

# 木の葉の落下運動の 2次元シミュレーション

大阪工業大学  
情報科学部 情報システム学科  
学生番号 B07-026  
木村 悠哉

平成23年2月15日

# 目次

§1	序論	3
§§1-1	本研究の目的	3
§§1-2	本論文の構成	4
§2	数値実験のモデル	5
§§2-1	木の葉のモデル	5
§§2-2	空気抵抗	5
§§2-3	モーメント	6
§§2-4	運動方程式	6
§3	数値実験方法	8
§§3-1	プログラムの作成	8
§§3-2	ルンゲクッタ法	8
§4	実験結果と考察	9
§§4-1	落下軌道	9
§§4-2	角速度の変化	13
§§4-3	横風を加えた場合	16
§5	アニメーション化	21
§§5-1	アニメーション化とその結果	21
§§5-2	アニメーション化における課題点と改善点	24
	参考文献	25
	付録	25
A	ルンゲクッタ法のアルゴリズム	25

## § 1 序論

### §§ 1-1 本研究の目的

本研究は、運動方程式をプログラミングを用いて解くことで、木の葉の落下運動を予測することを目的とする。木の葉の落下運動は本来、木の葉の形状、重さ、風、空気の温度や層等様々な要因に影響されて、周囲の微妙な変化で大きく軌道を変化させるものであり、カオス理論に属するものである。本研究では与える初期値の種類や周辺要素を絞り、シミュレーションを2次元に限定した上で、初期値や周辺要素の変化がどれほど木の葉の落下運動に影響するのかも踏まえて考察する。

さらに、シミュレーションを応用した iPad アプリケーションを作成することを最終目標とする。iPad とは、2010 年 4 月に Apple から発売されたタブレットコンピュータである。(図 1) マルチタッチスクリーン方式のインターフェイスを備えており、指で触れたり、指でなぞることで操作することが可能である。本研究で iPad を選んだ理由としては、iPad をプラットフォームとすることで、ゲーム等に应用した場合、指の動きで風の方向や強さを決める等、操作方法が体感的になると考えたからである。



図 1: iPad

## §§ 1-2 本論文の構成

§2では実験の準備として、実験において使用する木の葉のモデルについての説明をしていると共に、運動方程式についての考察を行う。また、運動方程式を立てる際に必要となる空気抵抗とモーメントに関する定義及び考察を行う。§3では実験方法について、作成したプログラムについてや、プログラムを作成するにあたって利用したものについての説明をしているさらに、運動方程式の数値解を求めるために必要なルンゲクッタ法に関する説明もしている。§4及び§5では様々なシミュレーション結果の提示する。落下軌道をグラフ化するだけでなく、その落下軌道が得られた理由や、落下軌道と回転の関係についても詳しく考察する。また、グラフだけではわかりにくいため、シミュレーション結果をアニメーション化し、動的に観察する方法及び結果についても考察する。

## §2 数値実験のモデル

### §§2-1 木の葉のモデル

シミュレーションを行う前に、木の葉のモデルを定義する。モデルを定義するにあたって文献[1]を参考にした。

まずは  $x$  軸について、右方向を正とする。また、 $z$  軸について落下方向を負とする。これは、重力  $-g$  を受けるためである。また、空気抵抗 (§§2-2) を受けると共に、モーメント (§§2-3) によって回転運動するものとする。外力である横風については §§4-3 にて詳しく考察する。今回の研究では、木の葉は長方形であり、密度は一樣とする。また、厚みは考えないが縦横どちらにも一切曲がらないものとする。木の葉は短辺 ( $b$ ) の中点を中心として回転し、長辺 ( $a$ ) の中点を中心として回転することはないと定義すると共に、短辺 ( $b$ ) 側から見た落下運動の軌道をシミュレーションする。(参照:図2, 図3)

$a > b > 0$  とするため、各辺に対する慣性モーメントは以下のように定義する。

$$I_{ab} \equiv \frac{m}{12}(a^2 + b^2) > I_a \equiv \frac{m}{12}a^2 \gg I_b \equiv \frac{m}{12}b^2 \quad (1)$$

つまり、木の葉は短辺 ( $b$ ) の中点を中心として回転し、長辺 ( $a$ ) の中点を中心として回転することや、木の葉の中点を中心として斜めに回転することはほぼないと定義すると共に、短辺 ( $b$ ) 側から見た落下運動の軌道のシミュレーションを行う。

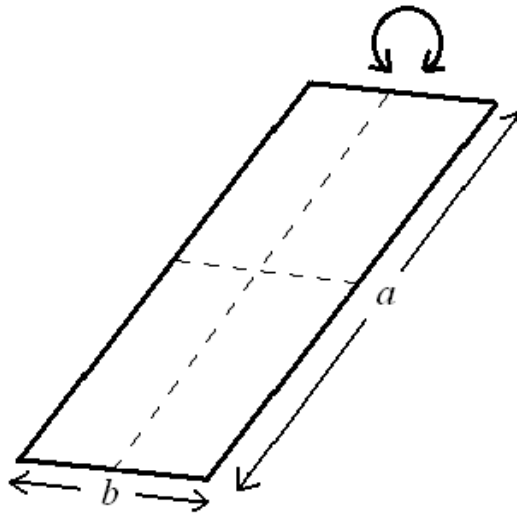


図2: 木の葉のモデル。

### §§2-2 空気抵抗

まずは空気抵抗を考える。木の葉の進行方向に対する空気抵抗を  $F$ 、空気の粘性係数を  $f$ 、木の葉の水平面に対する法線ベクトルの単位ベクトルを  $n$ 、木の葉の中点の速度を  $v$ 、木の葉の表面積を  $S$  とすると以下のように計算することができる。

$$F = -fSn(n \cdot v) \quad (2)$$

ただし、この  $F$  は進行方向に対する空気抵抗である。 $x$  軸方向および  $z$  軸方向の運動方程式を考える際、 $F$  を  $x$  成分と  $z$  成分に分解する必要がある。

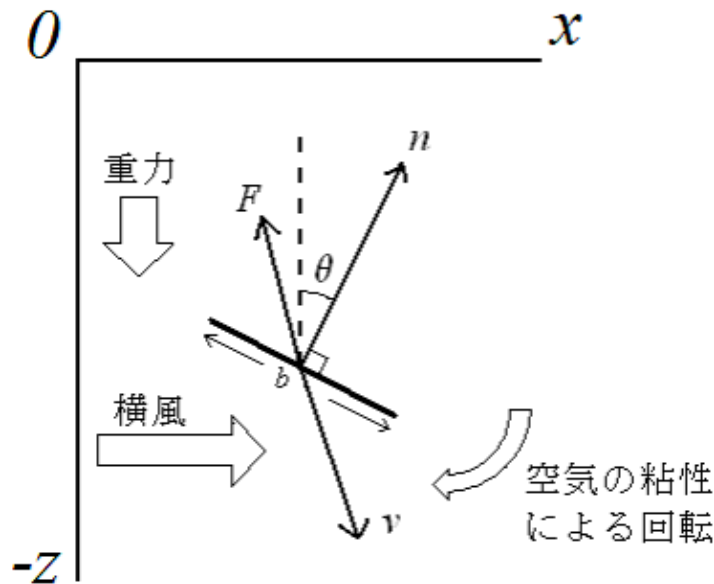


図 3: 木の葉にかかる様々な力。

### §§ 2-3 モーメント

木の葉は空気との摩擦によって回転運動をする。この回転運動によって木の葉にかかる空気抵抗は一定ではなく様々に変化する。この回転運動のモーメント  $L$  は以下の式で求めることができる。

$$L = c[\mathbf{n} \times \mathbf{v}] \cdot \text{sgn}(\mathbf{n} \cdot \mathbf{v}) \quad (3)$$

ただし、係数  $c$  は実験上で定義するものとし、§§ 2-4 にて詳しく論ずる。

### §§ 2-4 運動方程式

§§ 2-2 と §§ 2-3 で立てた式 (2)(3) を利用して運動方程式を組み立てる。

$x$  成分の運動方程式は以下のとおりである。ただし、 $\sigma$  は木の葉の表面密度である。質量  $m$  は  $m = 1$  とした。

$$\frac{d^2x}{dt^2} = -\frac{f}{\sigma} \left( \frac{dz}{dt} \cos \theta + \frac{dx}{dt} \sin \theta \right) \sin \theta \quad (4)$$

$z$  成分の運動方程式は、重力加速度を  $g$  とすると以下のとおりである。

$$\frac{d^2z}{dt^2} = -g - \frac{f}{\sigma} \left( \frac{dz}{dt} \cos \theta + \frac{dx}{dt} \sin \theta \right) \cos \theta \quad (5)$$

短辺  $b$  の中点を中心とした回転の運動方程式は以下のとおりである。

$$\frac{d^2\theta}{dt^2} = C \left\{ \left| \frac{dx}{dt} \right| \cos \theta \cdot \text{sgn}(\sin \theta) - \left| \frac{dz}{dt} \right| \sin \theta \cdot \text{sgn}(\cos \theta) \right\} \quad (6)$$

