D numerical simulation results and its application. IVS provides two major functions, a data browser function and a data measurement function. The former helps users to wholly understand a given dataset defined on a 3-D space. In IVS, it is composed of a volume rendering, a swept display of a volume slice and an isosurface, and a display of stream line starting from critical points. The latter helps users to explore the dataset as if they locate a measurement device at an arbitrary position in a real laboratory to measure a physical quantity. We applied IVS for understanding some results of CFD (Computational Fluid Dynamics) and confirmed its effectiveness.

1 はじめに

CAE は、「低コストで質の高いハードウェア製品をできるだけ早く市場に出すために計算機モデルを用いて設計開発の初期段階で性能評価を行なう手法」と定義された設計開発手法であり、多くの製造装置メーカーで実践されている。この CAE の中核をなす技術として、有限要素法や有限体積法といった数値シミュレーション手法がある。ここ数年における計算機のコストパフォーマンス向上により数値シミュレーションにおいて、計算機モデルは、ハードウェア製品の実機に近い形になりつつある。このため、これら数値シミュレーション手法は、設計開発に携わる技術者にとってより身近なものとなってくと同時に、より重要な技術的問題に適用されるようになってきた。数値シミュレーション結果は、当然のことながら大規模化、複雑化していく傾向にある。シミュレーションを担当する技術者が効率良くその結果を理解するため、そしてそのシミュレーションに関係する技術者にその結果から得られる情報をわかりやすく伝達するためにシミュレーション結果とのユーザインタフェースである CAE ポストプロセッサ¹の重要性は高まってきている。CAE ポストプロセッサそのものは、数値シミュレーション手法の出現とともに存在しているが、数値シミュレーション結果の大規模化や複雑化に追従するために新たな対応が求められている。1980 年代後半、Visualization in Scientific Computing の必要性が提唱されて [McCormick 87] 以来、3 次元空間で定義された数値データについての可視化技術（例えば、等値面表示やボリュームレンダリング表示）が多くの研究者によって開発され、商用 CAE ポストプロセッサもそれらの一部を取り込み始めてきている。これら手法の多くは、3 次元空間における単純な格子²上で定義されたスカラーデータを可視化する汎用的なものである。数値シミュレーション結果においては、データの定義される基本単位である格子が必ずしも単純な形状ではないこと、格子内部においてデータの定義される場所が必ずしも格子点ではないこと、対象となるデータが必ずしもスカラーデータではないことから汎用的手法ではうまく処理できない場合が多い。数値シミュレーションにおいて、現在進行形で新しい手法が開発されていくため、CAE ポストプロセッサは、最新可視化技術および最新数値シミュレーション技術の両方に歩調を合わせながら進化させていかなければならない。

本報告において、社内の数値流体シミュレーションユーザーとの協業により開発している CAE ポストプロセッサ IVS について、その技術的特徴および数値流体シミュレーション結果への適用事例について述べる。

2 入力データ

IVS は、入力データとして有限要素法および有限体積法のシミュレーション結果をサポートしている。こ

¹本報告において、数値シミュレーション結果の可視化という視点で開発された可視化システムを CAE ポストプロセッサと呼ぶ。
²ここでは、稜線が直線であるような格子を意味する。

れらのシミュレーション結果は、大きく分けて以下の 3 つから構成されている。

- **要素データ**
要素は、シミュレーション結果の定義されている 3 次元空間を構成する小立体のことを指す。要素を表現するために、小立体の頂点データのリストが用いられる。要素の形状として、4 面体、5 面体、6 面体をサポートする。
- **節点データ**
節点は、多くの場合、要素の頂点を指す。有限要素法においては、計算の精度を向上させるために要素稜線の間際に新たな節点を配置することがある。IVS では、要素稜線の間際に 0 個または 1 個の節点が配置された要素（それぞれ、1 次要素、2 次要素という）をサポートする。
- **数値データ**
数値データとして、スカラー、ベクタ、テンソルデータをサポートする。但し、テンソルデータは内部で等価なスカラーデータに変換している。データの定義される場所としては、節点、要素中心、要素毎の節点、そして要素面がある。これらデータは、一旦 IVS に読み込まれると、すべて節点上でのデータに変換される。この変換時には、シミュレーションで使われた境界条件や要素の属性が考慮される。

