

OpenGLを用いたビジュアルコンピューティングシステムの構築

日本電子計算 正会員 ○ 昼間 修
中央大学 正会員 檜山 和男

1. はじめに

数値解析の大規模化に伴い、可視化にも多大な計算時間と計算機容量を要するようになってきた。従来までの可視化は、時系列の計算結果ファイルをGWS上で、商用のソフトを用いて行う方法が主流であった。しかしこの方法は、計算環境からGWSへのデータ転送や、可視化ソフト特有のデータ変換が必要となり、特に3次元解析に対しては、結果が得られてから可視化が行われるまでに相当の時間を必要とした。また計算が様々なOS上で行われるようになった現在、その計算環境の変化に対応することは容易でない¹⁾。このことから、計算機のプラットフォームを選ばないビジュアルコンピューティングシステムを構築することは有意義である。

そこで本研究では、グラフィックス・ライブラリにOpenGL²⁾を用いて、計算機のプラットフォームを選ばない汎用性の高いビジュアルコンピューティングシステムの構築を行う。3次元キャビティ流れ解析、並列有限要素法による3次元スロッシング解析に本システムを適用し、その可視化例を示す。また可視化までの効率の向上という点について、本システムの有効性の検討を行う。

2. システム構成

本システムを構築するにあたり、グラフィックス・ライブラリにOpenGLを使用した。現在、解析はUNIXやWindows, Linuxなど様々なOS上で行われている。この解析環境の変化に合わせた可視化システムを構築するには、ハードウェアに依存しないグラフィックス・ライブラリであるOpenGLを用いることが最も適当であると考えられる。また、レンダリングを行うウィンドウの制御等の処理を容易にするために、OpenGL補助ライブラリであるGLUT, GLUIを使用した。これらのライブラリはWeb上で公開されており、無料で入手することが可能である。OpenGLが標準で装備されていないLinuxに対しては、OpenGLと互換性のあるライブラリであり、その機能のほとんどを有するMesa3Dライブラリを使用することによって対応した。

3. 可視化処理の流れ

本システムは図-1に示す各プロセスによって画像処理・可視化演算¹⁾・可視化表示が行われる。

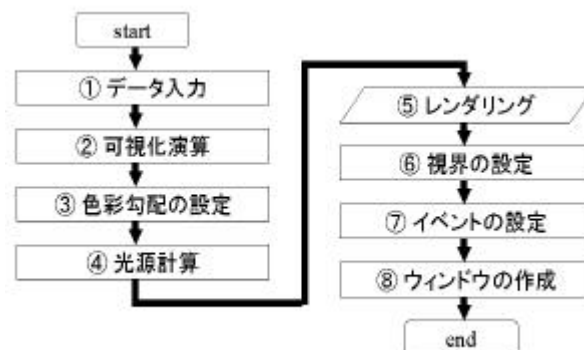


図-1 可視化処理の流れ

①はメッシュデータや計算結果、ファイル指定先等の可視化に必要なデータの読み込みを行うプロセスである。②は、①から得られたデータをもとに、等高線や等値面、流速ベクトル等を表示するための計算¹⁾を行うプロセスであり、可視化に必要な幾何学データの作成を行う部分である。③は、計算結果（圧力・ベクトル等）を色彩のデータに変換するプロセスである。ここで色彩勾配を作成し、各節点や各要素に対応した色彩のデータを割り当てる。④はライティング処理を行うプロセスである。光源の設定や、オブジェクトの質感や量感のプログラミングもここで行う。⑤は①～④で計算してきた各可視化演算のレンダリングを行うプロセスであり、グラフィックスの表示を行う。OpenGLの幾何学的描画プリミティブを使用して、表示方法（メッシュ表示、ソリッド表示等）の指示を出す。⑥は視界を定義するプロセスであり、カメラのレンズのような役割をする部分である。視点位置の決定や射影方法の決定を行う。⑦はマウス・キーボード操作、時間制御などの何らかのイベントを発生させるプロセスである。⑧はウィンドウのオープン/クローズといったOSに依存する処理を行う。

4. ビジュアルコンピューティング

本研究で用いたビジュアルコンピューティングの流れを図-2に示す。まずソルバーをスタートさせ、適当なタイムステップごとに計算結果をファイルに書き出させる。次に可視化プログラムの方も立ち上げ、前節で述べた順序で、計算結果を読み込み、可視化を行う。ソルバーで新しい計算結果が出力されたらその計算結果を再度読み込み、グラフィックスの再描画を行う。

