

単振り子の話

桂田祐史

2007年10月8日, 2012/3/2, 2015/11/12 訂正

<http://nalab.mind.meiji.ac.jp/~mk/labo/text/furiko/>

1 はじめに

紐の長さが l である単振り子 (単振り子, simple pendulum) の運動方程式は、 m を重りの質量、 g を重力加速度として

$$m\ell\theta''(t) = mg \sin \theta(t)$$

となる。これから

$$(1) \quad \theta''(t) + \omega^2 \sin \theta(t) = 0, \quad \omega := \sqrt{\frac{g}{\ell}}$$

という微分方程式が得られる (ここに m は出て来ないので、単振り子の運動は質量には依らない、ということはすぐ分かる)。

微小振動 ($|\theta| \sim 0$) の場合は、 $\sin \theta$ は θ に近いので、単振り子の運動は

$$\theta''(t) + \omega^2 \theta(t) = 0$$

に従う単振動で良く近似できる、とされる (高校物理の相場)。単振動は (大学の理系の学科では) 「常識的」で¹、一般解は

$$\theta(t) = C_1 \cos \omega t + C_2 \sin \omega t \quad (C_1, C_2 \text{ は任意定数})$$

であり、これは周期

$$T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{\ell}{g}}$$

を持つ周期関数である、と明解に解ける。この周期が g と ℓ だけで決まることから、「振り子の等時性が成り立つ (周期は振幅によらない)」ということにもなる。

しかし、単振動で近似するのはあくまでも近似である。どうして近似を用いた説明が多いのかというと、近似しないで解くのが難しいからである (後で見るように、解や周期を表すのに、楕円関数、楕円積分というものが必要になる²)。この文書の目的は、本来の振り子の運動 (微分方程式 (1) の解) がどうなるか (近似を用いずに)、きちんと考えてみよう、ということである。

¹単振動については、『単振動の数学的解析』(<http://nalab.mind.meiji.ac.jp/~mk/keisanki2/simple-harmonic-motion/>) という解説を書いたことがある。

²楕円積分、楕円関数については、内輪向けだが、『 π ノート』(<http://nalab.mind.meiji.ac.jp/~mk/labo/text/pi.pdf>) というのを書いた。

単振り子は、単振動ではないとすると、本当に周期運動をするか？と心配となるが、次節でも証明するエネルギー保存則

$$\frac{1}{2}\theta'(t)^2 - \omega^2 \cos \theta(t) = \text{定数}$$

より大丈夫であることが分かる (戻ってきた時、前と同じ速さであり、減衰したりはしない³)。

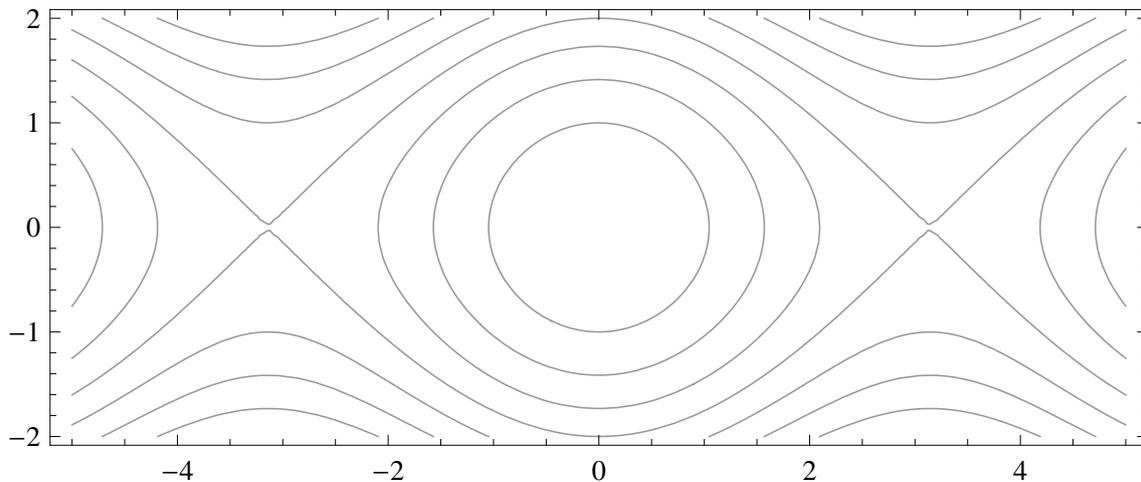


図 1: 横軸 θ , 縦軸 θ' . 閉曲線は、行ったり来たり。勢いが良いとぐるぐる (振り子じゃない!).