ここで  $C$  を  $C \equiv \frac{12c}{(mb^2)} \text{cm}^{-1} \text{sec}^{-1}$  と定義する。 $b$  が大きくなるほど  $C$  は小さくなる。 $C$  が小さくなると  $\frac{d^2\theta}{dt^2}$  は小さくなる。つまり、 $b$  が大きくなるほど回転しにくくなるという定義である。

また実験を行うにあたって、初期条件を設定する必要がある。原点から初速度、回転速度共にゼロで落下し始めると考える。初期条件は以下の通りとする。

$$x_{t=0} = 0 \quad (7)$$

$$z_{t=0} = 0 \quad (8)$$

$$\left(\frac{dx}{dt}\right)_{t=0} = 0 \quad (9)$$

$$\left(\frac{dz}{dt}\right)_{t=0} = 0 \quad (10)$$

$$\left(\frac{d\theta}{dt}\right)_{t=0} = 0 \quad (11)$$

初期の木の葉の傾き  $\theta_{t=0}$ 、回転モーメントの係数  $C$  はパラメータとして、様々に変化させてシミュレーションを行うものとする。

### § 3 数値実験方法

#### §§ 3-1 プログラムの作成

本研究では、プログラミングを用いて運動方程式の数値解を求め、落下軌道を予測した。使用したプログラミング言語はC++である。

プログラムでは初期値を設定し、ルンゲクッタ法を用いて計算を行った。ルンゲクッタ法については§§ 3-2にて説明する。ルンゲクッタ法で計算を行う際に、刻み幅  $\Delta t$  を設定する必要があるが、本研究では  $\Delta t = 0.005$  とした。これは  $\Delta t$  をこれよりも小さくしても、計算上大きな誤差は見られず、 $\Delta t = 0.005$  としても問題ないと判断したためである。出力方法は、Excel を利用してグラフを作成するために csv 形式でのファイル出力とした。結果を出力した変数は  $t$ 、 $x$ 、 $z$ 、 $\frac{d\theta}{dt}$  にあたるものである。

プログラム上、木の葉を水平に落とした場合や垂直に落とした場合は計算できないので別枠で計算を行った。水平に落とした場合  $x$  軸方向には動かず、 $z$  軸方向に落下するのみであるので、以下の式で計算することができる。

$$\frac{dz}{dt} = \frac{g\sigma}{f} \left\{ \exp\left(-\frac{f}{\sigma}t\right) - 1 \right\} \quad (12)$$

垂直に落とした場合も、横風等の外因的要素がない限り水平に落としたときと同様に  $x$  軸方向には動かず、また落下方向からの空気抵抗も受けないので自由落下と同様である。初期条件 (8)、(10) より以下の式となる。

$$\frac{dz}{dt} = gt \quad (13)$$

$$z = -\frac{1}{2}gt^2 \quad (14)$$

#### §§ 3-2 ルンゲクッタ法

ルンゲクッタ法とは、常微分方程式の近似解を求めるための解法の一つである。常微分方程式の近似解を求める解法としてはオイラー法などもあるが、本研究ではより精度の高い4次のルンゲクッタ法を採用した。4次のルンゲクッタ法を用いて常微分方程式の近似解を求める場合、以下の公式を解くことで段階的に求めることができる。

$x' = f(t, x)$ ,  $x(t_0) = x_0$  という問題を解く場合、

$$x_{i+1} = x_i + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (15)$$

添え字  $i$  はステップ数を表す。ここで、 $k_1$ 、 $k_2$ 、 $k_3$ 、 $k_4$  は以下の式で求める。

$$k_1 = f(t_i, x_i) \quad (16)$$

$$k_2 = f\left(t_i + \frac{\Delta t}{2}, x_i + \frac{\Delta t}{2}k_1\right) \quad (17)$$

$$k_3 = f\left(t_i + \frac{\Delta t}{2}, x_i + \frac{\Delta t}{2}k_2\right) \quad (18)$$

$$k_4 = f(t_i + \Delta t, x_i + \Delta tk_3) \quad (19)$$

ただし、本研究で解く常微分方程式は2階微分方程式であるため、次のように分解した。

$$\frac{dv}{dt} = f(t, x) \quad (20)$$

$$\frac{dx}{dt} = v_x \quad (21)$$

これをプログラム化させるために作成したアルゴリズムは付録 A に掲載している。



## § 4 実験結果と考察

### §§ 4-1 落下軌道

落下軌道のシミュレーションを行う際、 $\theta_{t=0}$ の値を $0.01\text{rad}$ (約 $0.6^\circ$ )、 $0.51\text{rad}$ (約 $29.2^\circ$ )、 $1.01\text{rad}$ (約 $57.9^\circ$ )、 $1.51\text{rad}$ (約 $86.5^\circ$ )の4つとした。また、 $C$ の値を $0.5\text{cm}^{-1}\text{sec}^{-1}$ 、 $1.0\text{cm}^{-1}\text{sec}^{-1}$ 、 $2.0\text{cm}^{-1}\text{sec}^{-1}$ の3つとし、 $\frac{f}{\sigma} = 40\text{sec}^{-1}$ として計12通りの落下軌道のシミュレーションを行った。以下に出力結果をグラフ化したものを示す。(図4、5、6)

図4、5、6について、同図内の各グラフを比較しやすくするために、各グラフを $x$ 軸に関してずらして表示しているが、シミュレーションする際の初期位置は全て $x_0 = 0$ 、 $z_0 = 0$ である。また、各図の $z$ 軸の補助目盛の間隔は全て同じである。

図4、5、6から、どの場合でも1回転もしくは2回転したところから揺れずに、ほぼ真っすぐ落下していくことがわかる。また、 $C$ を大きくすると、同じ時間落下させても落下距離( $|-z|$ の最大値)が明らかに小さくなることがわかる。これらの理由については、§§ 4-2にて、角速度の変化と併せて詳しく論ずる。

また、図4、5、6から、落下距離は $\theta_0$ に比例しないことが見て取れる。そこで、 $\theta_{t=0}$ を $0.01\text{rad}$ から $1.57\text{rad}$ まで $0.01\text{rad}$ ずつ回転させ、落下距離が最大になるときの $\theta_{t=0}$ を求めるプログラムを作成し算出した。その結果をグラフにしたものが図7である。このグラフより、 $C$ の値に関係なく $\theta_{t=0}$ に応じて落下距離は徐々に大きくなり、 $\theta_{t=0}$ がある一定の値を過ぎると徐々に落下距離が小さくなることがわかる。落下距離が最大値をとるときの $\theta_{t=0}$ の値は、全ての $C$ の値において $\theta_{t=0} = 0.78\text{rad}$ (約 $44.7^\circ$ )であった。つまり、この角度で木の葉を落とすと最も落下距離が大きくなると考えられる。

