

環境流体計算と3次元可視化手法について

首都大学東京 都市基盤環境コース 新谷哲也

1. はじめに

閉鎖性湾や湖内の流動・水質を解析するために様々な3次元数値モデルが開発され、有用なツールとして幅広く研究・実務に利用されている。これら3次元モデルを用いることによって、観測等では取得が困難な時空間的に解像度の高い情報を容易に得ることができる。しかしながら、一方で大量に吐き出されるデータの処理、特に可視化が問題となることが多い。通常、3次元データは統計的に処理された形（鉛直方向に平均化等）や2次元断面（スライス）で表現されることが多いが、局所的に生じる水質変化や汚濁物質の追跡等、3次元的な可視化表現が必要とされるケースも少なくない。著者が近年文献やインターネットで（主に市販ソフトウェアを使わない）3次元の流体可視化手法について調べた際に、簡潔に情報がまとまっている文献が意外と少なかった。そこで本稿では、その際に得られた動向とともに、現在、有力と思われる可視化ライブラリーVTK(Visualization Tool Kit)^①を環境流体モデルへ組み込んだ一例を紹介する。

2. 3次元CGの基本

3次元CG(Computer Graphics)とは、コンピューター内の3次元仮想空間に可視化する物体、光源、及び視点を配置し、その視点から見える映像を2次元の画面に投影する技術である^②。結果として作成された画像は2次元画像であるが、3次元的にデータを保持しているため、任意に視点を変更させても（例えば物体の裏側を見るなど）再描画することができる。3次元CG作成プロセスは一般的に以下のような手順となる。まず、モデリングと呼ばれるプロセスで物体の形状を決定し、色や質感を設定する。次に視点・光源の設定で3次元空間での物体の見え方を設定する。そして、最終的にレンダリングと呼ばれる処理で2次元投影面に描画する。一般的に3次元CGといえば、フォトリアリスティック（写実性を追求する）レンダリングであるが、医療関連や流体の可視化等（Scientific Visualization）では、物理的解釈が優先されるために写実性はあまり追求されない。また、静止画のレンダリングでは描画速度はあまり追求されないが、計算と描画を同時に行うリアルタイムレンダリングと呼ばれる手法では、描画速度向上（フレームレート維持）のために様々な工夫がなされている。

3. 3次元可視化手法の現状

数値計算で得られる3次元データの可視化手法は大きく分けて二通りある。一つは、既存の可視化ソフトウェアを用いる方法であり、もう一方は、（公開されている）可視化ライブラリーを利用して独自のプログラムを作成する方法である。前者としては、Paraview(<http://www.paraview.org/>)、OpenDX(<http://www.opendx.org/>)、Vis5D(<http://vis5d.sourceforge.net/>)のようなフリーウェアも存在するが、Matlab、AVS/Express、IRIS/Explorer、Tecplotなど非常に高価なものが少なくない。また、Povray (<http://www.povray.org/>)などフリーのレイトレーシングソフトウェアも3次元データの可視化（ボリュームレンダリング、等値面等）に用いることもできる。本稿ではこのようなパッケージソフトウェアを用いるのではなく、後者の可視化ライブラリーを用いた可視化プログラムの構築、もしくはモデルに可視化機能を加える手法に着目する。まず、3次元可視化ライブラリーとしてOpenGLとDirectXがその代表格として挙げられる（他にもVRMLなどがある）。OpenGLは、SGIによって科学計算における可視化を目的に開発されたライブラリーであり、DirectXはMicrosoftによって開発され、映像にとどまらずマルチメディアソフトウェアの開発を目的としている（Windows用のみ）。これらのライブラリー(API: Application Programming Interface)を直接プログラムから呼び出して可視化することは手間がかかるため、現在では、それらの拡張APIとして、VTK(<http://www.vtk.org/>)やJava3D(<http://java3d.java.net/>)等が開発されている。これらのAPIでは下位にどのような描画ライブラリーが使われているか気にする必要はない。Java3Dは、クロスプラットフォームであることに加え、JavaアプレットとしてWebブラウザ上で実行できるなどの利点がある。一方、VTKもC++で記述されているクロスプラットフォームライブラリーであるが、C++やJavaに加え、Tel、Python、(Ruby)などのスクリプト言語に対応している等の特徴を有する。また、VTKは科学計算可視化用の機能（等値面、

等高線，ベクトル図，スライス図，ボリュームレンダリング等) が非常に充実しているため，流体を3次元的に可視化するには，現時点では一番適していると思われる（この分野は日進月歩であるため，常に最新の情報をチェックする必要がある）．以下では，VTKを環境流体モデルに組み込む例について説明する．