ガリレオが発見したという『振り子の等時性』は、実は厳密には成立しない性質であるが (つまり周期は本当は振幅による)、どの程度成り立っているものなのかも、明らかにしよう (目で見て分かるようにする — 後の図 5)。

2 微分方程式の解析

前節で述べたことの繰り返しになるが、長さ l の単振り子の微分方程式は

$$(2) \quad \theta''(t) = -\omega^2 \sin \theta(t).$$

ただし

$$\omega := \sqrt{\frac{g}{l}}.$$

2.1 エネルギー保存則

(2) に $\theta'(t)$ をかけると、

$$\theta''(t)\theta'(t) + \omega^2\theta'(t) \sin \theta(t) = 0.$$

すなわち

$$\frac{d}{dt} \left[\frac{1}{2}\theta'(t)^2 - \omega^2 \cos \theta(t) \right] = 0.$$

ゆえに

$$(3) \quad \frac{1}{2}\theta'(t)^2 - \omega^2 \cos \theta(t) = E \quad (E \text{ はある定数}).$$

これは物理的にはエネルギー保存則に相当する ($m\ell E$ が全エネルギーを表す)。

³厳密に証明するには、途中で止まらないことを言う必要があるけれど、それは明らかでしょう。注意が必要なのは、 90° 以下の触れの範囲の ($0^\circ < \theta_0 \leq 90^\circ$ として、 $-\theta_0 \leq \theta \leq \theta_0$ の範囲を行ったり来たりする) 「普通の」振り子以外に、 90° より大きな触れの範囲の振り子 ($90^\circ < \theta_0 \leq 180^\circ$ — あまり現実的ではない?) や、鉄棒の大車輪のようなぐるぐる回る振り子 (これはリアリティがあるけれど、「振り子」という名前に相応しくない?) などの現象が方程式に含まれていることである。

予備的考察

後のために、 E を振幅 (θ の最大値) θ_0 で書き直しておこう⁴。 $\theta = \theta_0$ のとき $\theta' = 0$ であるから、

$$\frac{1}{2} \cdot 0^2 - \omega^2 \cos \theta_0 = E.$$

これから

$$E = -\omega^2 \cos \theta_0.$$

ゆえに (3) は次のように書き換えられる。

$$(4) \quad \frac{1}{2} \theta'(t)^2 - \omega^2 (\cos \theta(t) - \cos \theta_0) = 0.$$

ある時刻に $\theta = 0$ となったとすると、 $(\theta')^2 = 2\omega^2(1 - \cos \theta_0) = 4\omega^2 \sin^2 \frac{\theta_0}{2}$ であるから、

$$\theta' = \pm 2\omega \sin \frac{\theta_0}{2}$$

である。つまりつりあいの位置にあるときの速度が振幅で表せたわけである。

2.2 厳密解

結果の式が簡単になるように、適当な初期条件を課した初期値問題の解を求めよう。

θ_0 を $0 < \theta_0 < \pi$ なる定数として、

$$(5) \quad k := \sin \frac{\theta_0}{2}$$

とおき、初期条件を

$$(6) \quad \theta(0) = 0, \quad \theta'(0) = 2k\omega$$

とする。前項の議論から振幅が θ_0 であることが分かる。

この初期値問題の解は

$$\theta(t) = 2 \sin^{-1} (k \operatorname{sn}(\omega t, k))$$

であることを示そう。

まずエネルギー保存則 (4) から

$$\frac{d\theta}{dt} = \sqrt{2\omega} \sqrt{\cos \theta - \cos \theta_0}.$$

(\pm をつけるべきかもしれないが、 $\theta'(0) > 0$ であるから、 t が小さいうちはこの微分方程式に従うはずである。)

