

GPUを用いた Fast Marching Method の高速化の一手法

鶴賀淳樹[†] 岩崎慶[‡]和歌山大学システム工学部[†]和歌山大学システム工学部[‡]

1. はじめに

CG 分野において、シミュレーションの可視化は重要な研究分野の一つである。その中でも液体シミュレーションを可視化するためには液体界面の構築が必要であり、界面を追跡する必要がある。界面追跡の手法としてレベルセット法が用いられている[1]。レベルセット法は界面からの符号付距離をレベルセット関数として用いる。レベルセット法は、界面が変化するたびにレベルセット関数の再初期化をする必要がある。再初期化方法として、Fast Marching Method(以下 FM 法と略す)が用いられているが、FM 法は隣接データに依存する手法であるため、並列計算に適していないという問題がある。

そこで本論文では、FM 法における再初期化処理を並列化し、GPU 実装によって高速化する手法を提案する。

2. レベルセット法

レベルセット法は界面追跡の手法のひとつである。レベルセット法では、式(2.1)で表されるレベルセット関数 φ を用いて界面追跡を行う。

$$\begin{cases} \varphi(x, t) > 0 & \text{outside} \\ \varphi(x, t) = 0 & \text{interface} \\ \varphi(x, t) < 0 & \text{inside} \end{cases} \quad (2.1)$$

ここで、 \mathbf{x} はグリッド座標、 t は時刻を表す。また、レベルセット関数が正の時は液体の外部、負の時は内部を表す。

本研究ではレベルセット関数に界面からの距離を表す符号付距離を使用する。レベルセット関数の値がゼロとなる点をつなぐことで界面を構築する。しかし、レベルセット法は界面が変化するたびに符号付距離場ではなくなってしまうため、以下の再初期化方程式を解くことでレベルセット関数の再初期化を行う必要がある。

An Acceleration Technique of Fast Marching Method using GPU

[†]Junki Tsuruga, Faculty of Systems Engineering, Wakayama University

[‡]Kei Iwasaki, Faculty of Systems Engineering, Wakayama University

$$\frac{\partial \varphi}{\partial t} + S(\varphi)(|\nabla \varphi| - 1) = 0 \quad (2.2)$$

ここで、 $S(\varphi)$ は φ における液体内で-1、液体外で1の値となる符号関数である。(2.2)式を解くことでレベルセット関数の再初期化を行うことができる。

3. Fast Marching Method

3.1. Fast Marching Method の概要

レベルセット関数の再初期化処理の計算は非常に多くの時間が必要となり、計算領域の大きさに比例して計算コストが大きくなる。そこで、Sethian はレベルセット関数の再初期化を高速に行う手法として Fast Marching Method を提案した[1]。また、Yang らは、計算領域を複数に分割し、メニーコアで各領域に対し FM 法を適用させることで計算速度を向上させた[2]。FM 法は界面付近の符号付距離場を抽出し、そこから周囲へ値を広げていくことでレベルセット関数を再構築する手法である。また、符号付距離の性質が必要なのは界面付近だけなので、界面近くの格子数個分幅の帯域の符号付距離を計算すればよい。帯域幅に存在しないグリッドは液体内/液体外で定数を代入すればよい。しかし、この手法は隣接データに依存した手法であるため、並列処理に適していないという問題点が存在する。

3.2. FM 法の流れ

まず、界面が存在するグリッド(界面グリッドと定義)の符号付距離を計算し、界面グリッドを値の確定状態(確定状態のグリッドを確定グリッドと定義)とする。次に、確定グリッドの周囲のグリッドに対し、確定グリッドから符号付距離を計算し候補状態とする。候補状態のグリッドをひとつ取り出し、確定状態にする。以上の処理を候補状態のグリッドがなくなるまで繰り返すことで、帯域内のグリッドに値を広げる。

4. 提案法

4.1.FM法の問題点

FM法では、確定グリッドから周囲のグリッドの符号付距離を計算するため、GPU実装した場合1つのグリッドに対し、複数グリッドからの値の更新が同時に発生してしまうという可能性がある(図1参照)。この問題によって、界面付近のグリッドに正しい値が代入されず、誤った値が周囲に拡散してしまう。この状態でシミュレーションした場合、符号付距離をもとにした界面構築で正しい界面を表現できないという問題がある。