3 可視化機能

IVS では、シミュレーション結果を全体的に理解するためのデータブラウザ機能および局所的に理解するためのデータ計測機能というスケールの異なる 2 つの可視化機能を提供する。

3.1 スカラーデータブラウザ機能

3 次元空間で定義されたスカラーデータの分布を全体的に理解するための有効な手法の 1 つとして、スライス面上で等高線を描画しこれをスワイプ表示することが考えられる。スムーズなスワイプ表示を行なうには、スライス面上で等高線を描画するスピードをできるだけ高速化する必要がある。このスピードは、スライス面と交差する要素を特定する処理のパフォーマンスに大きく依存する。スライス面が更新される毎に全ての要素と交差判定を行なうと特定処理は要素の数に比例する。IVS の特定処理では、まず要素データから隣接要素リストを作成し、次に、スライス面と交差する 1 つの要素を出発点として隣接要素リストの幅優先探索を行なう（図 1 参照）。この時、1 つの要素に隣接する全ての要素を探索するのではなく、スライス面と交差する要素だけを探索することにより、探索される要素の数は、全体の要素数の $\frac{2}{3}$ 乗に比例したものとなる。この手法では、最初にスライス面と交差する 1 つの要素（シード要素）を見つけておく必要がある。IVS では、シード要素を見つけるに当たり、スライス面は境

界面³を持つ要素と必ず交差するという事実を使っている。

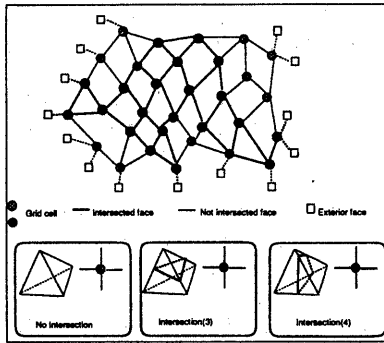


図 1: 隣接要素リストの幅優先探索

スライス面上で等高線を描画することに加えて、等値面 (isosurface) をスカラー値を変化させながらスイープ表示することもまた有効な手法である。等値面は、3次元空間で定義されたスカラーデータ $S(x,y,z)$ およびスカラー値 C に対し、 $S(x,y,z)=C$ を満たす点の集合として定義される曲面のことである。この場合も、スライス面のスイープ処理と同じく、シード要素を出発点とする隣接要素リストの幅優先探索を行なっている。シード要素を見つけるに当たっては、境界面に加えて、3次元空間内で定義される極点⁴も考慮している。IVS では、一般には複数存在する極点群を連結し、この結果作成される連結線上でシード要素を見つけている [Itoh 94]。

ボリュームレンダリングは、3次元空間で定義されたスカラーデータをデータの分布にしたがって発光する一種の密度雲 (発光粒子の集合体) として表示する手法である。ボリュームレンダリングでは、ディスプレイ上の輝度値 B を求めるために、輝度値方程式をディスプレイ上の画素点毎に積分する。IVS では、以下のようにして表される Sabella の輝度値方程式 [Sabe88] を採用している。

$$B = \int_{t_n}^{t_0} c(t) \times \rho(t) \times \exp\left(-\int_t^{t_0} \rho(\lambda) d\lambda\right) dt. \quad (1)$$

ここで、 t は、視線 (画素点と視線を結んだもの) 上にとられた変数で、 t_0 は、与えられた3次元空間と視線との交差点のうち最も近いもの、そして t_n は、最も遠いものを表す。 $\rho(t)$ は、単位長さ当たりの粒子数そして $c(t)$ は、単位粒子当たりの発光量を表す。式 1 は、視線上から発せられた発光エネルギーが指数項で表される減衰を受けて視点に到達する様子を示している。この式を解析的に計算することは一般的に不可能なので、基本的には区間 (t_0, t_n) を n 個の小区間に分けて数値積分を行なう。小区間内では、中点を代表点 (サンプリング点) とする。この基本的な方法では、視線毎に独立し

