

予測制御による爆発の キーフレームアニメーション

佐藤 周平[†] 土橋 宜典[†] 山本 強[†] 安生 健一^{††}

近年、流体シミュレーションに基づいた自然現象のシミュレーション手法が多く提案されている。その一つとして、爆発のシミュレーション手法が提案されているが、その形状は多くのパラメータに依存しており、ユーザの意図した形状を得ることは困難である。我々は、これまで、ユーザの指定した単一の目標形状に一致するよう爆発のシミュレーションを制御する手法の開発を行っている。本稿では、この手法を拡張し、複数の目標形状をキーフレームとして指定することで複数回の爆発の制御を可能とする手法の開発を目指す。提案手法により、異なる目標形状に制御された複数の爆発のシミュレーションを連続して行うアニメーションの生成が可能である。

Keyframe Animation of Explosion Simulation with Predicted Control

Syuhei SATO[†] Yoshinori DOBASHI[†]
Tsuyoshi YAMAMOTO[†] Kenichi ANJO^{††}

Recently, many simulation methods for natural phenomena based on fluids simulation have been proposed. A method for explosion is also proposed, but the shapes of the simulated explosions depend on many simulation parameters. Therefore, it is difficult to create a shape of explosion that users specified. In this paper, we propose a method for controlling explosion simulations to form shapes that users specified. The user specifies the multiple shapes of the explosion at several key frames. Using the proposed method, the animation of explosions controlled into specified shapes can be generated.

1. はじめに

コンピュータグラフィックスの分野において自然現象のシミュレーションは重要な研究課題となっている。その自然現象の中でも、煙、雲、水などの流体現象のシミュレーションに関する研究は特に注目されている。また、映像生成の場においてもこれら流体現象の表現は重要な要素となっており、それらのアニメーションは幅広い分野で利用されている。自然現象のアニメーションを生成する方法は数多く存在するが、その中でも流体力学の物理シミュレーションを基にした方法は、自然に近いリアルな動きを表現できる²⁾。そのため、流体シミュレーションを用いて効率良く、効果的に自然現象を表現する研究はこれまでも数多く行われてきた。しかし、従来の研究では、ユーザが意図したとおりに流体を制御することは、考慮されていなかった。そこで、自然現象のリアリティを保ちつつ、様々な形状を表現するため、流体力学に基づいた自然現象の動きをコントロールする方法がいくつか提案されている³⁾⁴⁾⁷⁾。これらの手法では、煙や水をユーザが指定した形状となるよう制御を行っている。本論文では、爆発のコントロールに注目する。

流体力学に基づいた爆発のシミュレーション手法はいくつか提案されている⁵⁾⁶⁾。これらの手法を用いれば、映画やゲームなどの分野において爆発や爆発に類似した現象のリアルなアニメーションを生成することができる。しかし、これらの方法では、シミュレーションの制御が多くのパラメータに依存しているため、容易にユーザの意図した形状を得ることは困難であると考えられている。そこで、我々は流体力学に基づいた爆発の動きをユーザの指定した形状となるようにコントロールする手法を提案した¹⁾。しかし、この手法では、単一の目標形状しか指定できないため、複数回爆発を起こし、その都度、異なる目標形状に制御するといったことは不可能であった。

そこで、本稿では文献¹⁾の手法を拡張し、複数の目標形状をキーフレームとして指定することで、目標形状の異なる爆発のシミュレーションを連続して行う手法の開発を目指す。本稿では、実験として2次元における爆発においてキーフレームによる制御を行う。提案手法により、異なる目標形状に制御された複数の爆発のシミュレーションを連続して行うアニメーションの生成が可能である。また、文献¹⁾では、制御によって爆発の詳細な乱流成分が失われるという問題や複雑な形状に対しては制御できないという課題があった。本稿では、それらの点についての改善も行ったので合わせて報告する。

以降、本論文の構成は、2節で従来研究について簡単な説明を行い、3節で文献¹⁾の

[†] 北海道大学

Hokkaido University

^{††} オー・エル・エム デジタル

OLM Digital

手法とその変更点およびキーフレーム処理の詳細について述べる．4 節では，提案手法による適用例を示し，その有効性について検討し，最後に 5 節でまとめる．

2. 従来研究

本研究は爆発のシミュレーションおよび流体シミュレーションのコントロールに関連している．文献²⁾の手法を応用した爆発のシミュレーションを行うものに文献⁵⁾が存在する．また，流体シミュレーションのコントロールに関する研究として文献³⁾⁴⁾などが存在する．