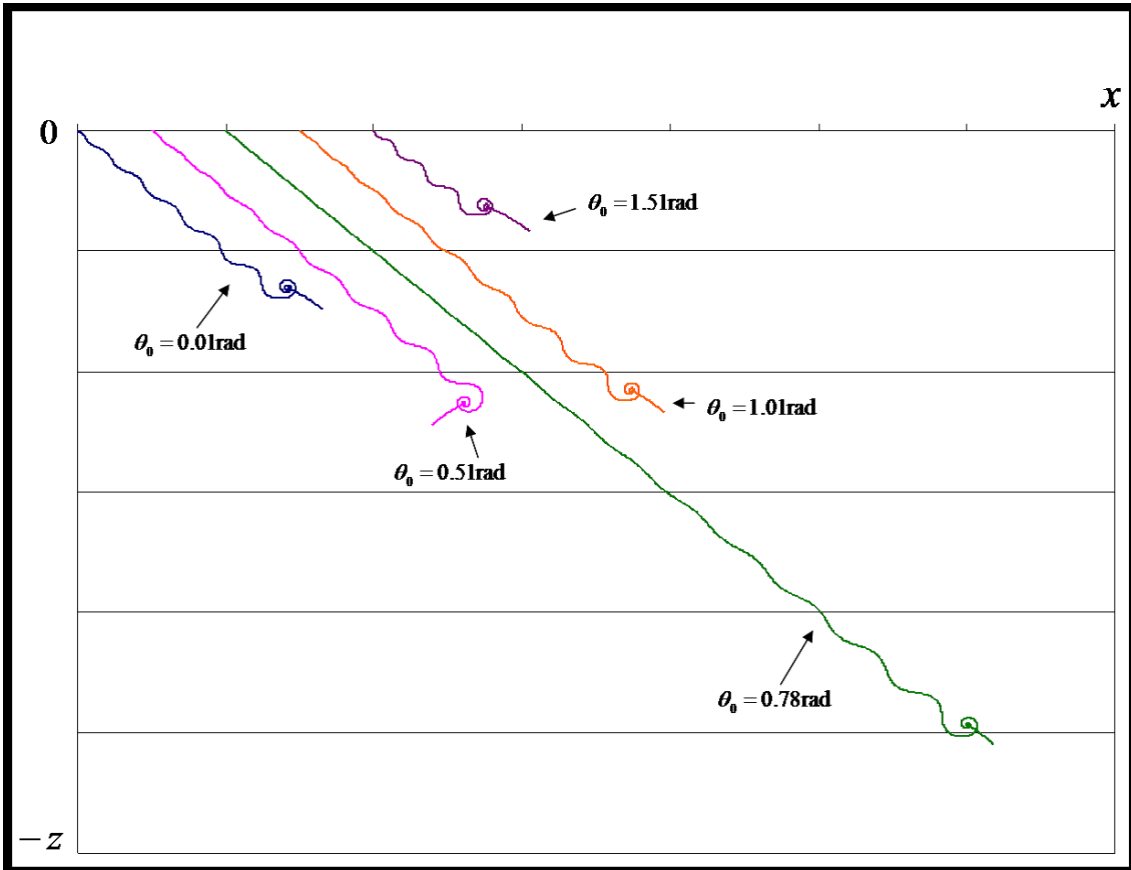


図 4:  $C = 0.5$  のときの落下軌道。落下距離は  $\theta_t = 0$  に比例しないことがわかる。

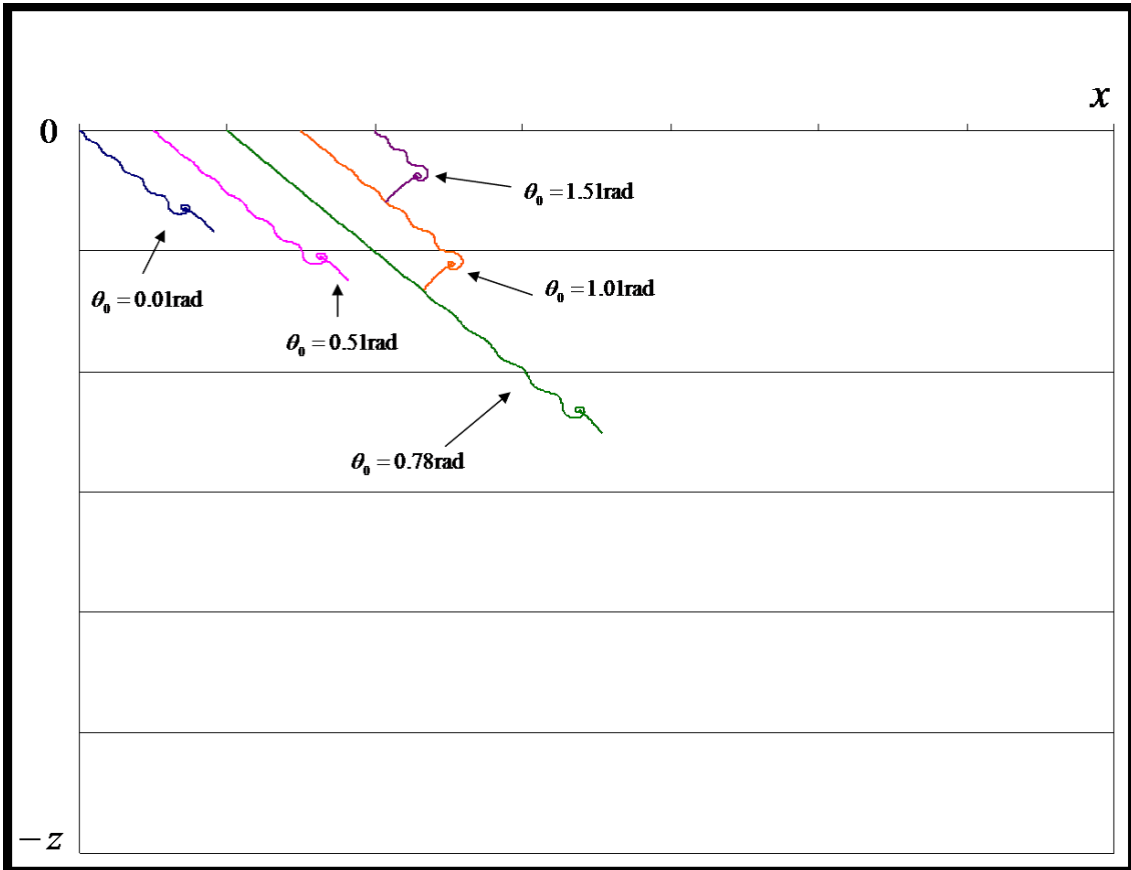


図 5:  $C = 1.0$  のときの落下軌道。 $C = 0.5$  の場合よりも落下距離が小さいことがわかる。

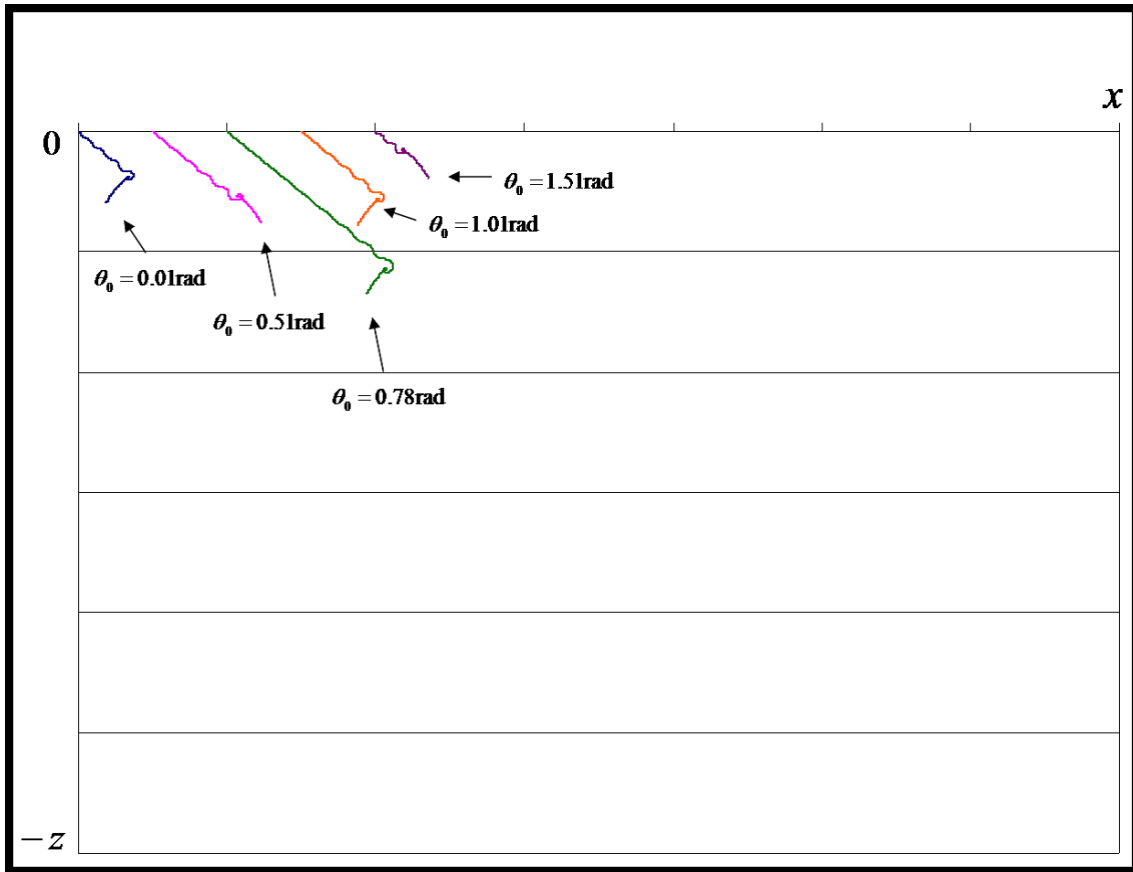


図 6:  $C = 2.0$  のときの落下軌道。 $C = 0.5$ 、 $C = 1.0$  のどちらの場合よりもさらに落下距離が小さいことがわかる。

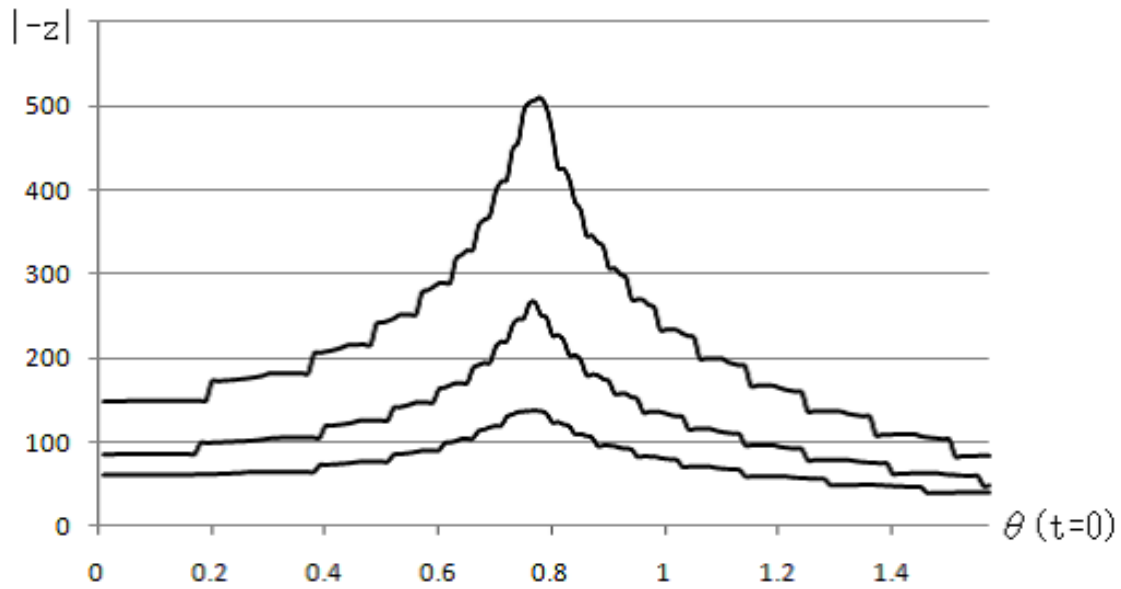


図 7:  $\theta_{t=0}$  を変化させたときの落下距離の変化。どのグラフも  $\theta_{t=0}$  がほぼ同じ値のときに  $| -z |$  が最大値をとっている。