³要素を構成する面のうち、隣接する要素を持たないものを指す。

⁴スカラーデータが極大または極小となる点であり、等値面表示に先だって計算しておく。

て輝度値を計算するので、隣接する視線間のコヒーレンスを考慮していない。1つの要素の大きさが画素点の大きさに比べて大きい時、このコヒーレンスを考慮していないことの影響は無視できない。IVS では、全ての視線においてサンプリング点が等間隔で配置されていると仮定して、対応するサンプリング点群を通る球状スライス面群を重ね合わせることでよりボリュームレンダリングを実現している [Koyamada 92]。球状スライス面は、一種のスライス面なのでスライス面をスイープ表示する時に開発した手法をそのまま用いることができる。この球状スライス面上で輝度値への寄与を計算し、ディスプレイ上の画素点に反映させる。球状スライス面を視線の上から順に計算していくことにより、ボリュームレンダリングにおける密度雲生成の途中経過を表示することができる。この場合、ユーザは、途中経過を見ながら、レンダリング処理を中断して新たなパラメータに基づくボリュームレンダリングを再実行させることができる。

3.2 ベクタデータブラウザ機能

3次元空間で定義されたベクタデータの分布を全体的に理解する場合の単純な手法として、ベクタデータを代表させる矢印を3次元空間内で表示することが考えられる。この手法は、矢印の向きに曖昧性があったり、奥行き方向に矢印が重なったりして、現実には、利用されることが少ない。しばしば、流線 (stream line) 表示が用いられる。流線は、各点での接線がそこのベクタデータに平行であるような曲線であり、以下の方程式を満足する。

$$\frac{dx}{u} = \frac{dy}{v} = \frac{dz}{w}$$

ここで、 (u,v,w) は、点 (x,y,z) で定義されたベクタデータを表す。流線計算には、開始点が必要となる。やみくもに開始点を設定して流線計算を行なってもベクタデータの分布を全体的に表現することは困難である。そこで、適切な開始点を決めることが重要となる。適切な開始点を求める1つの有効な手段として、特異点 (critical point) を考慮することが考えられている。特異点は、ベクタデータがゼロベクタとなるような点である。特異点理論 (critical point theory) によると、特異点付近で与えられたベクタ場を線形化することにより、その点付近でのベクタ場の挙動を調べることができる [Abra84]。

$$\begin{aligned} u &= \Delta x \times \frac{\partial u}{\partial x} + \Delta y \times \frac{\partial u}{\partial y} + \Delta z \times \frac{\partial u}{\partial z} \\ v &= \Delta x \times \frac{\partial v}{\partial x} + \Delta y \times \frac{\partial v}{\partial y} + \Delta z \times \frac{\partial v}{\partial z} \\ w &= \Delta x \times \frac{\partial w}{\partial x} + \Delta y \times \frac{\partial w}{\partial y} + \Delta z \times \frac{\partial w}{\partial z} \end{aligned}$$

ここで、 $(\Delta x, \Delta y, \Delta z)$ は、特異点からの移動ベクタを表す。上式で、移動ベクタの各成分にかかる係数行列 J

$$J = \begin{bmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{bmatrix}$$

は、ベクタ場のヤコビアン行列と呼ばれる。数値シミュレーション結果からこの特異点を探索するには、各要素の内部でベクタデータを補間関数を用いて表現し、ゼロベクタとなるような点があるかどうかを調べる。要素内部においてベクタデータは、節点で定義されているベクタデータおよび補間関数を用いて表現される。補間関数は一般に要素内部の局所座標で表現されているので、解くべき方程式は、

$$\begin{aligned} u(p, q, r) &= 0 \\ v(p, q, r) &= 0 \\ w(p, q, r) &= 0 \end{aligned}$$

となる。ここで、 (p, q, r) は、要素内部の局所座標を表す。この方程式は、一般に連立非線形方程式となる。求められた局所座標 (p, q, r) が定義域に含まれる時、その要素内に特異点が存在することがわかる。6面体1次要素についての計算法 [Glob91] 及び4面体1次要素についての計算法 [Koyamada 94] が報告されている⁵。特異点が見つかったと、続いて、その点におけるヤコビアン行列 J の固有値 λ および固有ベクタを計算する。固有値は、固有値方程式 (3次方程式) の解として計算される。

$$\det(J - \lambda \times I) = 0$$

ここで、 I は、単位行列を表す。特異点は、計算された固有値 λ を用いて分類される [Helm91]。固有値が全て実数の場合には、その特異点はソース点/シンク点/鞍点と分類される。正の固有値に対応する固有ベクタは特異点から離れる方向のベクタデータに対応し、負の固有値に対応する固有ベクタは特異点に向かう方向のベクタデータに対応する。また、固有値が複素数である場合は、ベクタデータが局所的に渦を形成していることに対応する (図 2 参照)。