2.1 爆発のシミュレーション

文献⁵⁾は，N-S 方程式の安定解法による手法²⁾を応用した爆発のシミュレーション方法を提案している．この手法では，gas model と particulate model を構築し，3 次元における N-S 方程式による流体解析と燃焼物やすすを表す粒子との間に発生する相互作用を考慮することで，爆発のシミュレーションを行う．しかし，爆発の形状が多くのパラメータに依存しているため，ユーザの意図した形状となるよう，それら全てのパラメータを調整することは困難である．

2.2 流体の形状コントロール

目的の形状が得られるよう流体シミュレーションをコントロールする手法として文献³⁾がある．これは，シミュレーション空間に粗い格子を配置し，その格子点から外力を発生させることで，流体を目的の形状を表す密度分布などに近づける手法であるが，シミュレーションに時間がかかるという問題がある．そのため，目標形状に近くような外力を発生するポテンシャル場を用いた，より高速な手法⁴⁾が提案されている．しかし，これらの手法では，煙が対象となっており，爆発のコントロールについては考慮されていない．爆発は煙に比べて，はるかに高速な動きがあるため，これらの手法による制御では精度よく爆発の動きをコントロールすることが難しい．この問題を解決するために，本研究では，ユーザが指定した形状に一致した爆発が生成されるようコントロールする手法の開発を行った¹⁾．しかし，この手法では目標形状を一つしか設定できず，複数の目標形状を指定し複数回の爆発を制御するといったことは不可能であった．

本稿では，上述の問題を解決し，キーフレームとして目標形状を複数指定し，連続して起こる複数回の爆発において，各々の爆発が指定された形状に制御されているアニメーションの生成を可能とする．

3. 提案手法

3.1 爆発のシミュレーション方法

基本となる爆発のシミュレーション方法について説明する．本研究では，まず 2 次

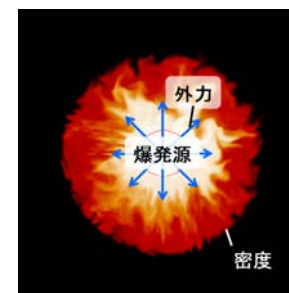


図 1：爆発のシミュレーション

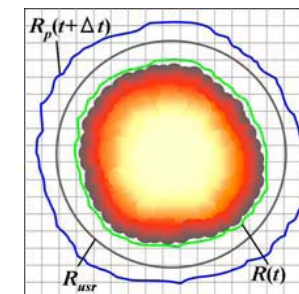


図 2：爆発形状の定義

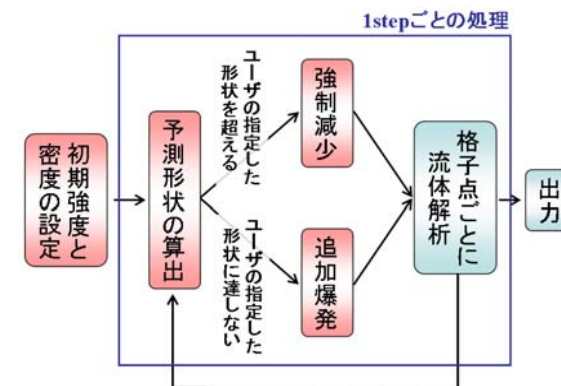


図 3：処理の流れ

元における爆発を対象とし，以下に述べる簡易モデルによりシミュレーションを行う．まず，2 次元のシミュレーション空間を格子分割し，その中心位置に爆発を発生させる点（爆発源）を配置する（図 1 参照）．そして，分割した各格子点に速度および密度を割り付け，格子点ごとにそれらの値の計算を行う．密度は爆発により飛散する燃焼物を表すものとする．爆発の発生のために爆発源から放射状に向かう大きな外力と密度を短時間だけ与える（図 1 参照）．この外力により，瞬間的に大きな速度が発生するので爆発が発生する様子を擬似的に表現できる．なお以下では，爆発源に与える外力の大きさを爆発の強度とする．その後，シミュレーション空間内の速度場および密度場の時間変化を文献²⁾の手法を用いて数値解析することで爆発をシミュレーションする．爆発のシミュレーションにおける流体解析の支配方程式は次式で表される．

$$\frac{d\mathbf{u}}{dt} = -(\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (2)$$