これは変数分離形の微分方程式である。天下りであるが、

$$(7) \quad kz = \sin \frac{\theta}{2}$$

⁴ぐるぐる回る大車輪似の「振り子らしくない」振り子では、 θ の最大値が存在しない。そういう場合は考えない、ということにしておく。この文書の議論は、色々な本に載っているものを適当にブレンドしたものではあるけれど、なかなかすっきりしないですねえ。

と変数変換する。

$$\begin{aligned}\cos \theta - \cos \theta_0 &= \left(1 - 2 \sin^2 \frac{\theta}{2}\right) - \left(1 - 2 \sin^2 \frac{\theta_0}{2}\right) = 2 \left(\sin^2 \frac{\theta_0}{2} - \sin^2 \frac{\theta}{2}\right) = 2(k^2 - k^2 z^2) \\ &= 2k^2(1 - z^2).\end{aligned}$$

また

$$k dz = \frac{1}{2} \cos \frac{\theta}{2} \cdot d\theta$$

より

$$d\theta = \frac{2k dz}{\cos \frac{\theta}{2}} = \frac{2k dz}{\sqrt{1 - k^2 z^2}}.$$

(多分、(4) を、 $z(t)$ に関する微分方程式に変換する、とやればもっとすっきりした議論になるのでしよう。いつか書き直そう。)

これから

$$\begin{aligned}t &= \int_0^t dt = \int_0^\theta \frac{1}{\sqrt{2}\omega} \cdot \frac{1}{\sqrt{\cos \theta - \cos \theta_0}} d\theta = \frac{1}{\sqrt{2}\omega} \int_0^z \frac{1}{\sqrt{2k\sqrt{1-s^2}}} \cdot \frac{2k ds}{\sqrt{1-k^2s^2}} \\ &= \frac{1}{\omega} \int_0^z \frac{ds}{\sqrt{(1-s^2)(1-k^2s^2)}}.\end{aligned}$$

すなわち

$$\omega t = \int_0^z \frac{ds}{\sqrt{(1-s^2)(1-k^2s^2)}}.$$

これは、Jacobi の楕円関数 sn を用いると ($\omega t = \text{sn}^{-1}(z, k)$ であるから)

$$\text{sn}(\omega t, k) = z = \frac{1}{k} \sin \frac{\theta}{2}$$

と解ける。ゆえに

$$(8) \quad \theta(t) = 2 \sin^{-1}(k \text{sn}(\omega t, k)).$$

これが初期値問題 (2), (6) の厳密解である。

3 単振動との比較

前節の問題の単振動 (調和振動) 近似

$$\theta''(t) = -\omega^2 \theta(t), \quad \theta(0) = 0, \quad \theta'(0) = 2k\omega$$

の解は明らかに

$$(9) \quad \theta(t) = 2k \sin(\omega t).$$

$\theta_0 = 0.1, 0.2, 1$ の場合に、単振り子の運動と、単振動を比較してみる。以下の図は横軸が t 、縦軸が θ で、単振動の解を赤、振り子の解を緑で描いたものである。

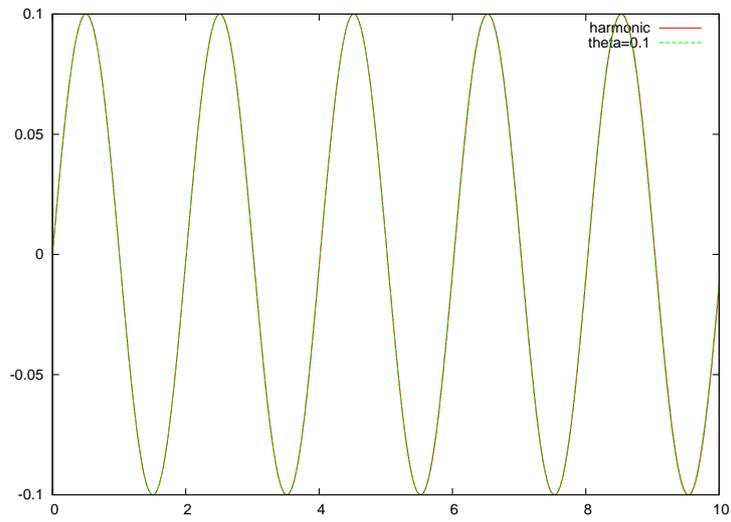


図 2: $\theta_0 = 0.1$: 振幅が小さいと単振動とほとんど同じ (上書きされて赤線が見えない)

