



HOKKAIDO UNIVERSITY

Title	流体中の運動計算のための物理シミュレーション環境 : 粒子群最適化によるダ・ヴィンチ型螺旋翼の形状決定
Author(s)	大江, 亮介; 鈴木, 育男; 山本, 雅人 他
Citation	精密工学会誌, 78(4), 348-353 https://doi.org/10.2493/jjspe.78.348
Issue Date	2012
Doc URL	https://hdl.handle.net/2115/64607
Type	journal article
File Information	78_348.pdf





流体中の運動計算のための物理シミュレーション環境*

—粒子群最適化によるダ・ヴィンチ型螺旋翼の形状決定—

大江亮介** 鈴木育男*** 山本雅人*** 古川正志***

Physical Simulation Environment for Motion Computation in Fluid
- Deciding a Shape of da Vinci's Spiral Wing by Particle Swarm Optimization -

Ryosuke OOE, Ikuro SUZUKI, Masahito YAMAMOTO and Masashi FURUKAWA

This paper describes a method of computing motion in fluid and an experiment using it. Motion of an object is computed by a physical simulation environment. It is constructed of a physical engine, which processes computation of rigid body dynamics. Computation of fluid dynamics is additionally implemented to the environment so that motion in fluid is simulated faster than other computational methods. The environment read easily the three dimensional mesh data, which is commonly used by CAD systems, then computes and visualizes motion immediately. The experiment is optimizing shapes of Leonardo da Vinci's spiral wings. Parameters of the spiral wing are optimized by a particle swarm optimization (PSO) or a sequential search so that the wing flies higher under two conditions. PSO usually find better parameters than a sequential search and do about sixty times as faster as a sequential search. The environment computes virtual motion for twenty seconds in real time of one second. The shapes given by an optimization are difference each other according to the given conditions, and proper from viewpoint of fluid dynamics.

Key words: physical engine, physical simulation, computer aided design, particle swarm optimization

1. 緒 言

本論文では、物理エンジンに流体力計算を実装した物理環境と進化計算を組み合わせた物理シミュレーションの有用性について報告する。物理エンジンは多くの物体が相互作用する多体物理演算を処理するためのソフトウェアであり、代表的なものとして ODE(Open Dynamics Engine), Havok, PhysX などが挙げられる。リアリティの高いゲームにおいて、破壊された壁の破片が飛び散ったり、衝撃を与えられた物体が飛ばされたりする効果物理と呼ばれる利用が物理エンジンの主な用途である。しかし近年では多体物理演算機能の有用性に着目し、次に示すような様々な用途で物理エンジンが用いられ始めている。

Laue らは、RoboCup 用の 3 次元物理シミュレータを開発し、AIBO の物理シミュレーションを行っている¹⁾。ここでは物理エンジンとして ODE が用いられており、剛体部品として直方体、球、円柱などの基本的な形状（プリミティブ）を用いてロボットを構成している。また、Maciel らは物理エンジン PhysX を用い、力覚フィードバックを伴う外科手術のシミュレーションを行っている²⁾。ここでは外科用器具、膜組織、血流、内臓器官を構成するために、剛体、クロスと呼ばれる布状の物体、弾性体、粒子法を用いた流体など PhysX の多くの機能が利用されている。

Sims のバーチャルクリーチャ³⁾は、仮想的な物理環境内で進化する人工生物あるいはロボットを取り扱った初期的な研究の

ひとつである。Sims と似た方法論は、物理シミュレーションと実機による動作検証を行った Lipson らの研究⁴⁾などでも用いられてきたが、近年では物理環境の構築に物理エンジンを用いた研究例⁵⁾も報告されている。

物理エンジンは多くの研究で利用されているが、水中のような流体中で動く物体の運動計算にはあまり利用されていない。たとえば PhysX では粒子法をベースにした流体を利用可能であるが、空間全域を覆うような流体のシミュレーションには多くの計算を必要とするためである。

このような背景から本論文では、物理エンジンを用いた流体中で運動する物体の物理演算を可能にすることを目的として、汎用な物理環境を提案する。物理環境は 3 次元メッシュデータから構造物を計算機内にモデリングし、流体中の運動の数値計算を行うソフトウェアシステムである。物理エンジンの多体物理演算に流体から受ける力を追加実装することにより、数値計算用の物理環境を構築する。この物理環境によって、物理エンジンを利用した従来の研究ではカバーできなかった遊泳する魚や飛行する鳥などのシミュレーションが可能となる^{10)~12)}。