$\mathbf{u}$ は、流体の速度ベクトル、 p は圧力、 ν は動粘性係数、 $\mathbf{f}$ は外力である。式(1)、(2)は、非圧縮性条件下における N-S 方程式であり、非圧縮性流体を扱う場合のみ式(2)が成り立ち、この式は連続の式と呼ばれる。また、式(1)は速度場の時間発展を表す方程式であり、右辺第1項から、移流項、圧力項、拡散項、外力項と呼ばれている。また、式(1)により計算された速度場を用いれば、密度場の時間発展を次式により計算できる。

$$\frac{dD}{dt} = -(\mathbf{u} \cdot \nabla)D + \nu \nabla^2 D \quad (3)$$

D は流体の密度を表す。密度場は速度場と違いスカラー場であるため、圧力項および外力項は含まれない。また本研究では、2次元の爆発のコントロールにおいて、熱および浮力については考慮しない。この爆発のシミュレーションを基にユーザが指定した形状に一致するよう制御を行う。次節において、提案手法による爆発の制御方法について説明する。

3.2 予測制御

シミュレーションの各タイムステップにおいて、ユーザにより指定された目標形状と現時刻での爆発形状との差を求め、目標形状に近づくようシミュレーションの制御を行う。現時刻 t での爆発形状は、爆発源から角度 θ 方向について、密度が指定した閾値以上となっている最遠方の格子点までの距離 $R(\theta, t)$ として表す (図2参照)。なお、角度 θ は等間隔にサンプリングを行う。目標形状はユーザが画面上に直接描くことで指定し、 $R_{usr}(\theta)$ で表す (図2参照)。図3に提案法による処理の流れを示す。爆発の制御は、爆発の初期強度の設定、予測形状の算出、速度場の制御からなる (図3の赤色部分)。初期強度は、角度 θ ごとに目標形状までの距離に比例した値を以下の式にしたがって設定する。

$$I_{init}(\theta) = I_{usr} \cdot R_{usr}(\theta) \quad (4)$$

これにより目標形状に類似した形状の爆発を生成することができる。しかし、これだけでは精度よく目標形状に一致させることは困難であるため、速度場の制御を行う。この処理では、現時刻から一定時間 Δt 後の予測形状 $R_p(\theta, t + \Delta t)$ (図2の R_p) を算出して制御する。以下に予測形状の算出方法について述べる。

まず、3.1節で述べた制御を行わない場合の爆発のシミュレーションを前処理として行う。この場合、半径が徐々に大きくなる円状の爆発が発生する。この円状爆発の成長速度 (単位時間当たりの平均半径の変化率) の履歴を計測し、時間積分を行うことで、成長速度が $v(t)$ であったときの一定時間 Δt 後の爆発半径を表す関数 $R_{ref}(v(t), \Delta t)$ を

以下のように構築する。

$$R_{ref}(v(t), \Delta t) = \int_t^{t+\Delta t} v(t) dt \quad (5)$$

この式の Δt を一定間隔で変化させたときの値をグラフ化したものは、それぞれ以下の式で表される対数関数によって精度よく近似できる。

$$R_{ref}(v(t), \Delta t) = a(\Delta t) \ln(v(t)) + b(\Delta t) \quad (6)$$

ここで a 、 b はともに係数であり、 Δt の値に応じて変化する。提案手法では、 a および b を直線近似し、 $R_{ref}(v(t), \Delta t)$ をさらに以下の式により表す。

$$R_{ref}(v(t), \Delta t) = (a_0 + a_1 \Delta t) \ln(v(t)) + (b_0 + b_1 \Delta t) \quad (7)$$

式(7)で表される R_{ref} および現時刻における爆発形状 $R(\theta, t)$ を用いて、予測形状 $R_p(\theta, t + \Delta t)$ を次式により求める。

$$R_p(\theta, t + \Delta t) = R_{ref}(v(\theta, t), \Delta t) + R(\theta, t) \quad (8)$$

ここで、 $v(\theta, t)$ は現時刻での θ 方向の爆発の成長速度を表し、各タイムステップにおいて計算する。

上記の方法により得られた予測形状を目標形状と比較し、その比較結果に応じて速度場を制御する。予測形状が目標形状に達していない場合は追加速度の処理を行い、予測形状が目標形状を超えている場合は速度場の強制減少の処理を行う。

