事例１

フォームクライオターゲットの爆縮実験データの可視化

大阪大学大型計算機センター研究開発部
出口　弘
deguchi@center.osaka-u.ac.jp

大阪大学レーザー核融合研究センター計算機室
福田　優子
yuko@ile.osaka-u.ac.jp

１．レーザー核融合と可視化

レーザー核融合の概念を図１に簡単に示します。レーザー光を燃料ペレットに照射するとペレット表面が高温に加熱されプラズマ化し噴出します。この噴出の反作用（ロケット作用）で、球殻は内向きに加速され、中央部が圧縮され、高密度（固体の約1000倍）高温（約１億度）の状態となります。このようにして核融合の自己点火条件が満たされると爆発的に燃焼し、照射エネルギーの数百倍のエネルギーが放出されます。一般的にはペレットターゲットの直径は約１mm程度であり、この加熱、圧縮、点火、燃焼の諸過程は１ナノ秒のオーダーで起こります。このように爆縮は非常に短い時間・空間スケールで変化するため、実験計測には技術的困難が伴います。レーザー核融合の研究においては物理過程の解明や、データ解析に計算機シミュレーションが重要な役割を担ってきましたが、実験データや、シミュレーションデータの可視化もデータを解析する上で必要不可欠となっています。

図２はフォームクライオターゲットの爆縮実験データを IRIR Explorer を用いで可視化したものです。フォームクライオターゲットは超低密度の中空プラスチックフォーム球に液体燃料を浸透させたもので、高密度圧縮に適しており、将来の高利得ターゲットの有力な候補となっているものです。５台のＸ線ピンホールカメラを用いてターゲット中心部から発生するＸ線を撮影し、ＣＴの手法を用いて処理した３次元表示です。時間積分された像で、実際は１mmの立方体の空間を２７×２７×２７のデータで表示しています。これは、1987年の実験データであり、３次元的に可視化することによって、歪んで圧縮している様子を立体的に知ることができ、初期ターゲットの厚さの不均一性との関連を解析することができました。