流体力の計算は物体を構成する面ごとに行う。したがって、直方体や円柱などの基本的な形状だけでなく、ポリゴンによって構成された複雑な形状の物体に対しても適用可能である。そこで物理環境の有用性を検証するため、複雑な形状の螺旋翼を持つレオナルド・ダ・ヴィンチのヘリコプターの飛行シミュレーション実験を行う。この際、螺旋翼の形状を粒子群最適化(Particle Swarm Optimization: PSO)によって最適化し、与えられた条件に対して飛行能力が最大となる形状を求める。実験の結果から、流体中における物体の運動表現の有効性と、現実環境に対する物理環境の有用性を示す。

論文の構成は次の通りである。2 章では物理エンジンによる処理の概要を示した後、提案する物理環境で用いる流体力の計算

* 原稿受付 平成 23 年 9 月 9 日

** 学生会員 北海道大学大学院 情報科学研究科（北海道札幌市北区北 14 条西 9 丁目）

*** 正 会 員 北海道大学大学院 情報科学研究科

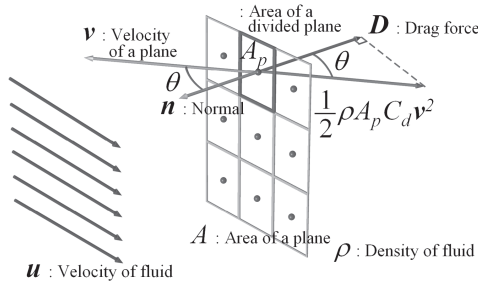


Fig.1 Drag force for moving board in uniform flow

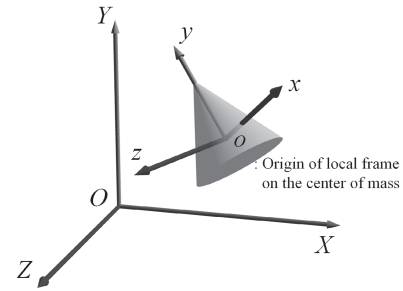


Fig.2 Global frame and local frame of reference

手法について説明する. 3 章ではダ・ヴィンチのヘリコプターの飛行実験の設定を説明する. 4 章で実験結果とその考察を述べ, 5 章で結論を述べる.

2. 物理環境の構築

2.1 物理エンジンの処理

物理エンジンは多体物理を数値積分によって処理するソフトウェアである. ロボットアームなどの制御では, ラグランジュの運動方程式などによって, 制御対象となる系全体の特性を導出する解析的な手法が多く用いられる. 一方, 物理エンジンは複数の物体の運動方程式を独立に求め, 物体間の相互作用を外力項として扱う数値的な手法を用いる. ゆえに物理エンジンでは, シミュレーション中にロボットが損傷を受けて分解したり, 逆に複数の部品が結合したりして構造が変化するという現象を容易に取り扱うことが可能である.

ここでは例として, ほとんどの物理エンジンで取り扱うことが可能な剛体の運動について簡単に述べる. 3 次元空間上に存在する剛体の状態は, 重心点の位置 p , 速度 v , 姿勢を表す回転行列 R , 回転速度 ω で表現できる. p , v , ω は 3 次元ベクトル, R は 3 行 3 列の行列である. 物理エンジンでは, 剛体ごとに式(1)の運動方程式を数値積分することで状態の更新を行う.

$$m \frac{dv(t)}{dt} = F(t), \quad \frac{dL(t)}{dt} = \tau(t) \quad (1)$$

ここに, m は剛体の質量, t は時間, L は角運動量, F は剛体に働く外力, τ はトルクである. 慣性テンソルを I とすれば, 式(2)を用いて角運動量 L から回転速度 ω を求めることができる.

$$\omega(t) = I^{-1}(t)L(t) \quad (2)$$

物理エンジンでは弾性・非弾性衝突, 摩擦, 関節による制約などを剛体に働く外力とトルクに変換し, 式(1)に作用させることで物体間の相互作用を計算する. より詳細な計算理論に関しては文献 6)~8)などで参照できる.

2.2 流体力の計算

2.2.1 計算方法

物体が流体から受ける力を計算する簡易的な方法として, 準定常的手法がある. 準定常的手法では, 周囲の流体の状態変化を考慮せず, 物体の姿勢と速度のみに従う流体力が働くことと仮定する. 有限要素法や粒子法などの流体解析手法と比較して精度は落ちるが, 計算量が少ない利点がある. ただし, 一般的な形状以外の物体に対して適用する場合には, 事前に解析的, あるいは実験的に係数などを求めておく必要がある. 本研究では, 任意の形状に対して適用が可能であり, 事前解析を必要としない流体力計算手法を提案する.