3.3 爆発の3次元化

これまでの手法により、2次元における爆発をユーザの意図した形状どおりにコントロールすることが可能となる。本節では、2次元におけるシミュレーションを3次元における爆発のシミュレーションへ拡張し、ユーザが指定した形状に一致するようその動きの制御を行う手法について述べる。3次元への拡張は文献⁶⁾の手法を利用する。この手法では、複数の2次元の流体シミュレーションを組み合わせることで3次元の流れ場を補間により生成する。そのため、前節までに述べた2次元の爆発における提案

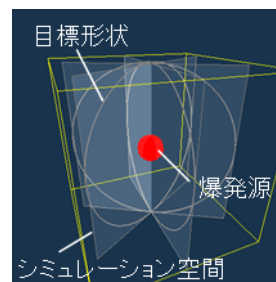
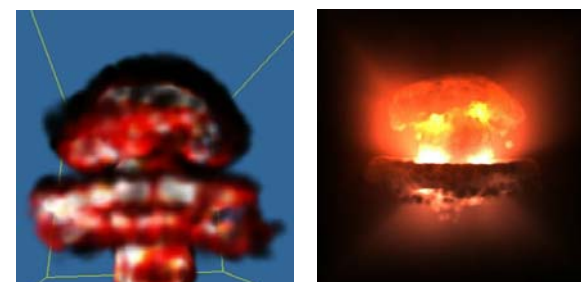


図4: 2次元平面の設定



(a) 変更前の結果 (b) 変更後の結果

図5: 比較結果

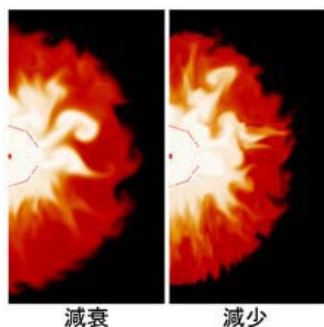


図 6：減少と減衰の比較

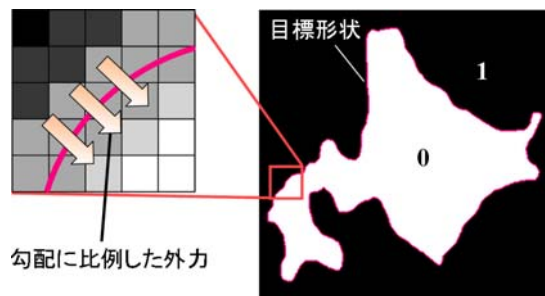


図 7：複雑形状への対応

手法をそのまま用いることができる。

3次元のシミュレーション結果のレンダリングは以下のように行う。文献¹⁾でのレンダリングでは、各ボクセルの頂点に割り付けられた密度および温度の値に応じて、各ボクセルに色付けを行っていたが、この方法ではリアリティにかけるといった問題があった。そこで、爆発の燃焼物は自身も発光することを考慮し、ボクセルの各頂点での温度の値から爆発自身が発する光を計算することで、よりリアルな爆発となるよう変更を行った。比較結果を図5に示す。(a)よりも(b)のほうが、よりリアルな爆発となっているのがわかる。

なお、これまでの提案手法の詳細については文献¹⁾を参照していただきたい。以下では文献¹⁾からの変更点について説明する。

3.4 速度場の強制減少

文献¹⁾では、速度場の制御において、予測形状と目標形状との差から算出した減衰率を速度場に対して乗算するという強制減衰なる処理を行っている。しかし、この処理では速度場の詳細な変化が失われ、乱流成分などが失われてしまうといった欠点があった。そこで、速度場の減衰から減少への変更を行った。具体的には、目標形状と予測形状との差に比例した値を各格子点に存在する速度場から減算する。減算する値は以下の式により算出する。

$$\alpha(\theta) = \alpha_a \cdot (R_{usr}(\theta) - R_p(\theta, t + \Delta t)) \quad (9)$$

ここで、 α_a は係数である。減衰と減少の比較結果を図6に示す。これにより、速度場の高周波成分が失われることがなく、微小変化を保持したまま平均的な大きさを小さく出来るので、乱流成分などの詳細形状を失うことなく爆発形状の制御を実現できる。