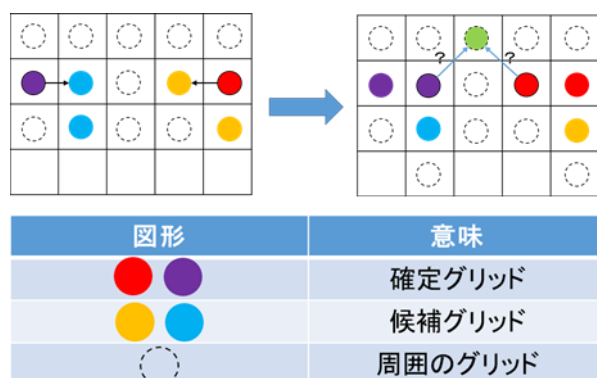


図1. 複数の確定グリッドからの同時代入

4.2.提案法の概要

FM法では、確定グリッドの処理が行われているときに周囲のグリッドの符号付距離の更新をしていた。提案法では確定グリッドに対しては何の処理も行わず、確定グリッドの周囲に存在する符号付距離の更新が必要なグリッド(これを計算グリッドと定義する)を処理の対象とする。計算グリッドが周囲に存在する確定グリッドの符号付距離を参照して符号付距離の計算を行うためGPUによる並列計算が可能となる。また、計算グリッドの周囲に確定グリッドが複数存在する場合は、計算した符号付距離の中から最少となる値を符号付距離として持つことになる。

4.3.提案法の処理の流れ

まず、界面グリッドの符号付距離場を求める。符号付距離の更新を行ったグリッドを確定状態にする。次に、確定グリッドの周辺に存在し、符号付距離の更新が必要なグリッドの符号付距離場を求め、確定グリッドとする。この処理を複数回繰り返す、帯域幅に存在しないグリッドは、液体内/液体外で値(-1.0 / 1.0)を代入することで、並列計算が可能となる。

FM法と異なり、提案法では計算グリッドから周囲の確定グリッドの値をもとにすることで計算グリッドの符号付距離の更新を行うため並列計算が可能となる。各格子点にスレッドを割り当てるため、計算時間を大幅に短縮することができる。

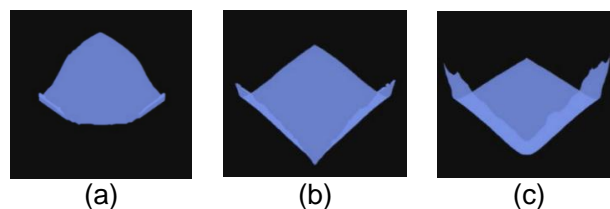


図2. 提案手法の結果画像

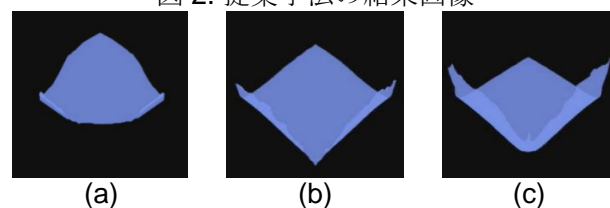


図3. FM法の結果画像

5. 結果

図2, 3に本研究の結果例を示す。ここで、図2, 3の(a)は15ステップ目、(b)は30ステップ目、(c)は45ステップ目の実行結果である。実行環境は、Intel Core i7 4790 3.60GHz、16GB RAM、NVIDIA GeForce GTX 970を搭載したPCで、格子分割数は64x64x64の空間でシミュレーションを行っている。図2が提案手法の結果画像、図3がFM法の結果画像である。1ステップ毎の再初期化計算の平均時間はFM法の場合は約90ms、提案手法の場合は約0.07msである。このことから、提案手法は約1250倍の高速化を達成した。

6. まとめ

本論文では、レベルセット関数の再初期化処理であるFM法を、CUDAを用いて並列処理する手法を提案した。今後の課題としては現在、符号付距離の計算を複数回反復させることで周囲に値を広げているが、FM法と同様に動的に収束判定を行う方法の実装が挙げられる。

参考文献

- [1]J. Sethian, Level Set Methods and Fast Marching Methods. Cambridge University Press, 1999.
- [2]J. Yang et al., A Highly Scalable Massively Parallel Fast Marching Method for the Eikonal Equation, arXiv:1502.07303v1 [physics.comp-ph].