流体力の計算は, 一様流中に配置された平板が流体から受け

る抗力を基に行う. たとえば, 密度が ρ であり, 流速ベクトル u を持つ一様流に対して, 面積が A の平板が u に対して垂直に存在するとき, u の 2 乗と A に比例する抗力 D が平板の重心に働く. ただし, 一般的な状況では, 平板は並進運動と回転運動を合わせた複雑な運動をするため, u に対する相対速度が平板上の位置によって異なる. そこで, 図 1 のように平板を等面積の微小要素に分割し, 面積 A_p の各要素ごとに抗力 D を求め, 平板全体に働く抗力を計算する. D は微小要素の法線ベクトル n と平行なベクトルであり, その大きさ D は式(3)で計算される.

$$D = \frac{1}{2} \rho A_p C_d v^2 \cos \theta \quad (3)$$

ここに, C_d は抗力係数, v は流速 u に対する相対速度ベクトル, θ は v と n がなす角である.

複雑な形状の物体に対して上記の計算方法を適用する場合, 物体を構成するポリゴン等を等面積の微小ポリゴンに分割し, 各微小ポリゴンの重心に働く抗力を求める. この際, 本来ならば形状によって抗力係数 C_d を適切に求める必要がある. しかし任意の物体について適切な C_d を求めることは困難であるため, 本研究では一般的な値である $C_d = 1.0$ を用いた. また, 水中などでは流体の密度が大きいため, アルキメデスの原理より浮力を計算する必要がある. しかし本研究では密度の小さい空気中の運動を扱うため浮力は考慮していない.

本計算手法は流体の状態変化を無視しているため, 計算を行う対象によっては誤差が大きくなることが予想される. しかし, 有限要素法を用いた流体解析ソフトウェア ANSYS で流体中の平板の状態を計算した結果, 抗力係数 C_d を適切に設定した場合, 本手法との精度差が 4% 以内であった¹⁴⁾. この結果から, C_d の設定を適切に行えば十分な精度で運動計算が可能であるといえる.

2.2.2 物理エンジンへの適用

すでに述べたように, 物理エンジンは物体間の相互作用を外力とトルクに変換し, 式(1)の形式で物体の運動に反映させる. 提案する流体力計算も同様に, 流体から受ける力を外力とトルクに変換して運動に反映させる. ここでは, 物体のローカル座標系を用いた具体的な計算手順を説明する.

物体が存在する空間の座標系をグローバル座標系, 物体の重心を原点とし, 物体に張り付いて移動する座標系をローカル座標系と呼ぶことにする. たとえば図 2 のように物体が存在するとき, 大文字で示す座標系がグローバル座標系, 小文字で示す座標系がローカル座標系である. ローカル座標系は物体とともに移動するので, 物体の形状が運動によって変化しないとき, ローカル座標系から見た物体上の任意の点の座標と, 任意の面の法線ベクトルは変化しない. したがって, 面の法線ベクトルを必要とする提案手法は, ローカル座標系を用いて計算することで計算量を低減できる.

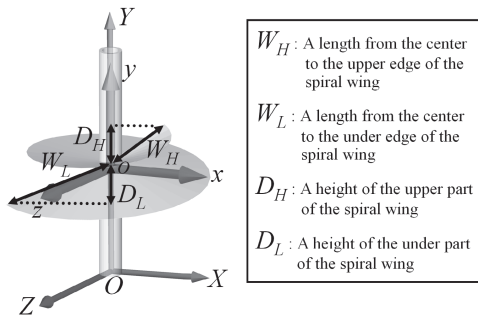


Fig.3 Spiral wing modeled by triangle polygons

ある物体の流体力を計算する際には、まず初めに流速ベクトル $\mathbf{u}$ に対する物体のローカル座標系から見た重心の相対速度 $\mathbf{v}_L(t)$ 、重心回りの回転速度 $\boldsymbol{\omega}_L(t)$ を式(4)により求める。

$$\mathbf{v}_L(t) = \mathbf{R}^T(t)\{\mathbf{v}(t) - \mathbf{u}\}, \quad \boldsymbol{\omega}_L(t) = \mathbf{R}^T(t)\boldsymbol{\omega}(t) \quad (4)$$

ここに、 $\mathbf{R}^T$ は物体の回転行列 $\mathbf{R}$ の転置行列、 $\mathbf{v}$ 、 $\boldsymbol{\omega}$ はそれぞれグローバル座標系から見た物体の重心の速度、回転速度である。

次に、分割された各微小ポリゴンの重心に働く流体力を計算する。ローカル座標系から見た微小ポリゴン i の重心位置を $\mathbf{r}_i$ とすると、点 $\mathbf{r}_i$ の速度 $\mathbf{v}_i$ は式(5)で計算される。

$$\mathbf{v}_i(t) = \mathbf{v}_L(t) + \boldsymbol{\omega}_L(t) \times \mathbf{r}_i \quad (5)$$