3.5 複雑形状への対応

文献¹⁾による手法では、爆発源から不可視な部分のある目標形状(複雑形状)については制御が不可能であった。これは、爆発形状の取得が中心からの角度 θ によって一意に決まってしまうためであった。この問題を解決するために、文献⁴⁾による手法と本手法を組み合わせることで、複雑形状についての対応を行った。この手法では、目標形状に対しポテンシャル場を設定し、その勾配の大きさに比例した外力を発生させることで目標形状どおりに制御する。これにより、複雑形状に対しても爆発を制御することが可能となる。この手法について以下で詳しく説明する。

まず、図7の右に示す目標形状に対して内部を1、外部を0とするポテンシャル場を設定する。さらに、シミュレーション空間全体にガウスフィルター処理を施すことで、目標形状付近に大きな勾配を持ったポテンシャル場(ターゲット密度)を生成する。そして、設定したポテンシャル場の勾配の大きさに比例した外力を加える。これにより、ポテンシャル場の勾配に比例した外力が、目標形状の外部から目標形状の内部に向かって発生するので、目標形状どおりに制御することが可能となる。

この手法を提案手法である予測制御と組み合わせることで、複雑形状に対する制御を目指す。具体的には、まず、複雑でない形状と同様に予測制御により爆発の制御を行う。その後、爆発が目標形状に対して一定の位置に達したところで文献⁴⁾の手法による制御を開始する。その理由は、シミュレーションの初期状態から文献⁴⁾の制御を適用すると、必要以上に大きな外力が目標形状の外部から発生し、中心からの外力との相殺により目標に達する前に速度が大幅に落ちてしまうためである。そして、適用後は予測制御との両方の制御により、目標形状への制御を行う。これにより、複雑形状に対しても爆発形状の制御が可能となる。

3.6 キーフレームアニメーション

これまでに説明した手法¹⁾では、目標形状は一つしか設定できず、複数の爆発を順に発生し、各々指定された目標形状に制御するといったことは不可能であった。そこで、目標形状をキーフレームとして、複数設定し複数回の爆発に対して各々目標形状どおりに制御する手法を提案する。以下では、このキーフレームコントロールの詳細について説明する。はじめに、ユーザは目標形状をキーフレームとして複数指定し、それぞれの目標形状について次の目標形状に切り替えるまでの時間間隔 t_c を設定する。また、目標形状を次の目標へ切り替えるタイミングを目標形状に対する爆発形状の到達割合 $(R(t)/R_{usr})$ によって設定する。その後、シミュレーションを開始する。まず図8のように最初の目標形状に対してシミュレーションおよび制御を行う。そして、爆発形状が設定した到達割合に達した時刻を基準とし、そこから t_c が経過した後目標形状を次に設定した形状に切り替え、再び切り替えた目標形状に対してシミュレーションおよび制御を行う。この処理を繰り返していくことでキーフレームコントロールを可能とする。なお、指定できる目標形状の数に制限は無い。ただし、提案法では、

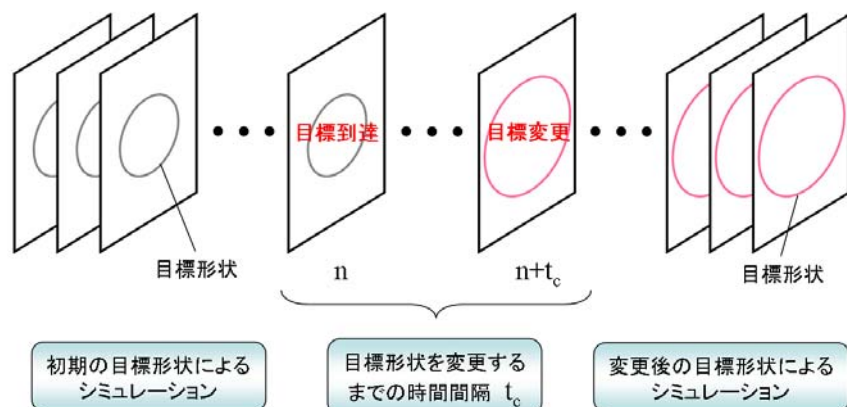


図 8：キーフレームアニメーションの概要

爆発が縮小するような制御は行うことはできない。そのため、一つ前の目標形状より小さくなるような目標形状を指定した場合は、不自然な結果となってしまう。

4. 実験結果

提案手法を適用したキーフレームアニメーションの結果を図 9 および図 10 に示す。実験環境は、CPU が Intel Core2 Duo E8400(メモリ 2GB), GPU が NVIDIA GeForce 9300 となっている。2 次元におけるシミュレーション空間は 360×360 の格子に分割した。