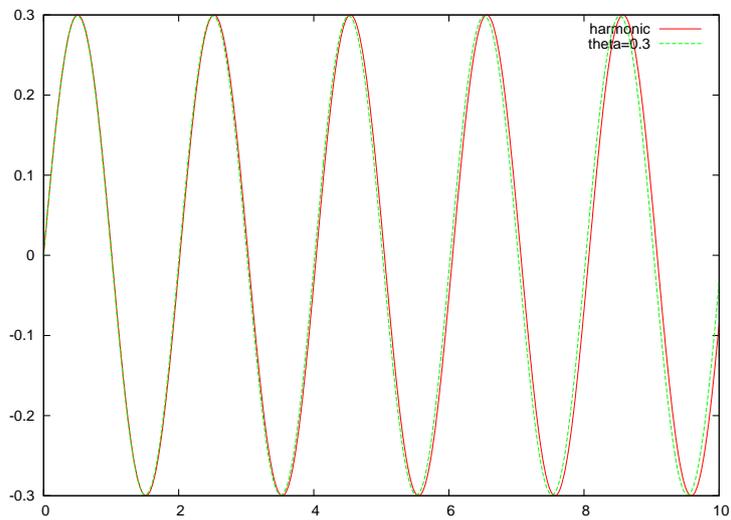


図 3: $\theta_0 = 0.3$: 振幅が大きくなると単振動からずれてくる

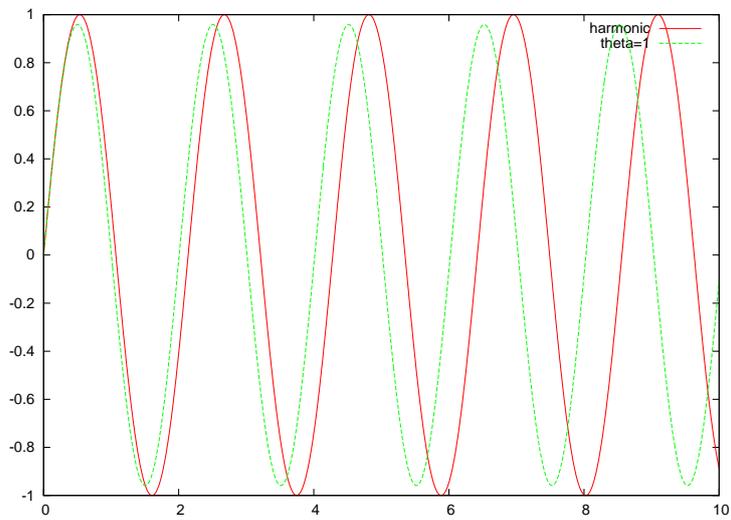


図 4: $\theta_0 = 1$ ($\cong 57.3^\circ$) となるとかなりずれる

4 「等時性」が成り立っているかを調べる

振幅が θ_0 である単振り子の周期は (s が 0 から 1 に変化するのに要する時間の 4 倍だから)

$$T = 4 \times \frac{1}{\omega} \int_0^1 \frac{ds}{\sqrt{(1-s^2)(1-k^2s^2)}} = 4\sqrt{\frac{\ell}{g}} K(k), \quad k := \sin \frac{\theta_0}{2}.$$