ただし、 $\times$ はベクトルの外積を表す演算子である。

提案手法では、物体上の面が流体から押される力のみを考慮している。そこで、 $\mathbf{v}_i$ と微小ポリゴンの法線ベクトル $\mathbf{n}_i$ の内積 $\mathbf{v}_i \cdot \mathbf{n}_i$ が正のときのみ、流体力 $\mathbf{D}_i$ を式(6)によって計算する。

$$\mathbf{D}_i(t) = \left\{ -\frac{1}{2} \rho A_i C_d |\mathbf{v}_i(t)| \mathbf{v}_i(t) \cdot \mathbf{n}_i \right\} \mathbf{n}_i \quad (6)$$

ここに、 A_i は微小ポリゴンの面積である。

全ての微小ポリゴンに対してローカル座標系から見た流体力の計算を行った後、式(7)のようにグローバル座標系から見た外力 $\mathbf{F}$ とトルク $\boldsymbol{\tau}$ を計算し、物体に作用させる。

$$\mathbf{F}(t) = \mathbf{R}(t) \sum_i \mathbf{D}_i(t), \quad \boldsymbol{\tau}(t) = \mathbf{R}(t) \sum_i \mathbf{r}_i \times \mathbf{D}_i(t) \quad (7)$$

以上の手順で提案手法を物理エンジンに適用して物理環境を構築できる。提案する流体力計算手法はポリゴンを用いた3Dモデルや物理エンジンとの親和性が高く、少ない計算量で流体中の物体の運動を計算可能である。

3. 螺旋翼の飛行パラメータ探索実験

3.1 螺旋翼のモデル

ダ・ヴィンチのヘリコプターとは、レオナルド・ダ・ヴィンチが手稿にスケッチした螺旋形の回転翼を持つ構造物を指す。この螺旋形の回転翼の飛行性能は、流体力学の知識を用いて直感的に評価することができる。しかし、3次元空間でどのような飛行を行うかを知るためには、動的な有限要素法などを用いた数値計算、または実機による飛行試験が必要である。そこで、螺旋翼をモデル化して構築した物理環境下で飛行させ、他の解析手法に対する優位性の検証を行う。

図3にモデル化された螺旋翼を示す。長さ1.4m、外径0.05m、内径0.03mの円筒形の軸に巻きつくように、厚さ0.001mの螺旋翼が固定されている。螺旋翼の外周は軸方向に向かって伸びる螺旋であり、ダ・ヴィンチのスケッチと同様に翼と軸が一点で

接合する。密度は軸と螺旋翼ともに 1000 kg/m^3 であり、水や比較的重い木材と同程度である。グローバル座標系は螺旋翼の軸底部が原点となるように設定し、ローカル座標系は螺旋翼全体の重心位置が原点となるように設定した。

軸と螺旋翼は、合計604個の三角形ポリゴンによってモデル化されている。これらのうち、空気から受ける力が無視できる軸の内側部分と、軸と螺旋翼の接合部分のポリゴンを除く454個のポリゴンについて流体力計算を行う。各ポリゴンを9等分して流体力計算を行ったため、微小時間ごとに式(6)を4086回計算する必要がある。

パラメータ W_H 、 W_L は軸の中心から螺旋翼の上端、下端までの距離、 D_H 、 D_L は y 軸方向の距離である。よって、 W_H 、 W_L が大きいほど螺旋翼の上下各部の表面積が大きくなり、 D_H 、 D_L が大きいほど螺旋の傾きが急勾配となる。図3はダ・ヴィンチのスケッチに近くなるよう $D_H = 0.2 \text{ m}$ 、 $D_L = 0.2 \text{ m}$ 、 $W_H = 0.4 \text{ m}$ 、 $W_L = 0.7 \text{ m}$ とした場合である。これら4つのパラメータを変化させることにより、螺旋翼の形状を様々に変化させることができる。ただし螺旋翼の構造的制約により、 $D_H \leq W_H$ 、 $D_L \leq W_L$ となる必要がある。

3.2 実験方法

実験の目的は、螺旋翼の4つの形状パラメータ D_H 、 D_L 、 W_H 、 W_L を様々に変化させて飛行シミュレーションを行い、最も上昇能力の高い形状を導き出すことである。パラメータ探索の方法は、各パラメータをそれぞれ0.1から0.7まで0.025刻みで変化させる全数探索方法と、粒子群最適化(PSO)による探索の2種類を独立に行い、結果を比較する。PSOのアルゴリズムとパラメータ設定は3.3節で述べる。