## §§ 4-2 角速度の変化

角速度の変化をみるため、 $t$  と  $t$  に対する  $\frac{d\theta}{dt}$  を出力するプログラムを作成した。以下に出力結果をグラフ化したものを示す。

図 8 をみると、 $C$  がどの値でもグラフには  $\frac{d\theta}{dt}$  が正と負を行き来する波形 (以下 波形) の部分と、正または負の値を取り続ける波形 (以下 波形) の部分があるのが確認できる。波形の部分は木の葉が揺れる動きを表している。また、波形の部分は一方向に回転する動きを表している。どのグラフも、一方向に回転し始めるとそれ以降は揺れる動きに戻らないことがわかる。ここで、波形の部分と波形の部分の変わり目に着目すると、 $C$  の値が大きくなるほど波形から波形に早く変わることがわかる。この変わり目は図 4、5、6 から、木の葉が一回転、または二回転する動きを見せ、ほぼ直線に落下し始める部分であると考えられる。つまり、揺れる運動から回転する運動に早く変わるほど、落下距離 ( $|-z|$ ) が小さくなると考えられる。そこで、 $t$ 、 $\frac{d\theta}{dt}$ 、 $\frac{dz}{dt}$  を出力するプログラムを作成した。出力結果をグラフ化したものが図 9 である。 $\frac{dz}{dt}$  のグラフで、大きく負の値をとっている部分が木の葉が一回転したときである。ここから  $\frac{d\theta}{dt}$  はほぼ一定の小さな値で安定していると同時に、 $\frac{d\theta}{dt}$  のグラフをみると木の葉が回転運動に変わっていることがわかる。これらのことから、同時間落下させた場合  $C$  の値が大きいくほど、落下速度の遅い回転運動をする時間の比率が大きいくので、落下距離が小さくなると考えられる。

エネルギー的に考えると次のように解釈できる。回転がなければ落下によって得た位置エネルギーの差は、全て落下の速度の増加になる。しかし回転の自由度があると、回転エネルギーの一部が使われる為に、落下の速度増加がその分少ない。したがって、回転するほど落下距離は短くなる。

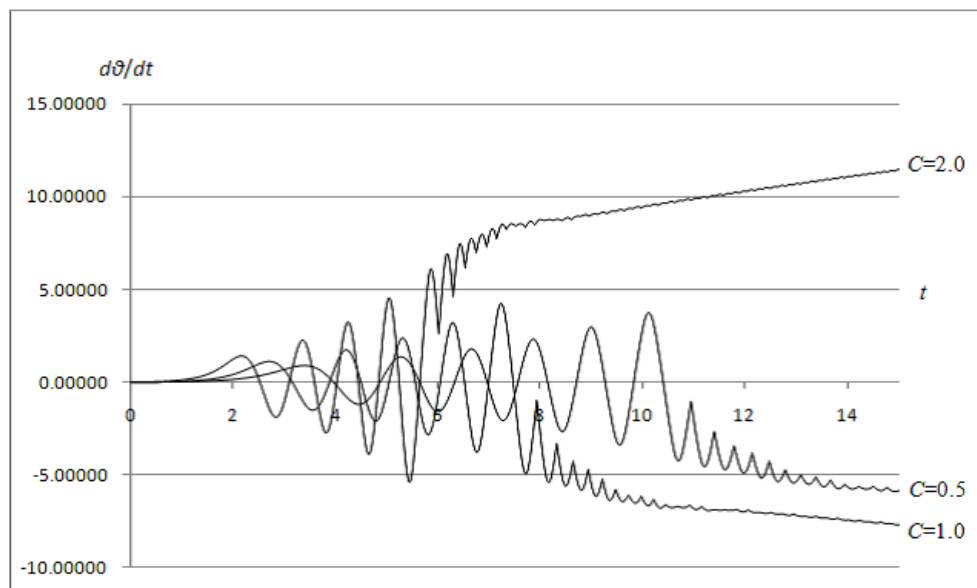


図 8:  $\theta_{t=0} = 0.01$  のときの角速度の変化。どのグラフも徐々に振幅が大きくなるが、最終的に正負どちらかで安定する。

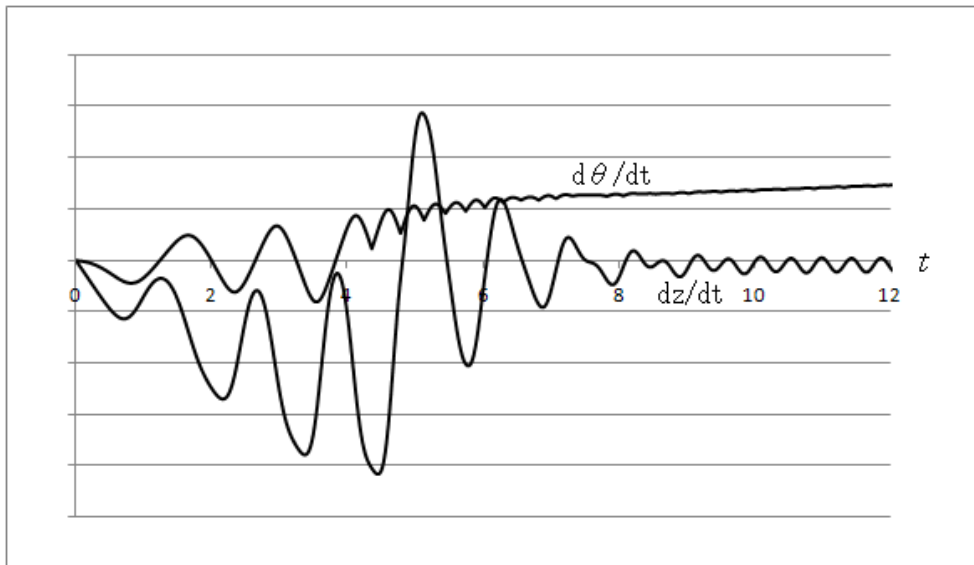


図 9:  $C = 1.0$ 、 $\theta_{t=0} = 1.51$  のときの角速度と落下速度 ( $\frac{dz}{dt}$ ) の関係。

図 8 を見ると、波形が正の値を取り続ける場合と負の値を取り続ける場合があることについて考察する。 $(C = 2.0, \theta_{t=0} = 0.01)$  の場合のように波形が正の値を取り続ける場合、木の葉の回転方向は時計回りとなる。逆に  $(C = 1.0, \theta_{t=0} = 0.01)$  や  $(C = 0.5, \theta_{t=0} = 0.01)$  の場合のように負の値を取り続ける場合、木の葉の回転方向は半時計回りとなる。ここで、図 4、図 5、図 6 の  $\theta_{t=0} = 0.01$  の時の落下軌道を見ると、回転運動中に  $x$  軸に対して木の葉が正負どちらの方向に流れるかで違いが確認できる。具体的には  $(C = 2.0, \theta_{t=0} = 0.01)$  の場合にのみ、負の方向に木の葉が流れている。このことから、波形部分が正の値を取り続け、木の葉が時計回りに回転運動をすると木の葉は  $x$  軸に対して負の方向に流れ、逆に波形部分が負の値を取り続け、木の葉が半時計回りに回転運動をすると木の葉は  $x$  軸に対して正の方向に流れるものと考えられる。

