

# 翼の三角形分割と剛体近似を用いた 紙飛行機のリアルタイムフライトシミュレーション

寺島 広† 小池 崇文†

法政大学情報科学部†

## 1. はじめに

本研究は、紙飛行機の翼を三角形分割し、分割した各部分を剛体に近似することで、リアルタイムに紙飛行機のフライトシミュレーションを行う。リアルタイムに紙飛行機のフライトシミュレーションを行うことで、ゲームエンジンに実装されていない空気力学のシミュレーションを実装するための足がかりとする。

## 2. 関連研究

紙飛行機のフライトシミュレーションは、ナビエストークス方程式（以下 NS 方程式）を直接解くか解かないかで、再現度の高さが決まってくる。吉川らは、NS 方程式を直接解くことで、紙飛行機の飛行軌道を厳密に計算している[1]。しかし、[1]のように NS 方程式を直接解き、飛行軌道を計算すると、計算コストが大きくなってしまふ。そこで梅谷らは、NS 方程式を直接解かず、翼を短冊形分割し、剛体に近似してシミュレーションを行うことで、計算コストを下げている[2]。

しかし、短冊形分割するより三角形分割の方が、より細分割できる。また細分割することで近似の精度をあげ、シミュレーションの精度を上げられる。さらに、服部らは三角形に分割するときの三角形を出来るだけ一様に分割することで、同じ数で分割したときに、形が一樣な方が、解析精度が良くなることを示し、分割された三角形が一樣になるバブルメッシュ法を提案している[3]。

本研究では、NS 方程式を直接解かず、翼を出来るだけ一様な形の三角形に分割することで、計算コストを抑えながら飛行軌道の計算精度を上げる。

## 3. 提案手法

翼を三角形に近似し、剛体として扱い、紙飛行機の運動方程式を立てて解くことでフライトシミュレーションを行う。まず、紙飛行機の翼をバブルメッシュ法で三角形に分割する。次に、三角形分割した各部分で、翼を剛体として扱える薄翼理論のモデル式を用い、揚力・抗力・ピッチングモーメントを計算する。分割した各部分をまとめることで、翼全体にかかる力を求め、運動方程式を立てる。最後に、立式した運動方程式を解くことで、紙飛行機の飛行軌道を計算できる。提案手法の概要を図1に示す。

ここで、薄翼理論のモデル式について説明する。Stengel らは、翼にかかる圧力を求めるための空力学係数のモデルを考え[4]。そのときに、使われる翼にかかる力を計算するための式が薄翼理論のモデル式である。式(1)が、薄翼理論のモデル式である。

Real-time Flight Simulation of Paper Airplane Based on Wing Triangulation and Rigid-body Approximation  
†Hiromu TERASHIMA, Takafumi KOIKE  
†Faculty of Computer and Information Sciences, Hosei University, Tokyo, Koganei-shi, 3-7-2 Kajino-chou

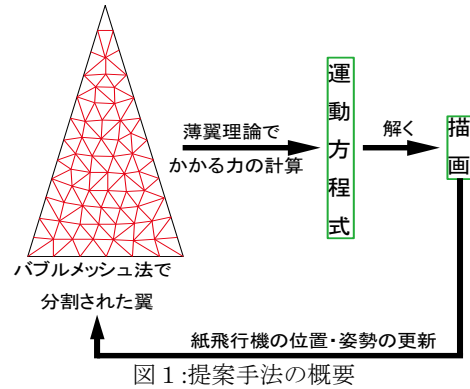


図1:提案手法の概要

## 3.1. 分割領域にかかる力

まず、バブルメッシュ法で、翼を三角形分割する。バブルメッシュ法は、図形にバブルを配置し、配置されたバブルの中心点をもとにドロネー法で三角形分割を行う。バブルメッシュ法を図形に適用したときの流れが図2である。ただし、図形の辺上の塗りつぶされたバブルは動かない。塗りつぶされていないバブルのみ図形の中を、他のバブルと重ならないように動く。