PSOでは $D_H \leq W_H$ 、 $D_L \leq W_L$ を満たさない場合、 W_H を D_H に、あるいは W_L を D_L に修正した後にシミュレーションを行う。一方、全数探索方法では、 $D_H \leq W_H$ 、 $D_L \leq W_L$ を満たさない組み合わせは除外して、合計105625パターンでのシミュレーションを行う。

上昇能力の評価方法は、軸の底部が接地した静止状態からシミュレーションを開始し、螺旋翼に対して外部から反時計回りにトルク $\tau = 40 \text{ N} \cdot \text{m}$ を与えた続けたときの最大高度とする。ただし、異なる条件に対して異なるパラメータが得られることを確認するため、以下の2通りの終了条件についてシミュレーションを行う。

- (a) 10秒間飛行を行った時点
- (b) 回転数が100回となった時点

シミュレーションにおける数値積分の微小時間は $1/120$ 秒とする。よって、たとえば(a)の場合、1回の飛行実験につき物理計算と流体力計算のサブルーチンを1200回実行する必要がある。

なお、全ての飛行実験には Athlon II X4 640 プロセッサを搭載した一般用途の計算機を用いた。3.0GHzのCPUコアを4つ持つクアッドコアの計算機であるが、そのうちの1つのCPUコアのみを用いてプログラムを実行した。

3.3 PSOによるパラメータ探索

本実験におけるパラメータ探索では、パラメータと飛行高度の関係を表す関数が未知であり、かつ単峰性とは限らないため、関数の勾配を利用した探索手法を用いることはできない。一方、螺旋翼の形状パターンと飛行高度の関係性を考慮すれば、極めて複雑な多峰性関数であるとは考えにくい。そこで、遺伝的アルゴリズムなどと比較して高い収束性を持つPSOを用いた。

PSOはKennedyら⁹⁾によって提案された探索手法であり、複

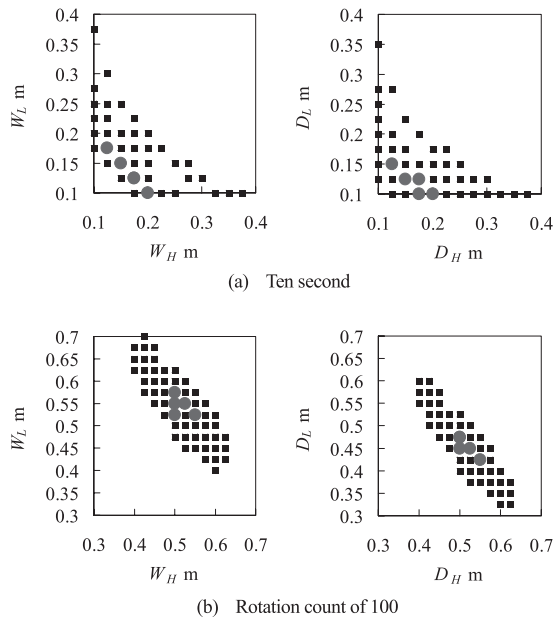


Fig.4 Parameters found in the sequential search
(circles of top 5, squares of ranking 6th – 50th)

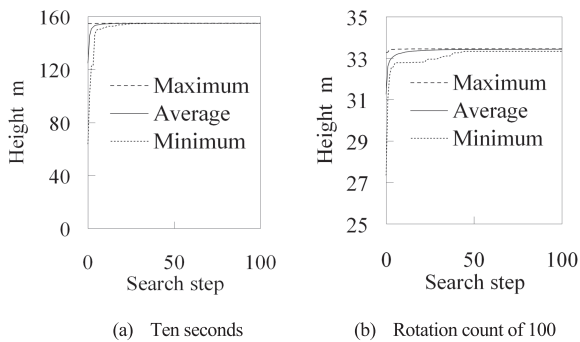


Fig.5 Average evaluations in experiments of 50 times

数の粒子が多次元の解空間内を動き回って探索を行う。各粒子は解を表す位置ベクトル $\mathbf{x}$ 、速度ベクトル $\mathbf{v}$ 、現在までに自身が発見した最良の評価を与える位置ベクトル $\mathbf{p}_d$ を持つ。また、群れ全体が現在までに発見した最良の評価を与える位置ベクトル $\mathbf{p}_g$ を全粒子が共有している。各粒子の1ステップ前の速度を $\mathbf{v}_{old}$ 、位置を $\mathbf{x}_{old}$ とすると、1ステップが経過した後の粒子の速度 $\mathbf{v}$ 、位置 $\mathbf{x}$ はそれぞれ、式(8)~(9)によって更新される。

$$\mathbf{v} = w\mathbf{v}_{old} + c_1 r_1 (\mathbf{p}_d - \mathbf{x}_{old}) + c_2 r_2 (\mathbf{p}_g - \mathbf{x}_{old}) \quad (8)$$

$$\mathbf{x} = \mathbf{x}_{old} + \mathbf{v} \quad (9)$$

ここに、 r_1 と r_2 は $[0, 1]$ の乱数である。 w 、 c_1 、 c_2 は係数であり、本実験では $w = 0.3$ 、 $c_1 = 2$ 、 $c_2 = 2$ を用いた。