3. モデルへの組み込み例

多くの環境流体を扱うモデル（例えば POM, ROMS, ELCOM, FVCOM）では，可視化処理をモデル本体とは別に扱っているが，数値計算の結果をリアルタイムに表示・操作することは，現象の理解を助けるとともに，モデル初心者への敷居を下げることに，また，モデル開発中の修正・確認作業において有用となると思われる．本稿では，コアとなる流体モデルとして，オブジェクト指向環境流体モデル Fantom3D[®]を用いた例を示す．Fantom3DはROMSなどと同様にCUI（Character User Interface）ベースのモデルである．このモデルの計算ループ内にVTKによる描画処理を加え，流れや物質の輸送状況のリアルタイム表示を試みた．

ここでは，紙面の関係上ライブラリーのインストールや言語上での使用方法は省略して，VTKを用いた流体可視化の要点を説明する（図-1）．描画プロセスは大きくデータ処理セグメントと描画処理セグメントに分けることができる．まずデータ処理において，計算で使っている3次元格子情報を元にグリッドオブジェクト（vtkStructuredGrid等）を作成する．続いて，グリッドの各格子点で時々刻々得られるスカラー（温度，塩分など）やベクトル（速度成分）値を配列形式でグリッドオブジェクトへ代入してSource（基礎データ）とする．次に，このSourceをFilterに渡してSourceのデータ構造を加工する．このFilter操作には地形のマスク(threshold)や等値面(contour)への変換等が含まれる．このSourceが持つ情報を使用して描く物体ごとにMapperを作成し，描画前の物体の物理形状等を決定する（コンターの範囲，カラーマップ等はMapperで決定する）．次に描画処理セグメントへと移行する．MapperのデータはActorへと渡され，Actorで物体の透明度や色の変更，表示・非表示の設定などを行う．最終的にRendererへActor群，カメラ（視点），光源を登録して2次元平面に投影し，RenderWindowに渡して画面表示する．さらにInteractorを追加することで，マウスを用いて画面上のオブジェクトの平行移動，回転，拡大縮小等の操作が可能となる．その他，文字列や凡例もRendererに登録することで画面上に自由に配置できる．

上記の3D描写を毎計算ステップ行うと計算時間を大幅に増加させてしまう可能性があるため，一定の計算ステップ間隔で更新し，必要なときにはすぐに最新の状態にアップデート可能にするなどの工夫が必要となる．最後に，VTKを組み込んだモデルのスナップショットを図-2に示す．この図では，描画内容を動的に変更するためのインターフェイス(GUI)としてQt (<http://qt.nokia.com>)を利用した．

参考文献

(1)Schroeder et al.: The Visualization Toolkit: An Object-Oriented Approach to 3-D Graphics, 4th edition, Kitware, 2006. (2)荒屋真二: 明解3次元コンピュータグラフィックス, 共立出版, 2003. (3)新谷哲也・中山恵介: 環境流体解析を目的としたオブジェクト指向流体モデルの開発と検証, 水工学論文集, 2009.

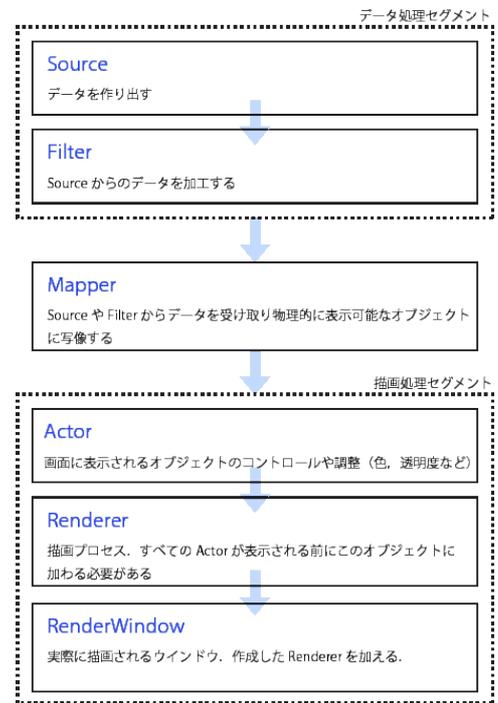


図-1 VTK 描画の流れ

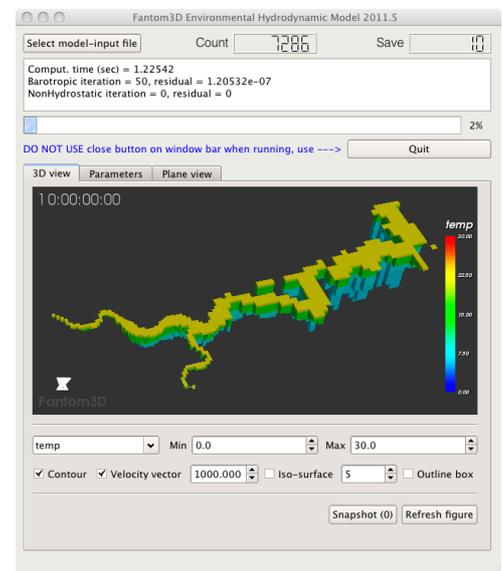


図-2 VTK 導入後のスナップショット