可視化に要する計算機容量を最低限にするため、ある一定時間間隔で可視化プログラムが起動するように設定した。これにより、可視化プログラムを作動させること

Key Word: OpenGL, リアルタイム可視化, 有限要素法

〒112-8551 東京都文京区春日 1-13-27

TEL 03-3817-1815, FAX 03-3817-1803

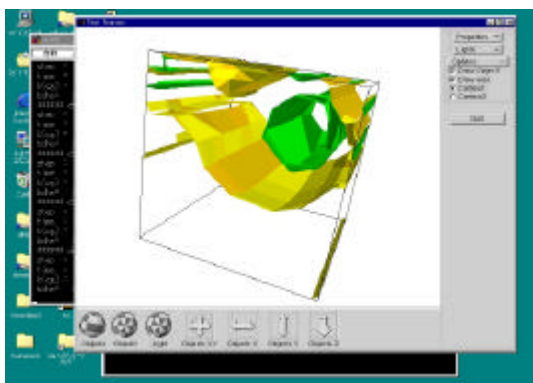


図-3 ビジュアルコンピューティングの様子

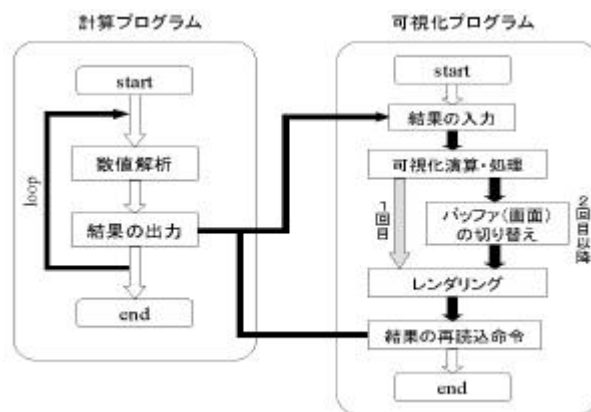


図-2 ビジュアルコンピューティングの流れ

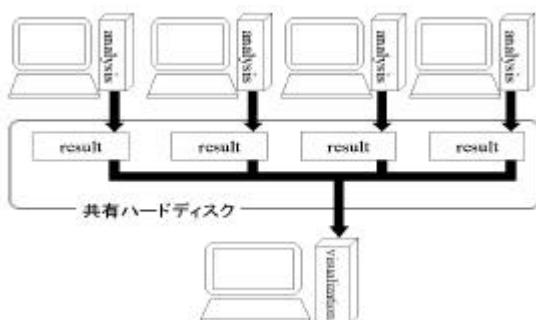


図-4 データ移動概念図

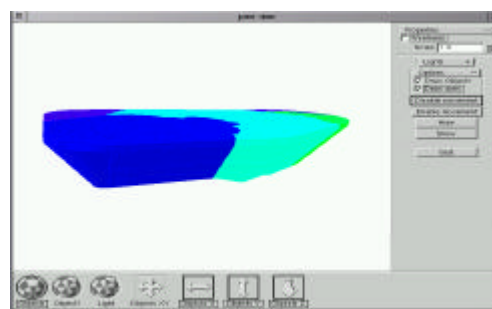


図-5 並列計算への適用例

による計算速度の低下を防ぐことができる。本システムはソルバーと可視化プログラムが完全に独立している。そのためソルバーのプログラムを変更することなく、誰でも簡単にビジュアルコンピューティングを行うことが可能である。

4.1. 適用例1(3次元キャビティー流れ解析)

本システムを、3次元キャビティー流れ解析に適用した。図-3は実際にビジュアルコンピューティングを行っている様子を示したものである。システムの下にあるパネルが視点を変換するためのコントローラーであり、右のパネルが表示方法の切り替えや可視化のパラメーターを変更する部分である。これらのGUIは、GLUIライブラリをC++言語でプログラミングし作成した。

4.2. 適用例2(並列スロッシング解析)

本システムをPCクラスターを用いた並列計算機(OS:Linux)に移植し、並列有限要素法によるスロッシング解析に適用した。並列計算には4台のプロセッサを用いた。各プロセッサはそれぞれの領域の計算結果を、可視化を行うコンピュータと共有されたハードディスクに書き込む。可視化コンピュータ側に計算結果のある場所を示すpathを指定し、結果を読み取り、可視化を行う。可視化プログラムは、各領域ごとにそれぞれの演算を行い、画像表示の際に可視化のデータを合成表示する。データの移動を示す概念図を図-4に、並列計算の可視化例を図-5に示す。

5. おわりに

本論文では、OpenGLを用いて計算機のプラットフォームを選ばないビジュアルコンピューティングシステムの構築を行い、3次元キャビティー流れ解析に適用した。また並列計算機に本システムを移植し、並列有限要素法による3次元スロッシング解析に適用した。その結果以下の結論を得た。

- 本システムは、UNIXやLinux、Windowsで稼動し、あらゆる計算環境（並列計算含む）に適用可能であることが示せた。
- 並列計算に対するビジュアルコンピューティングは、大規模計算の妥当性の判断を瞬時に行うことを可能にし、計算から可視化までの効率を大幅に向上させることが出来る。
- 本計算からのデータをそのまま可視化することができ、計算と可視化が完全に独立しているため、解析者への負担が少ない。

今後の課題は、さらに本システムを拡張し、ポストからプリやソルバーに値を返し、物体形状の修正・計算条件の変更等を行うことの出来るシステムを構築することである。

参考文献

- 1) 数値流体力学編集委員会：格子形成法とコンピュータグラフィックス，東京大学出版会，1995
- 2) The Official Guide to Learning OpenGL Version 1.1, Addison-Wesley Publishers Japan, 1997