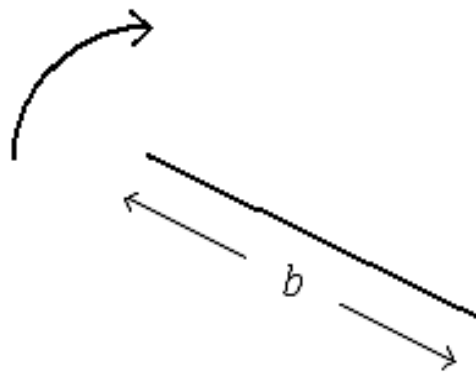


図 10: 波形部分が正の値を取り続けるときの木の葉の回転方向は時計回りである。

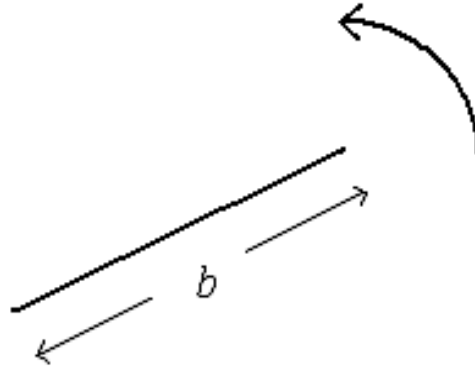


図 11: 波形部分が負の値を取り続けるときの木の葉の回転方向は反時計回りである。

さらに  $\theta_{t=0}$  値を変化させ、 $\frac{d\theta}{dt}$  が正負どちらの値で安定するのかを求め、最終的に木の葉が  $x$  軸に対して正の方向 (左) に流れるのか負の方向 (右) に流れるのかをグラフ化したものが図 12 である。 $\theta_{t=0}$  は  $\theta_{t=0} = 0.0001\text{rad}$  から  $\theta_{t=0} = 1.51\text{rad}$  まで、 $0.0001\text{rad}$  ずつ増加させ、 $C = 2.0$  としてシミュレーションを行った。図 12 を見ると、§ 4-1 で最も  $z$  の変位が大きいとわかった  $\theta_{t=0} = 0.78\text{rad}$  の前後において、正と負の反転が小刻みになっている部分が 2 箇所あるのがわかる。

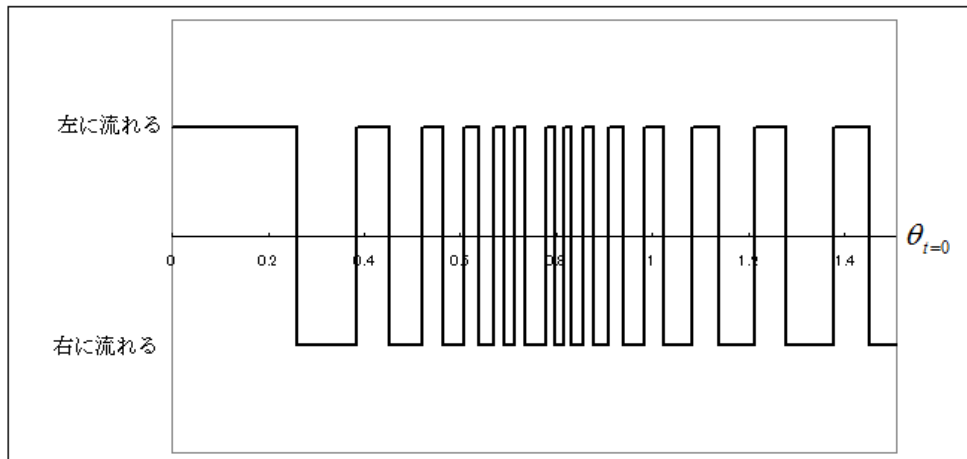


図 12:  $\theta_{t=0}$  と回転方向 (木の葉が流れる方向) の関係。  $\theta_{t=0} = 0.78\text{rad}$  の前後に反転が小刻みになる部分が 2 箇所ある。

### §§ 4-3 横風を加えた場合

次に、風が木の葉の運動にどのような効果を及ぼすかについて考察する。まずは右向きを正とした横風の強さを風力  $k$  として、式 (4)、(5) に組み込む。

ここで、木の葉の角度  $\theta$  によって木の葉が受ける風力が変化することに注意しなければならない。 $\theta$  を、 $0 \leq \theta < \pi/2$ 、 $\pi/2 \leq \theta < \pi$ 、 $\pi \leq \theta < 3\pi/2$ 、 $3\pi/2 \leq \theta < 2\pi$  の4つの場合に分け、 $x$  成分と  $z$  成分の運動方程式を変更する。 $x$  成分は  $\theta$  と  $k$  を利用した、以下の式である。

$$\frac{d^2x}{dt^2} = -\frac{f}{\sigma} \left( \frac{dz}{dt} \cos \theta + \frac{dx}{dt} \sin \theta \right) \sin \theta + k |\sin \theta| \quad (22)$$

$z$  成分は、 $\theta$  の値によって符号を変えなければならないため注意が必要である。理由は、木の葉にかかる風の力を  $z$  成分のみで考えた場合、図 13 のように  $\theta$  の値によって力の向きが逆転するからである。 $0 \leq \theta < \pi/2$ 、または  $\pi \leq \theta < 3\pi/2$  のときは以下の式である。

$$\frac{d^2z}{dt^2} = -g - \frac{f}{\sigma} \left( \frac{dz}{dt} \cos \theta + \frac{dx}{dt} \sin \theta \right) \cos \theta + k |\cos \theta| \quad (23)$$

また、 $\pi/2 \leq \theta < \pi$ 、または  $3\pi/2 \leq \theta < 2\pi$  のときは以下の式である。

$$\frac{d^2z}{dt^2} = -g - \frac{f}{\sigma} \left( \frac{dz}{dt} \cos \theta + \frac{dx}{dt} \sin \theta \right) \cos \theta - k |\cos \theta| \quad (24)$$

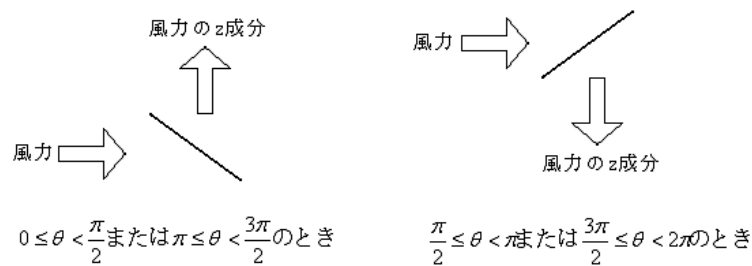


図 13:  $\theta$  による力の向きの変化

式 (4)、(5) を式 (22)、(23)、(24) に変更しシミュレーションを行った。風力は  $0\text{m/s}^2$ 、 $10\text{m/s}^2$ 、 $50\text{m/s}^2$ 、 $100\text{m/s}^2$ 、 $-10\text{m/s}^2$ 、 $-50\text{m/s}^2$ 、 $-100\text{m/s}^2$  の7つの場合を考え、 $C = 1.0$ 、 $\theta_{t=0} = 1.01$  とした場合の結果のグラフが図 14、15 である。



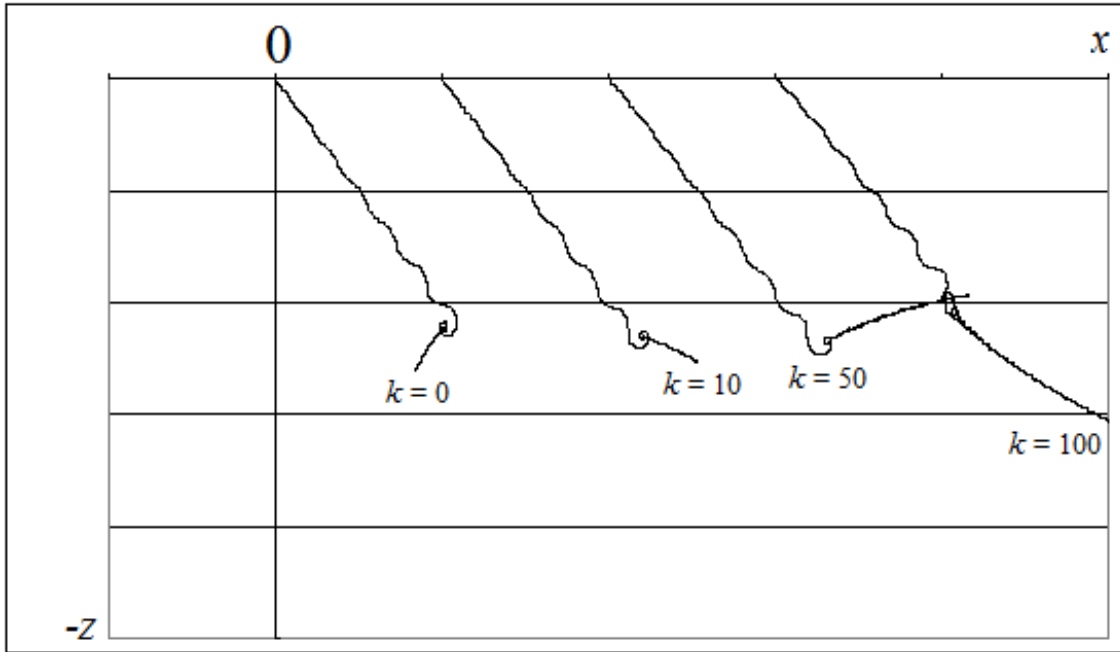


図 14:  $C = 1.0$ 、 $\theta_{t=0} = 1.01$ 、風力が正のときの落下軌道の変化。風力が大きくなるほど、右に大きく流されているのがわかる。