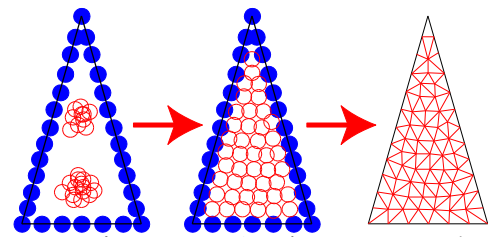


図2:バブルメッシュ法を適用するときの流れ

図2のように翼を三角形の細かい領域に分割したら、各領域でかかる力を求める。求める力は、揚力、抗力、ピッチングモーメントの三つである。この三つの力は式(1)によって求められる。

$$(f_d, f_l, \tau_z)^T = \frac{1}{2}(C_d, C_l, C_m L)^T \rho V^2 A \quad (1)$$

$f_d$ は抗力、 $f_l$ は揚力、 $\tau_z$ はピッチングモーメントである。 $C_d$ は抗力係数、 $C_l$ は揚力係数、 $C_m$ はピッチングモーメント係数である。 $\rho$ は空気密度、 $V$ は対する紙飛行機の相対的な速さ、 $A$ は進行方向から大気が流れる面の面積である。

## 3.2. 紙飛行機の運動方程式

翼を分割した各部分でかかる力を求めたら、各部分にかかる力を足し合わせて、翼全体にかかる力を求める。次に、翼全体にかかる力と重力を紙飛行機にかかる力として、運動方程式を連立する。このとき、紙飛行機の運動は、図3のような平面内の運動とし、そこに回転が加わっている。

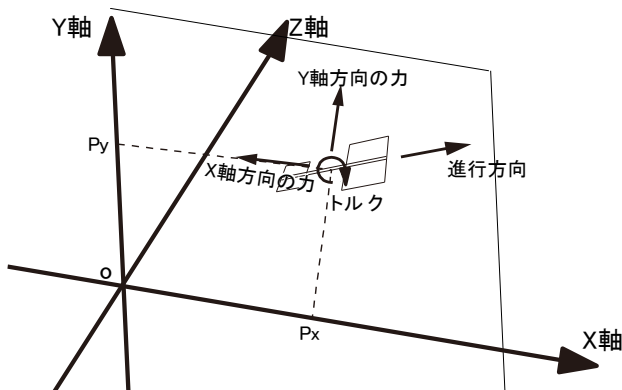


図3:運動中の紙飛行機にかかる力

図3から立てる運動方程式は式(2)~(4)の三つと分かる.

$$M\ddot{p}_x = F_x \quad (2)$$

$$M\ddot{p}_y = F_y - Mg \quad (3)$$

$$I\ddot{\beta} = \tau_z \quad (4)$$

$M$ を紙飛行機の質量,  $I$ を紙飛行機の慣性モーメントとし, 重力加速度を $g$ とする. また $p_x, p_y, \beta$ は, 左から X 軸に関する位置, Y 軸に関する位置, 紙飛行機の仰角である. さらに $F_x, F_y, \tau_z$ は, 左から X 軸方向にかかる力, Y 軸方向にかかる力, トルクである.

そして連立で運動方程式を解き, 得られた紙飛行機の位置・姿勢を描画する.

#### 4. 実験結果

提案手法を, 実飛行軌道と従来手法と比較することで, 計算精度の評価を行った.

実飛行軌道は, 紙飛行機を飛ばしている様子を撮影し, それをもとに飛行軌道を作成した. また, 紙飛行機を 3m になるように投げた時と 5m になるように投げた時で飛行軌道を用意し, それぞれの初速度と初期の仰角を撮影した動画をもとに求めた. また, 複数回投げて出来るだけ図3の XY 平面内を動く軌道を選択した. 初速度 370m/s, 初期の仰角 0 度で投げた場合と初速度 720m/s, 初期の仰角 0 度で投げた場合の二通りになった.

Stengel らは, 紙飛行機のフライトシミュレーションを提案している[5]. 今回は, そのフライトシミュレーションを従来手法として比較する. 実験結果が図4である. 図4の凡例の()内に各飛行軌道の飛行時間を示す. また, 実験は, Unity 上で実行している. また使用 PC は, dynabook R732/H である. GPU は, Intel HD Graphics 4000 で 1792MB で, CPU が, Intel Core i5-3340M - 2.70GHz で, メモリが 8192MB である.