式(9)を用いて解の探索を行うと、パラメータの値が $[0.1, 0.7]$ 以外となる場合がある。その場合、0.1 未満であれば 0.1 に、0.7 より大きければ 0.7 に変更して飛行高度の評価を行った。粒子数は 20 とし、100 ステップが経過した後に探索を終了した。

4. パラメータ探索実験の結果と考察

4.1 全数探索による探索結果

まず初めに、4 つのパラメータ D_H 、 D_L 、 W_H 、 W_L をそれぞれ 0.1 から 0.7 まで 0.025 ずつ変化させた場合の実験結果を示す。

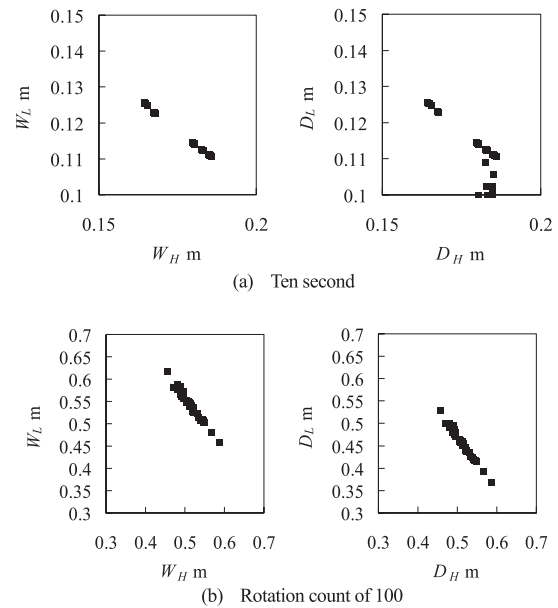


Fig.6 Parameters found in the PSO search

飛行実験に要した計算時間は、(a)の場合が 51402 秒、(b)の場合が 51298 秒であった。(a)の場合は物理環境における時間で 10 秒間の飛行を行っているため、計算時間 1 秒当たり 20.5 秒程度の飛行を計算できていることになる。

探索したパラメータの組のうち、飛行高度が高かった上位 50 パターンを (D_H, D_L) 、 (W_H, W_L) に分け、2 次元平面にプロットした結果を図 4 に示す。ただし、上位 5 パターンを丸、それ以外の 45 パターンを四角でプロットしている。(a)の場合、パラメータの値は全て 0.2 以下の小さな値となるときに飛行高度が高くなっている。一方、(b)の場合はパラメータの値が 0.4 から 0.6 程度の比較的大きな値で飛行高度が高くなっている。(a)の場合に最大飛行高度 153.174m を得たパラメータは $(D_H, D_L, W_H, W_L) = (0.175, 0.125, 0.175, 0.125)$ であり、(b)の場合に最大飛行高度 33.414m を得たパラメータは $(D_H, D_L, W_H, W_L) = (0.525, 0.45, 0.525, 0.55)$ であった。

4.2 PSO による探索結果

PSO による探索を(a)、(b)の各終了条件に対して合計 50 回ずつ行った。1 回の探索に要した平均計算時間は、(a)の場合が 825 秒、(b)の場合が 882 秒であった。

各探索ステップまでに発見した最大飛行高度について、50 回の探索の最大値、平均値、最小値を求めた結果を図 5 に示す。(a)、(b)の場合ともに、探索開始時の平均値と最小値は全数探索によって得られた最大飛行高度を下回っている。しかし、探索が進むにつれて値は増加してゆき、(a)の場合では 20 ステップ前後で平均値、最小値ともに全数探索の最大飛行高度を超える。(b)の場合では 40 ステップ前後で平均値が全数探索の最大飛行高度を超えるが、最小値は探索を終了する 100 ステップ目においてもやや低い値のままであった。

50 回の各探索について、100 ステップ目の最良パラメータ (D_H, D_L) 、 (W_H, W_L) を 2 次元平面にプロットしたグラフを図 6 に示す。(a)、(b)の場合ともに、探索の結果得られた最良パラメータは、図 4 でプロットした領域の一部分に集中して分布している。50 回の探索で発見したうちの最良パラメータは、(a)の場合が最大飛行高度 154.96m を得る $(D_H, D_L, W_H, W_L) = (0.18, 0.115, 0.18,$