図 9 は、キーフレームの目標として初期形状に小さい円、最終形状に初期形状よりも大きい円を指定し、図 10 では初期形状に円型の形状、最終形状にハート型の形状を目標として指定した。図 9、図 10 ともに指定した目標形状の数は 2 つであり、目標形状を変更するまでの時間間隔 t_c は 10step である。図 9、図 10 ともに(a), (b), (c)が初期形状におけるシミュレーションであり、(d), (e), (f)が目標を変更した後のシミュレーションである。どちらの場合も指定した 2 つの形状において目標形状どおりの爆発が得られていることがわかる。

また、どちらの結果においても 1 フレームあたりのシミュレーション時間は約 1.36 秒である。

5. まとめと今後の課題

本稿では、2 次元における爆発および 3 次元における爆発のシミュレーションに対して、複数回の爆発を発生させ、各々設定された目標形状へ制御を行う手法について提案した。提案手法により、複数回の爆発に対して、ユーザの希望通りの形状に爆発をコントロールすることができ、キーフレームアニメーションの作成が可能となった。しかし、3.5 節でも述べたとおり、設定できる目標形状に制約があるため、改善が必要である。また、3 次元における爆発への適用も今後の課題としてあげられる。そのほか、本手法の炎のシミュレーションへの応用があげられる。

参考文献

- 1) 佐藤, 土橋, 山本, 安生: 流体力学に基づく爆発の動きのコントロール, Visual Computing/グラフィックスと CAD 合同シンポジウム 2009, No. 03
- 2) J. Stam 1999, Stable Fluids, In Proceeding of ACM SIGGRAPH 99, pp.121-128
- 3) A. Treuille, A. McNamara, Z. Popovic, J. Stam 2003, Keyframe Control of Smoke Simulations, In Proceeding of ACM SIGGRAPH 2003, pp. 716-723
- 4) R. Fattal, D. Lischinski 2004, Target-driven smoke animation, In Proceeding of ACM SIGGRAPH 2004, pp. 441-448
- 5) B. E. Feldman, J. F. O'Brien, O. Arikan 2003, Animating Suspended Particle Explosions, In Proceeding of ACM SIGGRAPH 2003, pp.708-715
- 6) N. Rasmussen, D. Q. Nguyen, W. Geiger, R. Fedkiw 2003, Smoke simulation for large scale phenomena, In Proceeding of ACM SIGGRAPH 2003, pp. 703-707
- 7) L. Shi, Y. Yu, Taming Liquids for Rapidly Changing Targets, In Proceeding of ACM SIGGRAPH / Eurographics symposium on Computer animation 2005, pp. 229-236