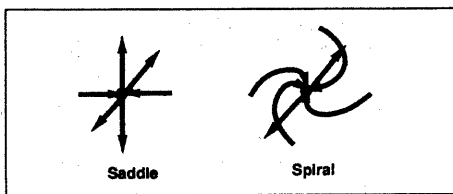


図 2: 特異点の分類

3.3 データ計測機能

数値シミュレーションユーザーの多くは、物理現象を解明するため、関連する物理量を計測したという経験を持っている。このような計測においては、どのように計測するのかと同様、どこで計測するのかも重要である。3次元数値シミュレーション結果を、数学モデルを使って再現された物理現象と仮定するならば、

⁵前者については、方程式が連立3次方程式となり、後者については、方程式が連立1次方程式となる。

「可視化」は、「計測」に対応するものと考えられることができる。したがって、彼らは、ごく自然に、3次元空間中の任意点に計測器を配置して物理量を計測する感覚で、可視化処理を行いたいと考えるであろう。残念ながら、従来の可視化システムは、エンドユーザが連続的に指示する3次元位置座標をパラメータとした可視化が行えるようなユーザインタフェースを提供していなかった。このような背景をもとに、IVS は、3次元空間におけるデータ計測に類似した作業環境を提供している。

データを「計測」するにあたって、まず最初に行なわなければならないことは指示された点 (プローブ点) を含む要素を探索することである。数値シミュレーション結果の定義される空間は、重なることなくそして隙間を作ることなく、要素によって埋めつくされており、また隣接する要素は、節点を共有している。われわれは、この特徴を利用して効率の良い要素探索法を開発している。IVS では、まず、1つの要素を「適当に」(メモリ内に最初に格納されているもの/前回の探索で見つかったもの) 決めて、その内部点 (探索開始点) からプローブ点に向かって半直線を引く。半直線は、探索開始点において $t=0$ 、そしてプローブ点において $t=1$ となるように、パラメータ表現するものとする。探索開始点から半直線に沿って順次、要素を構成する面 (要素面) との交点でのパラメータ t の値を計算していく。要素面は、一般的には、平面であるとは限らない。曲面と直線との交点計算は、時間がかかるので、本システムでは、要素面の三角形分割により交点計算を高速化している。要素を順次探索するうえで、交点における t の値が始めて1を越えた時、その要素は、与えられたプローブ点を含むものとする (図 3 上参照)。

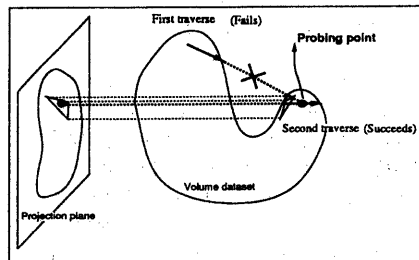
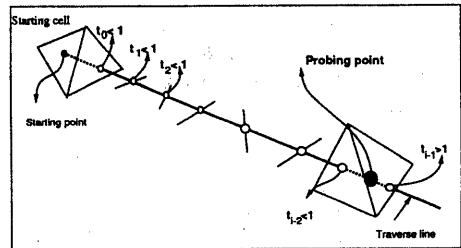


図 3: プローブ点を含む要素の探索

以上が基本的な考え方であるが、データの定義されている空間に凹部がある場合、 t の値が1を越える前に領域の境界に到達することがあり得る。この場合、探索開始点を変更して、再探索を行なう。IVSでは、再探索の回数を減らすために、以下の条件を満足する要素に含まれる探索開始点から再探索を行なっている(図3下参照)。