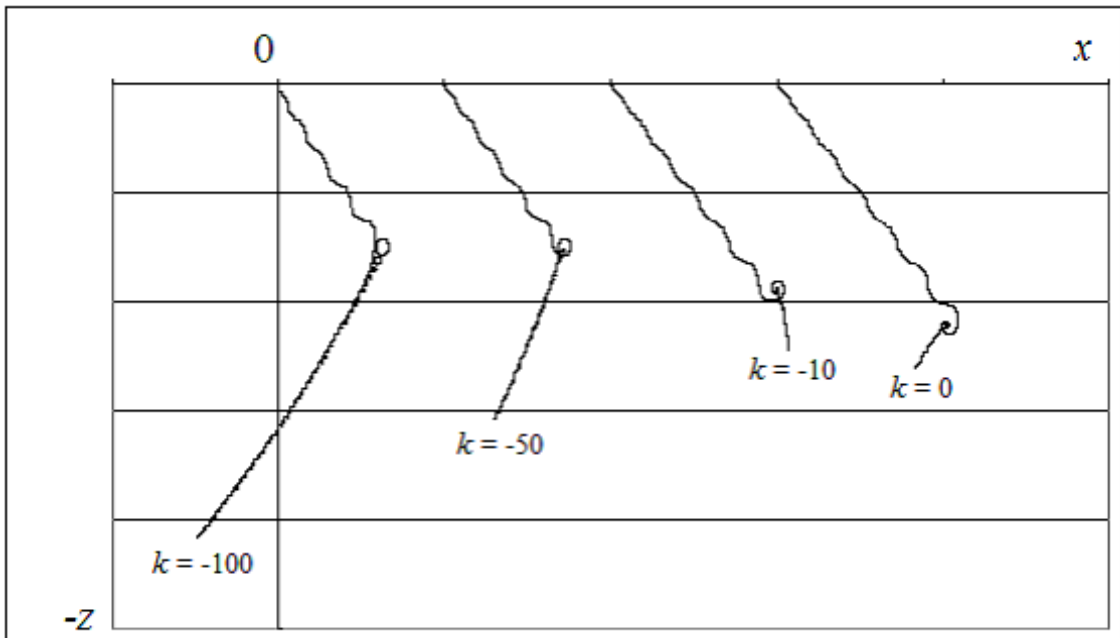


図 15:  $C = 1.0$ 、 $\theta_{t=0} = 1.01$ 、風力が負のときの落下軌道の変化。風力が大きくなるほど、左に大きく流されているのがわかる。

ここで風力が負の場合の落下軌道について考察する。風力が負の場合、木の葉は  $x$  軸に対して負の方向、つまり左に流れると思われる。しかし、シミュレーション結果のグラフを見てみると、風力の大きさに関わらず  $t = 0$  から木の葉が回転運動に変わるまでの間は右に流れている。これは、木の葉が  $t = 0$  から回転運動に変わるまでの間は、木の葉の傾き  $\theta$  が完全に  $90^\circ$  にならないので風力を完全に受け切らず、 $\frac{d^2x}{dt^2} > |k|$  となるので右に流れるものと考えられる。 $t = 0$  から僅かな時間の間は、 $\frac{d^2x}{dt^2}$  が小さいので左に流れているが、変位が小さいので図 15 のグラフではわかりにくい。以下の表 1 は、 $C = 0.5$ 、 $\theta_{t=0} = 1.01$ 、 $k = -100.0$  でシミュレーションしたときの  $t = 0$  から  $t = 1.000$  までの  $x$  の値である。また、この表より作成したグラフが図 16 である。これを見るとごく僅かに左に流れていることがわかる。

また、風力が正の場合に  $z$  軸に対して正の方向、つまり上方向に木の葉が浮き上がることが確認できた。これに関しても、動きが非常に小さく図 14 のグラフでは確認できないため、表にしたものが以下の表 2 である。また、この表より作成したグラフが図 17 である。ごく僅かな時間だけ  $z$  軸に対して正方向に変位していることがわかる。

表 1:  $C = 0.5$ 、 $\theta_{t=0} = 1.01$ 、 $k = -100.0$  でシミュレーションしたときの  $t = 0$  から  $t = 1.000$  までの  $x$  の値。

$t$	$x$
0.000	-0.000988
0.100	-0.154692
0.200	-0.308020
0.300	-0.416856
0.400	-0.477716
0.500	-0.485551
0.600	-0.432995
0.700	-0.311738
0.800	-0.114925
0.900	0.160491
1.000	0.512154

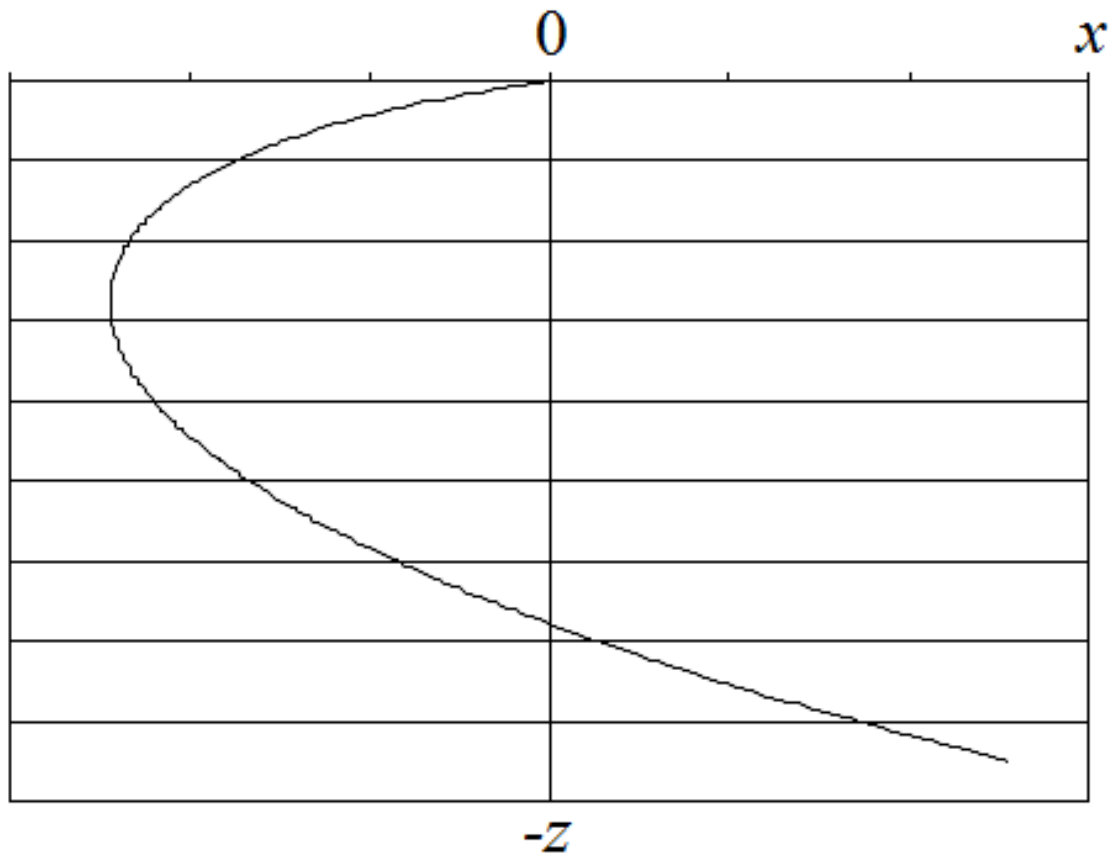


図 16:  $k = -100$  のときの  $t = 0$  周辺を拡大したグラフ。左に変位した後に右に流れているのがわかる。

表 2:  $C = 0.5$ 、 $\theta_{t=0} = 1.01$ 、 $k = 100.0$  でシミュレーションしたときの  $t = 0$  から  $t = 1.000$  までの  $z$  の値。

$t$	$z$
0.000	0.000000
0.100	0.062551
0.200	0.078998
0.300	0.024532
0.400	-0.103194
0.500	-0.306615
0.600	-0.588095
0.700	-0.949367
0.800	-1.390977
0.900	-1.911760
1.000	-2.508295



図 17:  $k = 100$  のときの  $t = 0$  周辺を拡大したグラフ。僅かに浮き上がった後に落下し始めていることがわかる。

## §5 アニメーション化

### §§5-1 アニメーション化とその結果

ここまでは木の葉の落下運動のシミュレーション結果をグラフ化してきたが、これでは毎時どのような動きをしているのかわかりづらい。そこで JavaApplet を利用してアニメーション化することで、動的にシミュレーション結果を見ることができるようプログラミングを行った。以下にその作成したアプレットの画面をキャプチャしたものを示す。(図 18、19、20)