ただし $K(k)$ は k を母数とする第 1 種完全楕円積分である:

$$K(k) := \int_0^{\pi/2} \frac{dt}{\sqrt{1-k^2 \sin^2 t}}.$$

これと単振動の周期

$$T_h := 2\pi\sqrt{\frac{\ell}{g}}$$

との比

$$\frac{T}{T_h} = \frac{2}{\pi} K(k)$$

を具体的に計算してみよう。

$$\frac{T(\theta_0 = 1)}{T_h} = 1.06633424557996 \dots,$$

$$\frac{T(\theta_0 = 45^\circ)}{T_h} = 1.039973343196804 \dots,$$

$$\frac{T(\theta_0 = 90^\circ)}{T_h} = 1.180340599016096 \dots.$$

つまり $\theta_0 = 45^\circ$ でほぼ 4% しかずれていない。さすがに $\theta_0 = 90^\circ$ になると 18% ほどずれている。

$$K(k) = \frac{\pi}{2} \left[1 + \left(\frac{1}{2}\right)^2 k^2 + \left(\frac{1 \cdot 3}{2 \cdot 4}\right)^2 k^4 + \left(\frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}\right)^2 k^6 + \dots \right]$$

であるから

$$(10) \quad \frac{T}{T_h} = \frac{2}{\pi} K(k) = 1 + \left(\frac{1}{2}\right)^2 k^2 + \left(\frac{1 \cdot 3}{2 \cdot 4}\right)^2 k^4 + \left(\frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6}\right)^2 k^6 + \dots \doteq 1 + \frac{k^2}{4}.$$

$\theta_0 = 45^\circ$ のとき、 $k = \sin \frac{\pi}{8} = 0.38268 \dots$ であるから、 $1 + \frac{k^2}{4} = 1.0366 \dots$ 。あまり良い近似でないかな？(もう 1 項くらい取れば…)

| θ_0 (度) | T/T_h |
|----------------|--------------------|
| 0 | 1.0000000000000000 |
| 10 | 1.001907188143217 |
| 20 | 1.007669025791545 |
| 30 | 1.017408797595956 |
| 40 | 1.031340519130037 |
| 50 | 1.049782960623032 |
| 60 | 1.073182007149365 |
| 70 | 1.102144909639270 |
| 80 | 1.137492559923922 |
| 90 | 1.180340599016096 |
| 100 | 1.232229196737520 |
| 110 | 1.295339998876550 |
| 120 | 1.372880500618350 |
| 130 | 1.469819325894477 |
| 140 | 1.594446101177228 |
| 150 | 1.762203729503756 |
| 160 | 2.007507401244124 |
| 170 | 2.439362719673885 |
| 179 | 3.901065160389150 |

図 5 は、これをグラフにしたものである。90° < θ_0 < 180° はあまり現実的な振り子の運動と言えないから、0° < θ_0 < 90° の場合のみ注目すると、振り子の等時性はほどほどの精度で成立する、と言えそうである。

(振り子の等時性が成り立たないことは実験してみれば分かる、という人がいるけれど、かなり真剣に取り組まないとはっきりしないのではないか?と思う。)

A プログラム

この文書に示した数値計算結果を得るのに用いたプログラムを紹介する。

趣味色が強い。情報処理教室で再現してもらうためには、Mathematica でやるのがよいか?と考えて試しに書いてみたら、1行で済んだりして…

A.1 単振り子 vs 単振動

振幅 θ_0 を指定して、単振り子の運動 $\theta(t)$ と、同じ初期条件を課した単振動の運動 $\theta_{\text{harmonic}}(t)$ を計算する。Runge-Kutta 法のようなアルゴリズムを用いても良いが、ここでは **GSL** (GNU Scientific Library) のテストをかねて、Jacobi の楕円関数を使って計算した。

```
/*
 * furiko.c --- 単振り子の運動
 *  $\theta = 2 \arcsin(k \operatorname{sn}(\sqrt{g/l} t, k))$ 
 */

#include <stdio.h>
#include <math.h>
#include <gsl/gsl_specfunc.h>
```