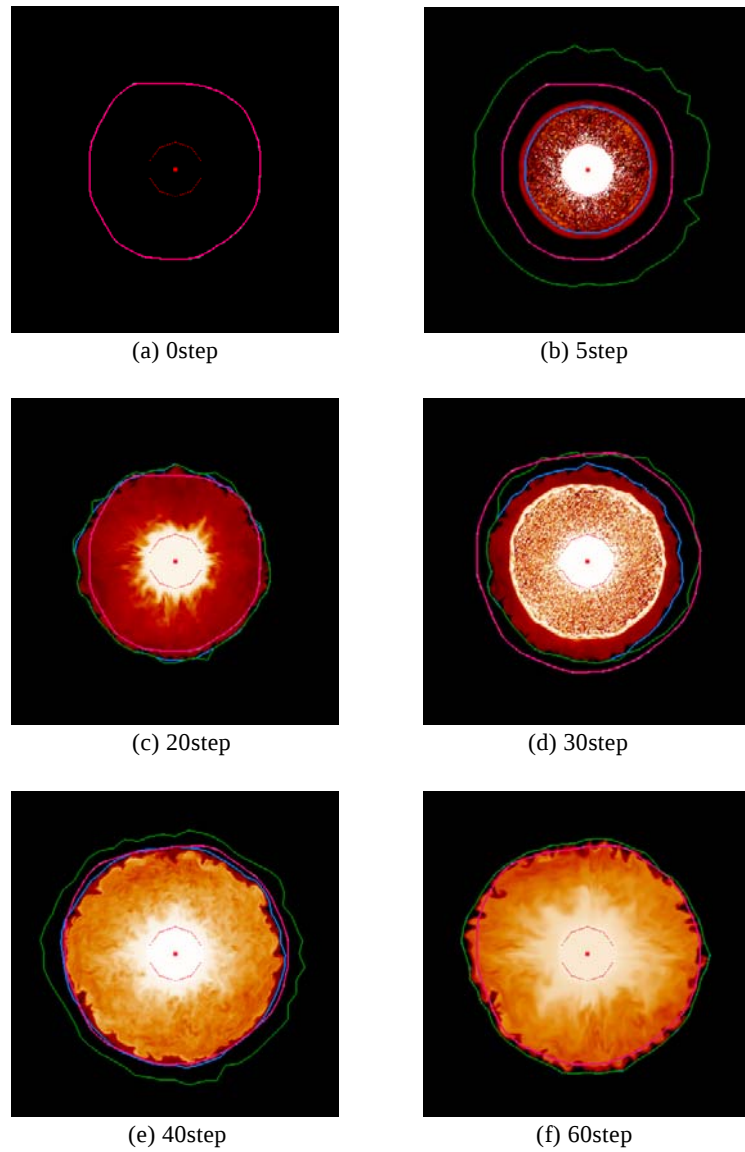


図 9 : 実験結果

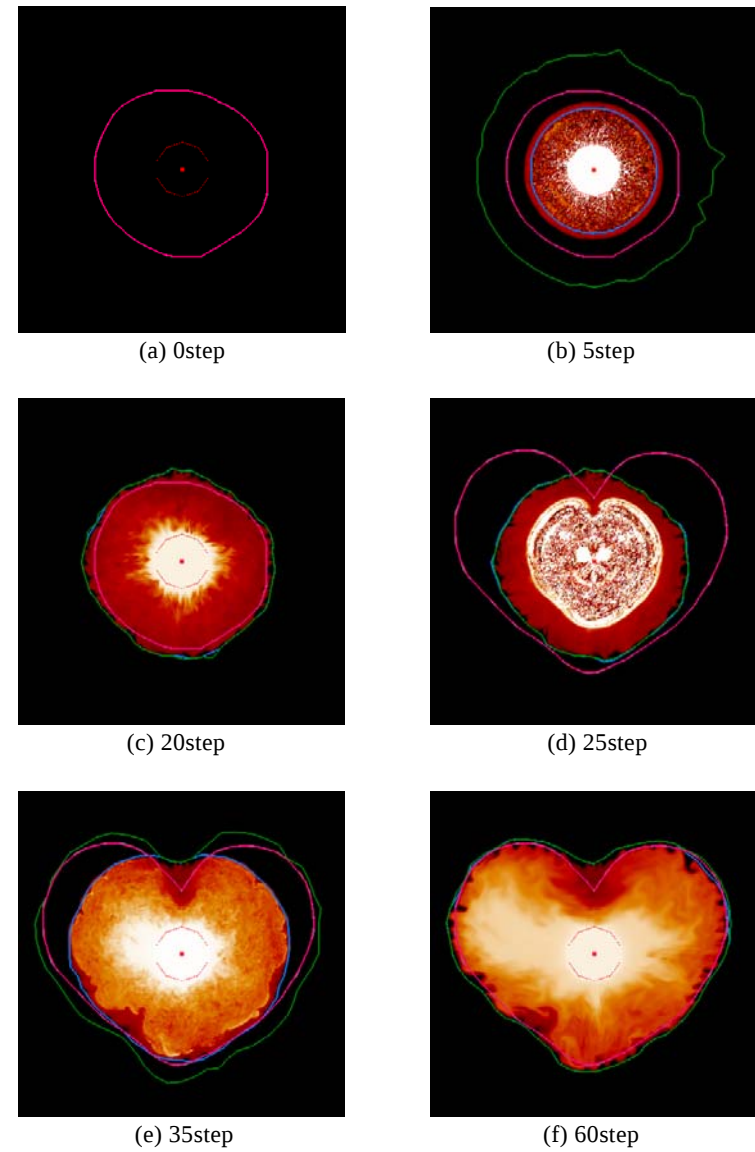


図 10 : 実験結果