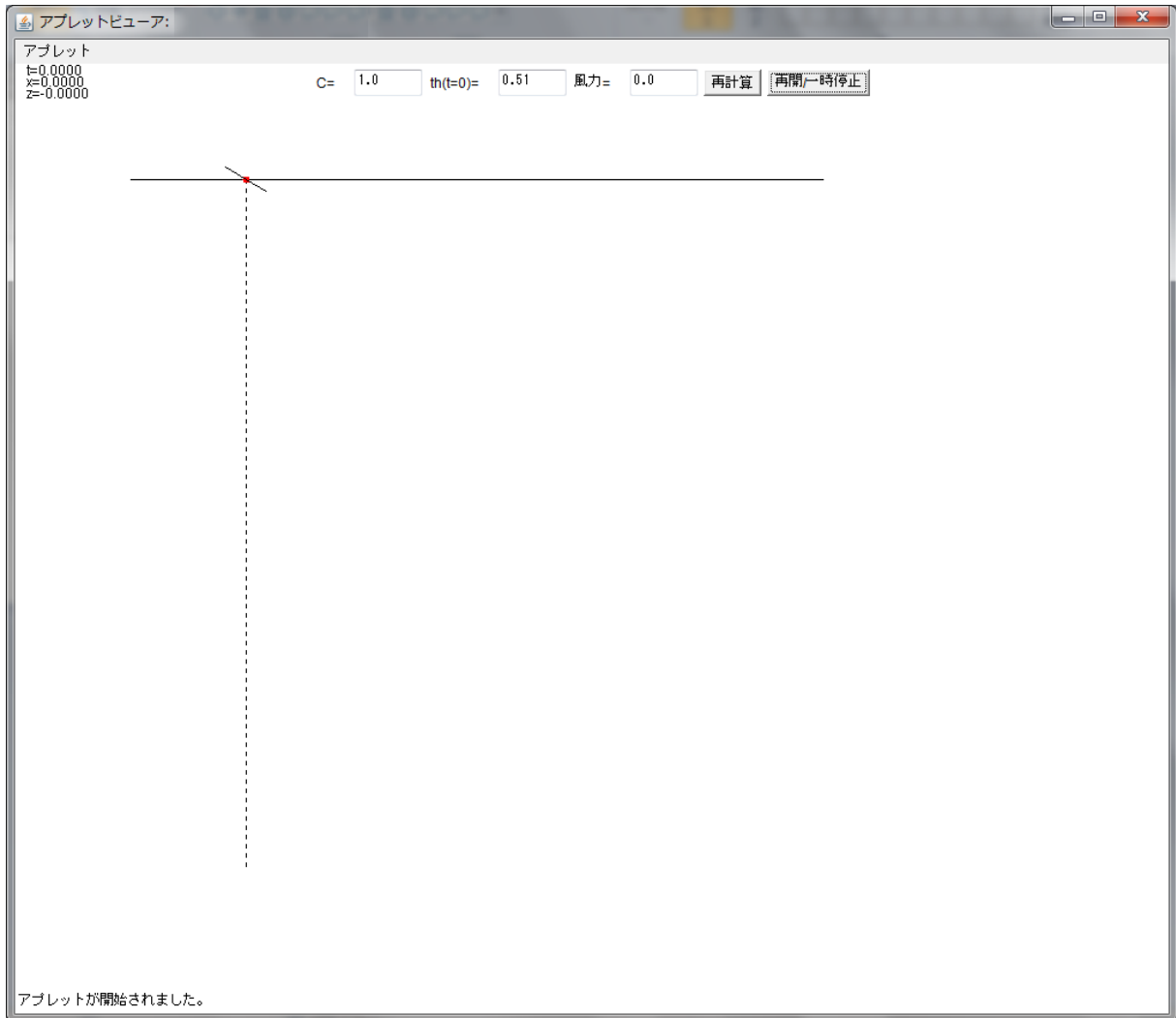


図 18: アプレット読み込み時のキャプチャ画像。最初に読み込まれた時の初期値は  $C = 1.0$ 、 $\theta_{t=0} = 1.01$ 、風力  $k = 0.0$  に設定されている。