- 要素面が領域の境界を構成する。
- ある平面を考えた時、プローブ点のこの平面への投影点が要素の投影部分に含まれる。

プローブ点を含む要素が特定されると、次にその点でのデータ値を補間計算する。「ベクタデータブラウザ機能」のところで述べたように、要素内部においてデータ値は、節点で定義されているデータ値および補間関数を用いて表現される。補間関数は一般に要素内部の局所関数で表現されている。したがって、補間関数値を計算するには、プローブ点の座標を全体座標系から局所座標系に変換しなければならない。ここで、要素内において、プローブ点の座標(全体座標系)が、データ値と同様、節点座標(全体座標系)および補間関数を用いて表現されると仮定すると、解くべき方程式は、

$$\begin{aligned}x(p, q, r) &= x \\y(p, q, r) &= y \\z(p, q, r) &= z\end{aligned}$$

となる。ここで、 (x, y, z) は、プローブ点の全体座標を表す。この方程式は、4面体1次要素以外の要素では、一般に連立非線形方程式となる。

プローブ操作中は、プローブ点と関連する要素が常に特定されている。この要素をシード要素と考えることによりプローブ点を含むスライス面/等値面、プローブ点を出発する流線を高速に表示することができる。ユーザが連続的に指示するプローブ点の位置に合わせてこれらジオメトリを自動再生成することによりプローブ点の動きが生成ジオメトリに与え影響をリアルタイムに観察することができる(図4参照)。

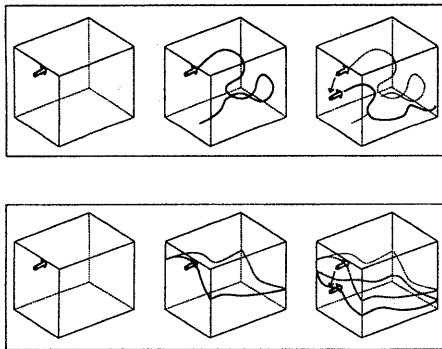


図4: プローブ点をパラメータとした可視化

4 数値流体シミュレーション結果への適用

数値流体シミュレーションは、計算機のパフォーマンス向上に当たって適用分野が拡大しつつある。従来航空機メーカーを中心に利用されていた数値流体シミュレーションは、コンピュータ製造メーカーにおいても、さまざまところで適用され始めている。ここでは、クリーンルーム内における数値流体シミュレーション結果への適用について述べる。

クリーンルームは、コンピュータ部品のような精密機器の製造ラインに広く使われている。クリーンルームにおいて、全体と同様の高清浄度が求められていないにもかかわらず、これまで一様に清浄度を高める試みがなされてきた。この結果、不必要でかつ膨大な運転コストが問題となってきている。クリーンルーム内には、それほど清浄度が必要とされない空間が存在する。コストのかかる高清浄度空間を局所化するためにレイアウトの最適化・整流板の配置といった検討が多く試みられている。このような検討案の効果を調べるための一手段として、数値流体シミュレーションが用いられている。この数値流体シミュレーションでは、速度場および速度場によって運ばれる物理量(汚染濃度)が各節点で計算される。

まず、汚染濃度の全体的な分布を知るためにスカラーブラウザ機能を利用した。スカラーブラウザ機能として、スライス面表示・等値面表示・ボリュームレンダリング表示を適用した。このうち、ボリュームレンダリングがもっとも有効であった。これは、ボリュームレンダリングによって生成される雲状の画像が現実の流れ場において利用されている可視化手法⁶により得られる画像と対比しやすいため理由の一つとして考えられる。

次に、数値流体シミュレーション結果から渦を抽出するためにベクタデータブラウザ機能を利用した。渦を探索するために固有値が複素数である特異点の近傍から流線を表示した。クリーンルームでは、アイリッド板と呼ばれる板を用いて室内全体を通路エリアとプロセスエリアとに分離している。アイリッド板付近で渦が存在すると、汚染物質のプロセスエリアへの誘引につながり好ましくない。このために渦が存在すると、アイリッド板の長さの変更等、再デザインが検討される。IVSの提供するベクタデータブラウザ機能により、渦の存在を自動的に調べることができた。