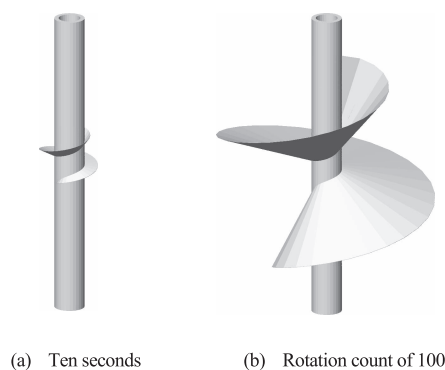


Fig.7 Optimal shape of spiral wings

0.115)であり、(b)の場合が最大飛行高度 33.456m を得る(D_H, D_L, W_H, W_L) = (0.52, 0.447, 0.52, 0.535)であった。

4.3 実験結果の考察

図 7 に示すのは、(a)、(b)の各終了条件に対し PSO による探索によって得られた最良の形状である。図からわかるように、(a)の場合は螺旋翼が非常に小さく、逆に(b)の場合は比較的大型である。またどちらの螺旋翼も、図 3 に示したダ・ヴィンチのスケッチどおりの螺旋翼よりも急峻な傾きである。図 8 に螺旋翼の状態を 0.25 秒ごとに描画したスナップショットを示す(動画を 13)に示す)。スナップショットより、安定飛行時の上昇速度は(a)の場合の方が高速である。

回転する螺旋翼上の任意の点に力が加わったとき、その力は螺旋翼の運動を変化させる力としてはたらくだけでなく、回転を妨げるような抗トルクを発生させる。この抗トルクは力の大きさと回転軸からの距離に比例する。したがって、螺旋翼に一定の力を作用させるとき、回転軸に近い点ほど抗トルクが小さくなる、すなわち抗トルクに対する運動の効率が低い。

(a)の場合、運動の効率が低い回転軸付近にのみ螺旋翼が存在し、高速回転を行うような形状が最も飛行高度が高くなる。一方(b)の場合は回転数が制限されているため、螺旋翼の表面積をできる限り大きくし、回転速度に対して最大の上昇力を生み出すような形状が最も適している。全数探索、PSO による探索ともにこの条件に見合うパラメータを導出することが目的であり、探索によって条件を満たすパラメータが得られている。このことから、定性的な評価ではあるが、提案する物理環境が流体中の運動を適切に表現しているといえる。

上記のように、流体力学の知識を用いればヒューリスティクスによってある程度適切なパラメータの予想は可能である。ただし、実際に螺旋翼の傾きがどの程度であればよいのか、あるいは状態がどのように変化してゆくのかを知るためには、正確な数値計算、または実機による動的な解析を行わなければならない。

3次元空間における流体中の物体の運動を有限要素法や粒子法などで計算する場合、計算精度は非常に高いものの、計算量は膨大となる。計算機の性能が向上した現在の超高速計算機を用いても、本実験のように 20 秒間のシミュレーションを 1 秒で計算することは難しい。

実機を用いた解析は最も確実な手段であるが、実機を製造するためのコストを要する。今回の飛行実験のようにパラメータを様々に変化させる場合は、その都度実機を製造し直さなければならない。一方、提案する物理環境はメッシュデータから剛体を自動生成し、即座にシミュレーションを実行可能である。

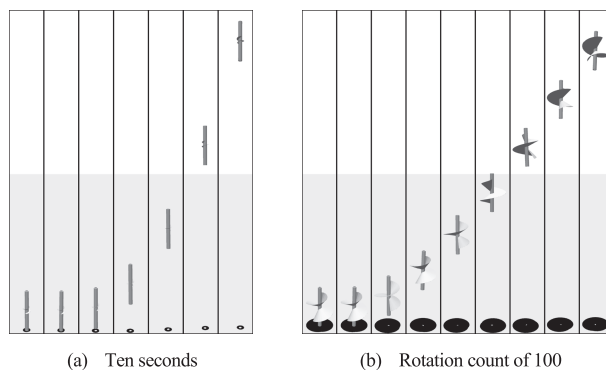


Fig.8 Flight of optimal shape of spiral wings

PSO を用いた探索では、全数探索に対して約 60 倍程度高速に探索が終了するにもかかわらず、平均的には全数探索より良いパラメータを発見できている。パラメータの刻み幅をさらに細かくすれば、全数探索の方が良いパラメータを発見できるかもしれない。しかし、その場合はより多くの試行回数が必要となるため現実的ではない。PSO による探索のようにあらかじめ想定されるパラメータセット以外のパラメータで計算を行う場合でも、提案する物理環境は事前解析などを必要とせずに計算を実行できる。