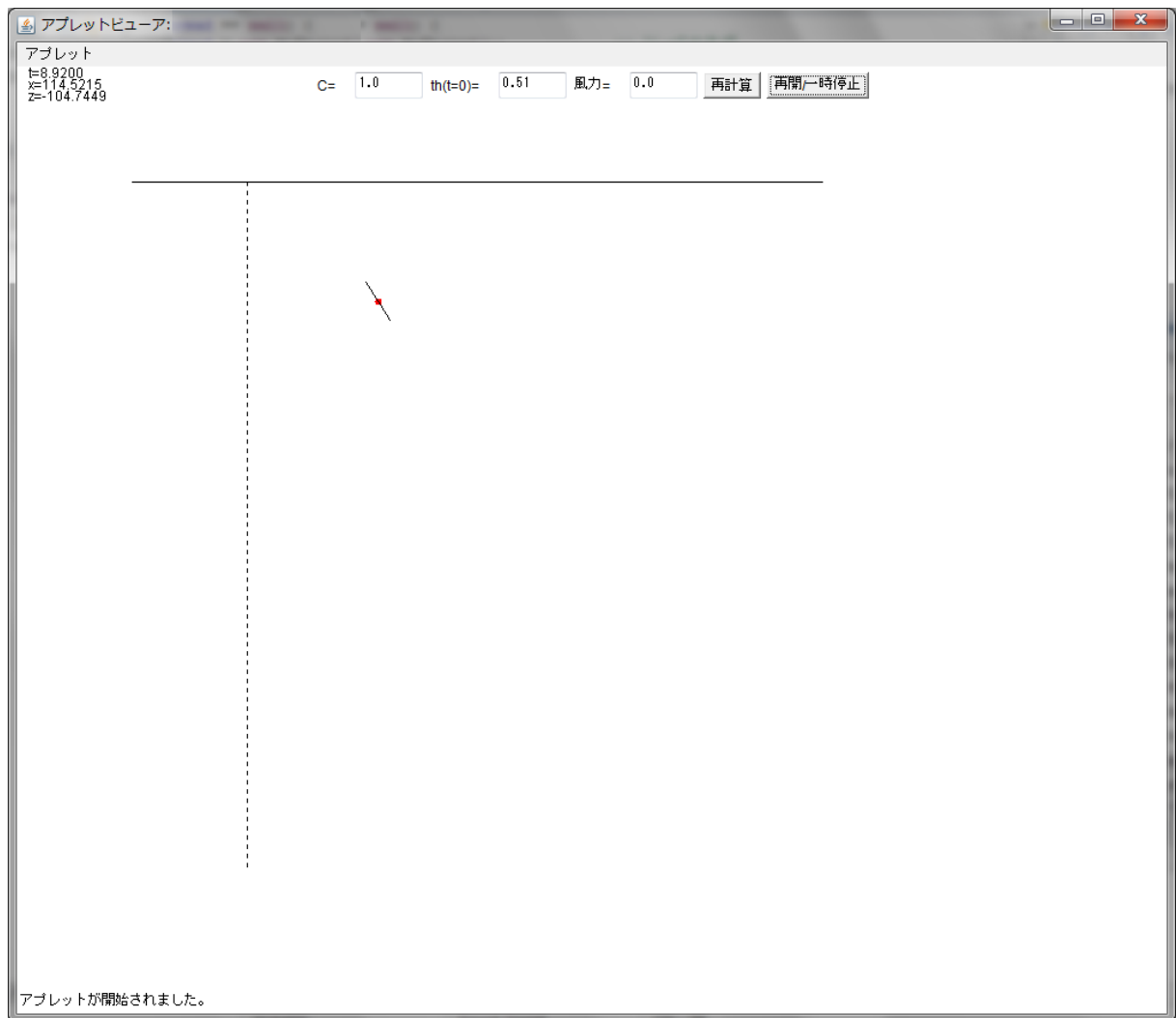


図 19: シミュレーション時のアプレットのキャプチャ画像。木の葉の落下の起動と回転の様子を同時に確認することができる。

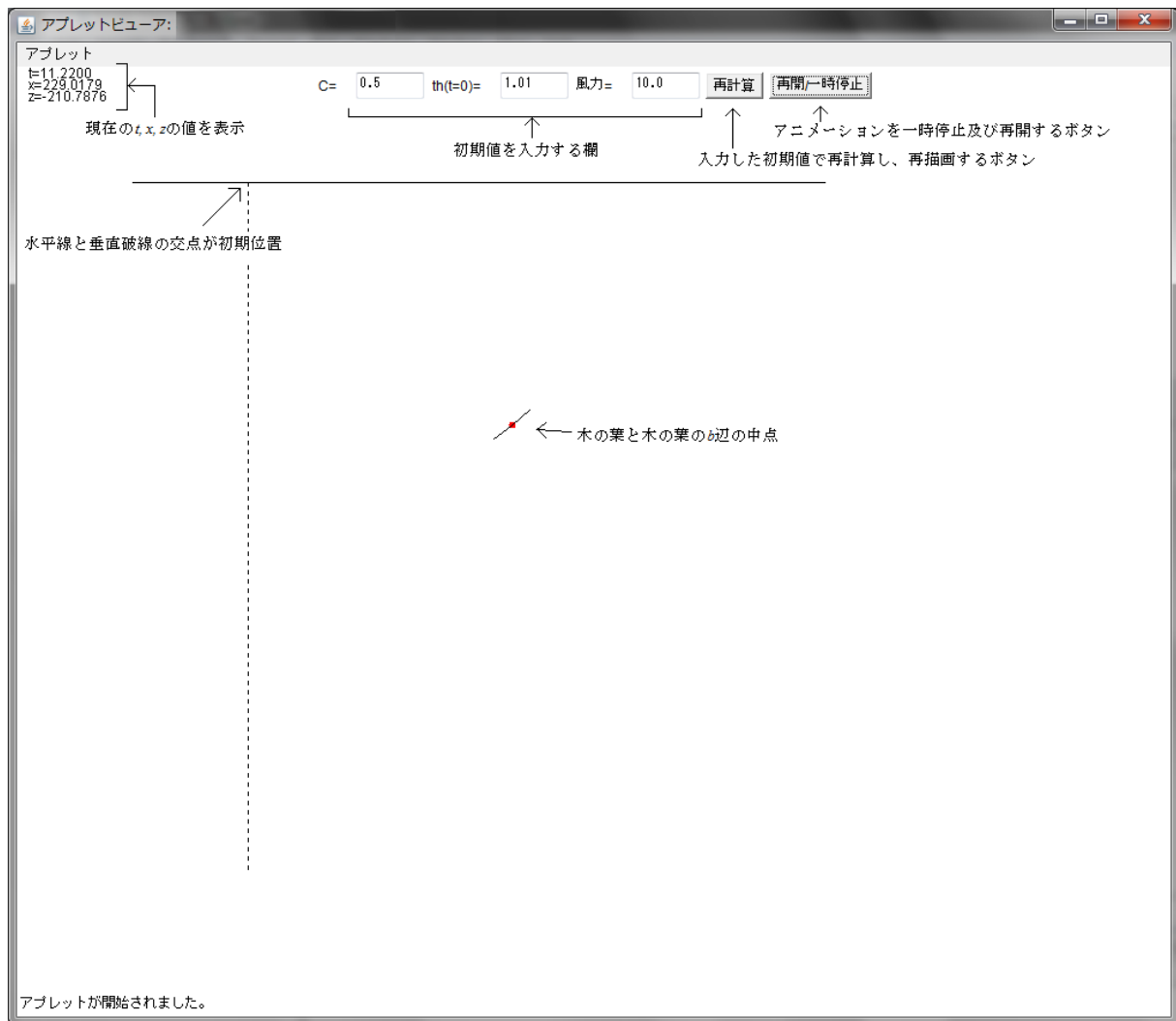


図 20: アプレット内の各 GUI の説明。初期値の変更をしてからの再シミュレーションや一時停止、再開が可能である。

## §5-2 アニメーション化における課題点と改善点

アニメーション化する際に注意したことは、計算結果の  $x$  と  $z$  の値を引数として図を描くまでの流れである。ルンゲクッタ法で1ステップ計算する毎に計算結果を引数として入力する方法でアニメーションを作成すると、アプレットを実行するコンピュータによっては1ステップの計算に時間がかかり、アニメーションにした際に滑らかさが失われる問題が生じた。そこで、 $x$ 、 $z$  それぞれの計算結果を格納する配列を作成し、そこに全ステップの計算を終えたのちにアニメーション化する方法に変更したところ、アニメーション化した際の滑らかさが向上した。(図 21)

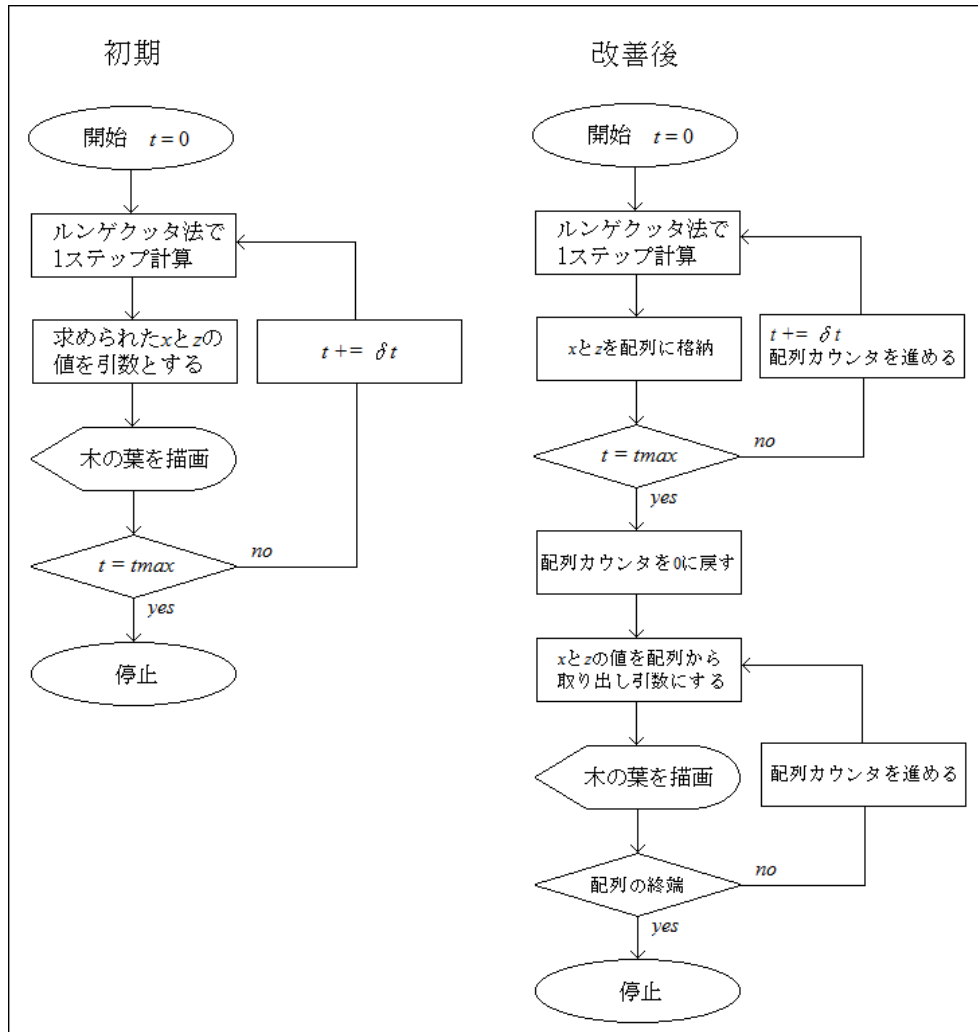


図 21: 改善前と改善後のフローチャートの比較。



## 参考文献

- [1] Takao Yamamoto(1990) "Falling Motion of a Leaf" Sci.Rep.Hirosaki Univ.37,6-9
- [2] 真貝 寿明 (2010) 『徹底攻略 常微分方程式』 共立出版株式会社
- [3] 伊理 正夫・藤野 和建 (1985) 『数値計算の常識』 共立出版株式会社
- [4] 小泉 修 (2006) 『理系のための Java 入門』 株式会社工学社
- [5] 橋本 佳幸 (2010) 『iPad プログラミングの作法』 株式会社秀和システム

### A ルンゲクッタ法のアルゴリズム

問題とする常微分方程式は  $\frac{d^2x}{dt^2} = x$  とする。

```
double x=0;
double vx=0;

double t;
double dt=0.005;    //t の刻み幅
double tmax=50.0;  //t の最大値

double k1[2], k2[2], k3[2], k4[2];

for(t = 0; t < tmax; t += dt){

    k1[0] = dt * xfunc1(t, x, vx);
    k1[1] = dt * xfunc2(t, x, vx);
    k2[0] = dt * xfunc1(t + dt / 2.0, x + k1[0] / 2.0, vx + k1[1] / 2.0);
    k2[1] = dt * xfunc2(t + dt / 2.0, x + k1[0] / 2.0, vx + k1[1] / 2.0);
    k3[0] = dt * xfunc1(t + dt / 2.0, x + k2[0] / 2.0, vx + k2[1] / 2.0);
    k3[1] = dt * xfunc2(t + dt / 2.0, x + k2[0] / 2.0, vx + k2[1] / 2.0);
    k4[0] = dt * xfunc1(t + dt, x + k3[0], vx + k3[1]);
    k4[1] = dt * xfunc2(t + dt, x + k3[0], vx + k3[1]);

    x = x + (k1[0] + 2.0 * k2[0] + 2.0 * k3[0] + k4[0]) / 6.0;
    vx = vx + (k1[1] + 2.0 * k2[1] + 2.0 * k3[1] + k4[1]) / 6.0;

}

//main 関数外に以下の関数を加える。
```

```
double xfunc1(double t, double x, double vx){  
    return vx;  
}  
double xfunc2(double t, double x, double vx){  
    return x;  
}
```