最後に、データ計測機能を使って、プローブ点を3次元空間内で対話的に動かし、その点における汚染濃度や速度を調べたり、プローブ点を通過する流線/スライス面コンタ図/等値面を表示した。IVSでは、これら可視化手法の起動がマウスボタンの押下で行なわれるので、思考の中断を伴うことなく可視化のスケール(局所から全体)を変えることができる。データ計測機能の利用することにより、実際の計測器の

⁶例えば、煙やメタルアルデヒド等の粒子をトレーサとして流れ場に供給し、気流によって運ばれるトレーサ粒子の密度分布を観察することにより気流性状を把握する方法などがある。

⁷数値流体シミュレーション結果を使っても、実際の計測作業そ

設置場所の決定に役立てることができた。

5 おわりに

本報告では、IVS のデータブラウザ/計測機能及びクリーンルーム内における数値流体シミュレーション結果への適用について述べた。ここでは、データブラウザ機能のうちボリュームレンダリングについて考察する。適用事例においても確認できたように、スカラデータの全体的を知るうえで、ボリュームレンダリングは有効である。しかしながら、現在、ボリュームレンダリングをサポートする可視化システムが流通するようになっているにもかかわらず、設計開発の現場において、このボリュームレンダリングがそう多く使われているようには思えない。設計開発の現場において、ボリュームレンダリングの必要性が低いのであろうか？ 2つの問題について考えてみたい。

1つは、ボリュームレンダリングに膨大な計算時間がかかることである。残念ながら、現在のボリュームレンダリングでは、対話的であるとはいえない。リアルタイムで回転したり、データ値と雲の発光量との関係を変動的に変更できたりすることが設計開発の現場においてボリュームレンダリングが日常的に利用されるようになるための必要条件となる。

もう一つは、現実の数値シミュレーション対象が、ボリュームレンダリングを必要としない程に局所化されていることである。数値シミュレーションにおいて、現在のところ、メッシュ作成にかかる時間が膨大なので、数値シミュレーション対象は、全体のうちごく一部を切り出したものにならざるを得ない。このような切り出し及びそこでの境界条件の設定のために、関連する物理量の全体的な把握が必要となり、どうしても実験による実験/測定が必要となってしまう。このため、設計開発の現場においてシミュレーションを行なう時、物理量の全体的な把握はすでに完了していることになる。これは、現在、数値シミュレーションを行なううえでのノウハウとして位置づけられている。メッシュ作成に時間がかからなくなり、数値シミュレーションが全体モデルの解析に利用されるようになった時に始めて、ボリュームレンダリングによる物理量の全体的な把握が必須になってくる。この時までには、リアルタイムボリュームレンダリングを実現できるようにしていなければならない。

参考文献

- [Abra84] Abraham, R. and Shaw, C., "Dynamics: The Geometry of Behavior," parts 1-4, Ariel Press, Santa Cruz, CA, 1984.
- [Glob91] Globus, A. and Lasinski, T., "A Tool for Visualizing the

Topology of Three Dimensional Vector Fields," IEEE Visualization '91, pp. 33-40, 1991.

- [Helm91] Helman, J. and Hesselink, L., "Visualizing Vector Field Topology in Fluid Flows," IEEE Computer Graphics and Applications, Vol. 11, No. 3, pp. 36-46.

- [Koyamada 92] Koyamada, K., Uno, S., Doi, A., and Miyazawa, T., "Fast Volume Rendering by Polygonal Approximation," Journal of Information Processing, Vol. 15, No. 4, pp. 535-544, 1992.

- [Koyamada 94] Koyamada, K. and Itoh, T., "Finding of Critical Points by Using Tetrahedral Cells," WCCM'94 Proceedings, pp. 535-544, 1994.

- [Itoh 94] Itoh, T. and Koyamada, K., "Fast Isosurface Generation by Using an Extrema Graph," IEEE Visualization'94 Proceedings, 1994 (To appear).

- [McCormick 87] McCormick, B. H., DeFanti, T. A. and Brown, M. D., "Visualization in Scientific Computing," Computer Graphics, Vol.21, No.6, 1987.

- [Sabe88] P Sabella, "A Rendering Algorithm for Visualizing 3D Scalar Fields," Computer Graphics, Vol. 22, No. 4, pp. 51-58, 1988.

のものは、その回数は減らせても、現実には省略することはできない。