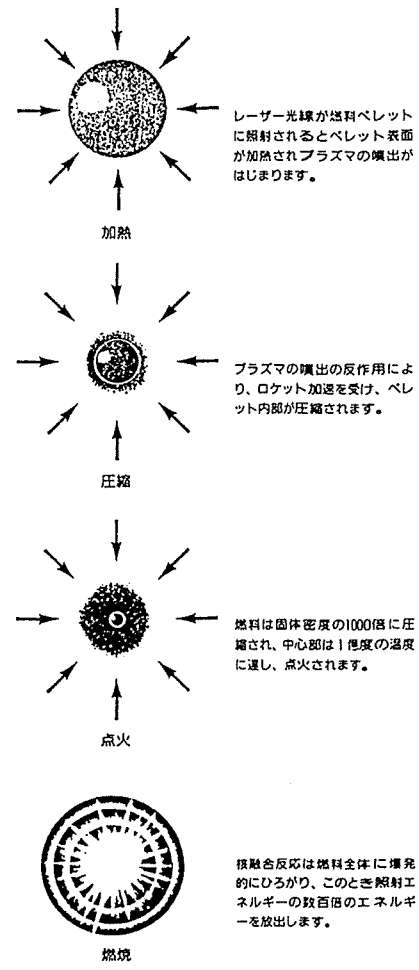

図１　レーザー核融合

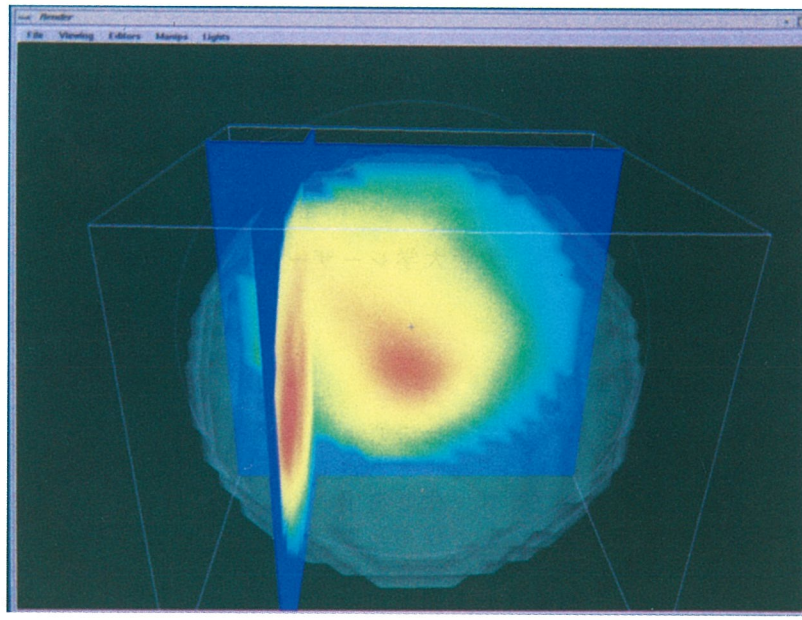

図２　フォームクライオターゲットの爆縮

２．入力モジュールの説明

可視化に用いた実験データは、以下のようなフォーマットでアスキーコードの文字列として表現されています。

```
27                                                    ｘ軸方向のデータの個数
27                                                    ｙ軸方向のデータの個数          ①
27                                                    ｚ軸方向のデータの個数
0.100E+01 0.200E+01 0.300E+01 0.400E+01 0.500E+01 0.600E+01 0.700E+01 0.800E+01
0.900E+01 0.100E+02 0.110E+02 0.120E+02 0.130E+02 0.140E+02 0.150E+02 0.160E+02
0.170E+02 0.180E+02 0.190E+02 0.200E+02 0.210E+02 0.220E+02 0.230E+02 0.240E+02
0.250E+02 0.260E+02 0.270E+02                         ２７×２７×２７個のｘ座標      ②
        ・・・ｘ座標省略・・・
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01
0.100E+01 0.100E+01 0.100E+01                         ２７×２７×２７個のｙ座標      ③
        ・・・ｙ座標省略・・・
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01
0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01 0.100E+01
0.100E+01 0.100E+01 0.100E+01                         ２７×２７×２７個のｚ座標      ④
        ・・・ｚ座標省略・・・
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
0.000E+00 0.000E+00 0.000E+00                         ２７×２７×２７個のデータ      ⑤
        ・・・データ省略・・・
```

実験データは、立方体空間を２７×２７×２７のセルに等分割して表現しているため、Explorer の３次元定型格子データとして扱うことができます。定型格子データには座標の値は不要なので、それは読み飛ばすことになります。図３は実験データを読み込み、Explorer データタイプを出力する read_laser モジュールを DataScribe で作成したところです。

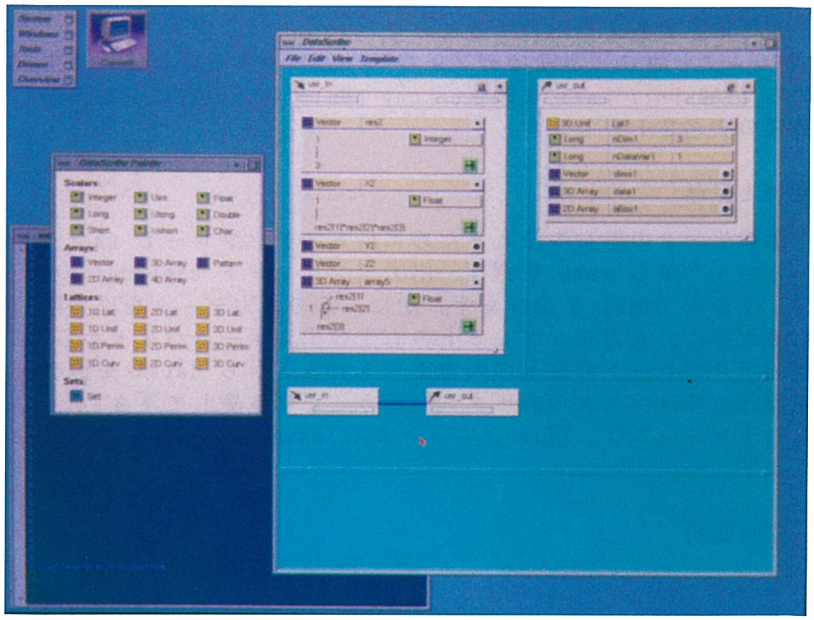

図３　入力モジュール read_laser

2.1 入力テンプレート

まず Template メニューで、Direction を Input、Type を Ascii として入力テンプレートを作ります。名前をデフォルトの Template1 から user_in に変更しています。

①のデータの個数を読み込むためにパレットから入力テンプレートに Vector をドラッグします（名前を res2 としています）。その Vector res2 の右丸をクリックして開き、インデックスをデフォルトの１－Ｎから１－３にします。要素の型はデフォルトで Integer となっています。これで Vector res2 は Integer 型の３次元ベクトルとなります。

②のｘ座標値を読み込むためにパレットから入力テンプレートに Vector をドラッグします（名前を X2 としています）。その Vector X2 の右丸をクリックして開き、インデックスを１－Ｎから１－res2[1]*res2[2]*res2[3]とし、パレットから Float をドラッグして要素の型を Float とします。これで、Vector X2 は Float 型の（res2[1]×res2[2]×res2[3]）次元ベクトルとなります。同様に③、④のｙ、ｚ座標値を読み込む Vector をそれぞれ作ります（Y2とZ2）。