結果をみると, 370m/s で投げた場合, 三つの飛行軌道

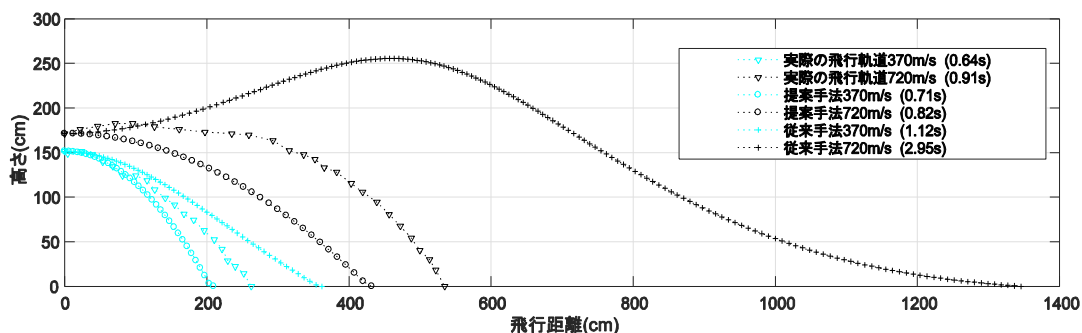


図4:各飛行軌道の実験結果

は大きな差がない. しかし, 720m/s で投げた場合, 従来手法は, 実飛行軌道と比べ, 大きく異なった軌道となった. 提案手法は実飛行軌道と比べ, 飛行軌道にあまり差が出なかった. このことから従来手法に比べて, 提案手法の方が, 紙飛行機の計算精度が高くなっていることがわかる. また, 提案手法の飛行時間は, 実紙飛行機が飛んでいた時間と比べると, 差がほとんどない. このことから, 飛行時間の再現も精度が高い.

#### 4.1. 考察

提案手法と従来手法の飛行軌道の差はいくつか理由が考えられる. 従来手法では回転の運動が慣性モーメントを求めずに行われている点や, 提案手法では各空力学係数を実測により求めた点などが考えられる.

また, 提案手法の飛行軌道と実飛行軌道に差が出る理由は, いくつか考えられる. 一つは, 紙飛行機が図3の XY 平面内でしか動かず, 実紙飛行機の動きを完全に再現できない点である. 次に, 撮影した動画から求めた実飛行軌道の初速度や初期仰角に, 誤差が生じている点である. さらに, 翼と機体の干渉なども計算していない点が考えられる.

#### 5. 結論

本研究は, 紙飛行機の飛行軌道を, 計算コストを抑ながらより正確に求めることができた. しかし, ゲームエンジンに空気力学シミュレーションを実装するには, まだ課題がある. それは, 紙飛行機の場合しか考えられない点である. 紙飛行機は, 既存の研究から空気の影響を近似により解けた. しかし, 紙飛行機以外の動きをシミュレーションする場合は, 空気そのものの動きをシミュレーションし, 圧力を計算する手順を踏む必要がある.

#### 文献

- [1] 吉川俊樹, 井ノ本健, 山川勝史, 松野謙一: “紙ヒコーキの飛行シミュレーション” 航空宇宙技術 Vol14 pp.145-152, 2015.
- [2] Nobuyuki Umetani, Yuki Koyama, Ryan Schmidt, and Takeo Igarashi: “Pteromys: Interactive Design and Optimization of Free-formed Free-flight Model Airplanes” SIGGRAPH, 2014.
- [3] 服部海斗, 横内康人, 三谷純, 西原清一: “バブル・メッシュ法によるメッシュ制御法の提案”, 情報処理学会研究報告グラフィクスと CAD(CG), 2007.
- [4] Robert F. Stengel, “Flight Dynamics”, pp.19-272, Princeton University Press Princeton and Oxford, 2004.