以上のように、提案する物理環境は、直感的に正しいと感じられる程度の流体中の物体の運動表現が可能であり、工業製品の設計でも用いられるメッシュデータとの相性も良い。また、有限要素法などの手法と比較して、計算速度が非常に速いという特徴がある。このような特徴から、提案する物理環境はより正確な計算や実機による試験の前段階で、明らかな不良設計の排除や、妥当なパラメータ範囲を試行錯誤的に絞り込むような用途に適している。しかし、数値的な精度の保証がないため、誤差の定量的評価が今後の課題として挙げられる。

5. 結 言

本論文では、物理エンジンに流体力計算を導入した物理環境の構築と有効性の検証を目的とし、以下のことを行った。

- (1) 物理エンジンの剛体計算処理に対して、物理エンジンに特化した流体力計算法を追加実装し、数値計算用システムである物理環境を構築した。
- (2) ダ・ヴィンチのヘリコプターの螺旋翼の形状を変化させ、飛行能力の高い形状を求めた。
- (3) 提案する物理環境が、複雑なメッシュ構造物の 3 次元動的シミュレーションにおいて、実時間の 20 倍程度の高速計算が可能であることを示した。

本研究は、人工生命分野における先に述べた人工生物の鳥、昆虫の飛翔や人工魚の遊泳へ向けた、あり得るかもしれない生物の発見と飛翔方法の解析、エンターテインメント分野におけるアニメーションの自動生成へと応用可能である。今後、これらについても報告を行いたい。なお、本研究は、平成 22~24 年度科学研究費補助金(研究課題名: 力学的プロセスによる生命行動の創発と認識, 課題番号: 2236099)の下で行ったものである。

参 考 文 献

- 1) T. Laue, K. Spiess and T. Röfer: SimRobot – A General Physical Robot

- Simulator and Its Application in RoboCup, Lecture Notes in Computer Science, 4020, (2006) 173.
- 2) A. Maciel, T. Halic, Z. Lu, L. P. Nedel and S. De: Using the PhysX engine for physics-based virtual surgery with force feedback, The International Journal of Medical Robotics and Computer Assisted Surgery, 5, 3, (2009) 341.
- 3) K.Sims: Evolving virtual creatures, SIGGRAPH '94 Proceedings of the 21st annual conference on Computer graphics and interactive techniques, (1994) 15.
- 4) H. Lipson and J. Pollack: Automatic design and manufacture of robotic lifeforms, Nature, 406, (2000) 974.
- 5) J. Rieffel, F. Saunders, S. Nadimpalli, H. Zhou, S. Hassoun, J. Rife and B. Trimmer: Evolving soft robotic locomotion in PhysX, GECCO '09 Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference (2009) 2499.
- 6) D. Baraff: Physically based modeling: Rigid body simulation, SIGGRAPH Course Notes, (2001).
- 7) R. Baber: Rigid Body Simulation, (2006)
<http://www.dcs.warwick.ac.uk/~rahil/files/RigidBodySimulation.pdf>
accessed 2011.11.7.
- 8) T. Preclik, K. Iglberger and U. R de: Iterative rigid multibody dynamics, (2008)
http://www10.informatik.uni-erlangen.de/Publications/Papers/2009/PreclikIglberger_Multibody09.pdf accessed 2011.11.7.
- 9) J.Kennedy and R.Eberhart: Particle Swarm Optimization, IEEE International Conference on Neural Networks, 4, (1995) 1942.
- 10) R. Ooe, I. Suzuki, M. Yamamoto and M. Furukawa: Research on the Behavior of Flying Creatures in Virtual Environment, 2010 JSPE Spring Meeting, (2010) D79 (in Japanese).
- 11) R. Ooe, I. Suzuki, M. Yamamoto and M. Furukawa: Experiments of Flying Creature's Flights by Flapping in a Virtual Environment, JSME Robomec 2010, (2010) 2P1-F27 (in Japanese).
- 12) R. Ooe, I. Suzuki, M. Yamamoto and M. Furukawa: A Flight of a Butterfly Model Based on the Physical Law, 2010 JSPE Autumn Meeting, (2010) 699 (in Japanese).
- 13) <http://junji.complex.eng.hokudai.ac.jp/researches/physics-modeling/movies/ooe/davincispiralwing/> accessed 2011.11.7.
- 14) M. Furukawa, M. Morinaga, M. Watanabe, I. Suzuki and M. Yamamoto: Artificial Flying Creature by Flapping, 2010 IEEE International Conference on Systems, Man and Cybernetics, (2010) 624.