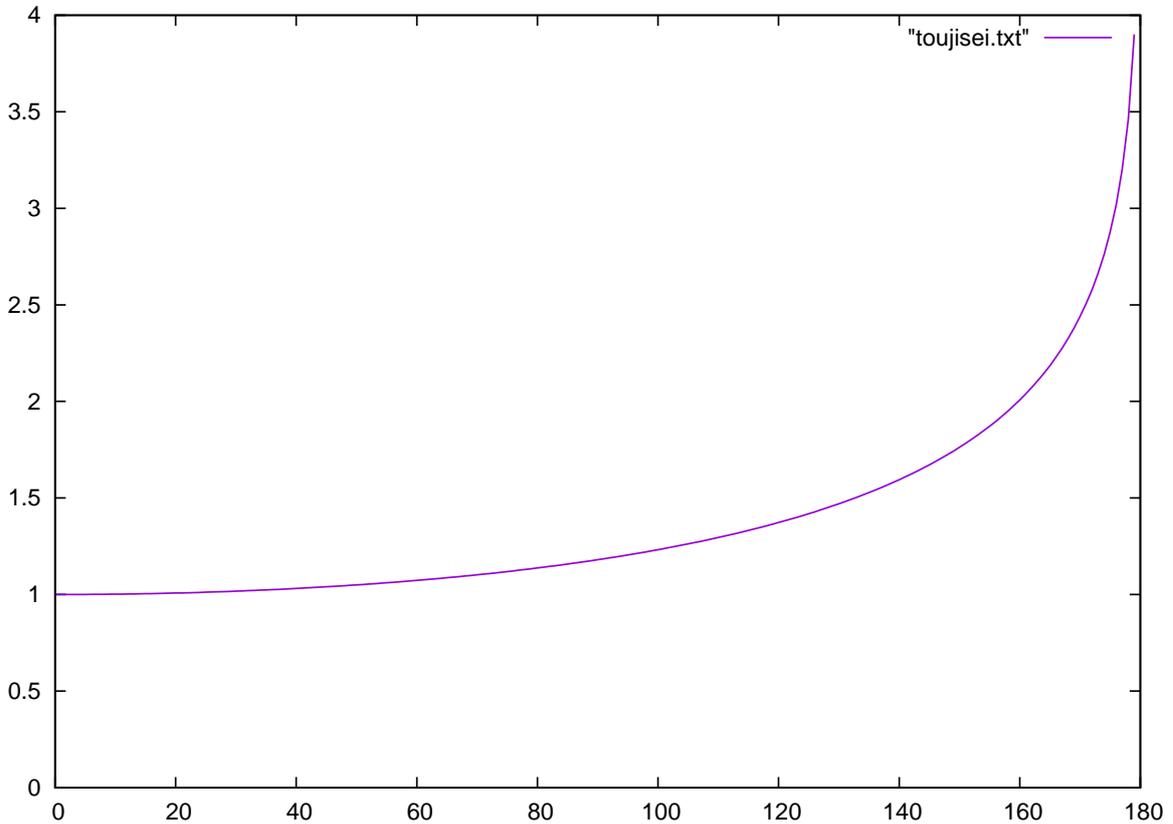


図 5: 振り子の等時性: 振幅で周期がどう変わるか

```
int main()
{
    double g,l,omega,theta0,k;
    int i,n;
    double t,dt,sn,cn,dn;

    g = 9.8;
    l = 1.0;
    omega = sqrt(g / l);
    scanf("%lf", &theta0);
    k = sin(theta0 / 2);

    dt = 0.01;
    for (i = 0; i <= 1000; i++) {
        t = i * dt;
        gsl_sf_elljac_e(omega * t, k*k, &sn, &cn, &dn);
        printf("%f %f %f\n", t, 2 * asin(k * sn), 2 * k * sin(omega * t));
    }
    return 0;
}
```

MacPorts で GSL をインストールした場合のコンパイル

```
gcc -I/opt/local/include furiko.c -L/opt/local/lib -lgsl
```

Mathematica ではこんな感じ

```
g=9.8; l=1; omega=Sqrt[g/l];
theta=1; k=Sin[theta/2];
Plot[{2k Sin[omega t], 2ArcSin[k JacobiSN[omega t,k^2]]}, {t,0,10}]
```