⑤のデータを読み込むためにパレットから入力テンプレートに 3D Array をドラッグします（名前を array5 としています）。その 3D Array array5 の右丸をクリックして開き、インデックスをデフォルトの１－Ｎ１、Ｎ２、Ｎ３からそれぞれres2[1]、res2[2]、res2[3]とし、パレットから Float をドラッグして要素

の型を Float とします。これで、3D Array array5 は Float 型の res2[1]×res2[2]×res2[3]の３次元配列となります。

2.2 出力テンプレート

まず Template メニューで、Direction を Output、Type を Explorer として出力テンプレートを作ります。名前をデフォルトの Template2 から user_out に変更しています。

３次元定型格子データを出力するためにパレットから出力テンプレートに 3D Unif をドラッグします（名前はデフォルト Lat1 のままにしています）。右丸をクリックして 3D Unif Lat1 を開き、Long nDataVar1 の右に１を入力して格子データの要素の次元を１とします。

2.3 接続と保存

まず、①のデータの個数を伝えるために、入力テンプレート user_in の右上の出力ポートパッドから res2 を選び、出力テンプレート user_out の左上の入力ポートパッドから dims1 を選びます。次に、⑤の入力データを出力格子データの要素とするために、入力テンプレートの右上の出力ポートパッドから array5 を選び、出力テンプレートの左上の入力ポートパッドから data1 を選びます。これらの接続操作の時はマウスの右ボタンを使うことに注意して下さい。

完成したモジュールは、File メニューから Save As を選び、read_laser として保存します。

3. マップの説明

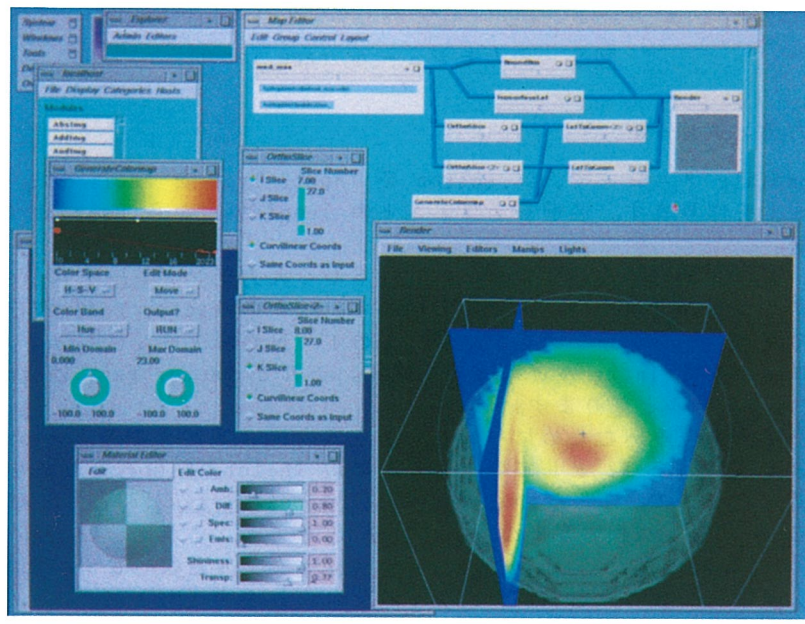

図４　可視化マップ laser.map

図４は図２の可視化画像を得るためのマップです。まず、DataScribe で作成したファイル入力モジュールの read_laser を呼び出します。次に、画像表示を得るための Render、等値面を表示するための IsosurfaceLat、データの境界を表示するための BoundBox、断面を得るための OrthoSlice と LatToGeom、断面に色付けするための GenerateColormap の各モジュールをテンプレートからマップエディタにドラッグします。あとは図４のようにそれぞれのモジュールの入出力ポートを接続します。GenerateColormap モジュールの接続は、その出力ポートパッドから Colormap--Lattice を選び、LatToGeom モジュールの入力ポートパッドの Colormap--Lattice と接続します。これらの接続操作の時はマウスの右ボタンを使うことに注意して下さい。

ファイル入力モジュール read_laser で実験データのファイル名を入力します。断面の軸と箇所を OrthoSlice モジュールのボタンとスライダーで調節、色付けの対応を GenerateColormap モジュールで調節します。また、等値面は Render モジュールの MaterialEditor を使って半透明表示にしています。

参考文献

1．Y. W. Chen et al. : three-dimensional imaging of laser imploded targets, J. Appl. phys. Vol. 68, p.1483 (1990).

2．出口　弘 : 汎用可視化ツール Explorer の使い方，大阪大学大型計算機センターニュース，Vol. 22, No. 2, 1992-8.