注意: これまで GSL では、

```
gsl_sf_elljac_e(omega * t, k, &sn, &cn, &dn) (間違い!)
```

Mathematica では

```
JacobiSN[omega t,k] (これも間違い!)
```

と書いて来たのですが、GSL, Mathematica の楕円関数、楕円積分関係の関数は、引数として k ではなく $m(=k^2)$ を使うようになっていることを教えていただいたので訂正しました (ご指摘に感謝します。2012/3/2)。

A.2 振り子の等時性

0 度から 179 度まで 1 度刻みの振幅に対して、振り子の周期を計算する。完全楕円積分であるから、算術幾何平均 (AGM) アルゴリズムを使って計算した (本当は GSL を呼び出せばよいわけで、半分道楽…いや、GSL を使わず標準的な関数だけですむのは多少意味があるか? 冪級数展開 (10) を使って計算する方が良いかな?)。

```
/*
 * toujisei.c --- 単振り子の等時性
 * T=4√ l/g K(k), k=sin θ 0/2
 * Th=2 π√ l/g
 * http://www.math.meiji.ac.jp/~mk/labo/text/prog/toujisei.c
 */

#include <stdio.h>
#include <math.h>

double pi;

/*
 * 算術幾何平均アルゴリズムで完全楕円積分 K(k), E(k) を求める
 */
int kanzen(double k, double *K, double *E)
{
    int i;
    double a,b,an,bn,cn,anp1,bnp1,cnp1;
    double power2, s, AGM;

    if (k < 0 || k >= 1) {
        fprintf(stderr, "k must be in [0,1)\n");
        *K = *E = 0.0;
        return 1;
    }
    a = 1.0; b = sqrt(1.0 - k * k);

    an = a; bn = b; cn = sqrt(an * an - bn * bn);

    /* s = 2^(n-1) * cn * cn の和 (n=0,1,2,...) */
    power2 = 0.5;
    s = power2 * cn * cn;

    for (i = 0; i <= 10; i++) {
        anp1 = (an + bn) / 2; bnp1 = sqrt(an * bn); cnp1 = (an - bn) / 2;
        an = anp1; bn = bnp1; cn = cnp1;
        /* printf("%20.15f, %20.15f, %20.15f\n", an, bn, cn); */

        power2 *= 2;
        s += power2 * cn * cn;
    }
}
```

```

    if (fabs(an-bn) < 2e-16)
        break;
}
if (fabs(an-bn)>1e-15) {
    fprintf(stderr,"k=%f: does not converge. i=%d, an=%f, bn=%f, |an-bn|=%e\n",
        k, i, an, bn, fabs(an-bn));
    return 1;
}
AGM = an;
*K = pi / 2 / AGM;
*E = (a - s) * (*K);
return 0;
}

int main()
{
    double theta,k,K,E;
    int deg;

    pi = 4 * atan(1.0);
    for (deg = 0; deg < 180; deg++) {
        theta = deg * pi / 180.0;
        k = sin(theta / 2);
        kanzen(k, &K, &E);
        printf("%2d %20.15f\n", deg, 2 * K / pi);
    }
    return 0;
}

```

Mathematica ではこんな感じ

```
Plot[2 EllipticK[Sin[t Degree/2]^2] / Pi, {t,0,179}, PlotRange->{0,4}]
```

B Runge-Kutta 法で解く

今さら数値計算する気がなくなる…まあ、気が向いたら。

と書いたのですが、歳は取りたくないもので、ずっと昔、授業(『振り子の運動』⁵) でやったことがありました。その計算でも、確かに振幅 1 ラジアンで 6.63% 周期が長くなっていました。問題として、完全楕円積分を級数展開で計算しろ、なんて書いてある(笑)。ともあれ、全然違う方法で計算して得た数値が良く一致したので、とりあえず安心。

C 周期を数値積分で求める

ははみまかりける年某月某日、学生 Y の襲撃に会う。何でも振り子の周期を求めるために、端点に特異性のある定積分を計算するように指示されたけど、自分でネットを徘徊して、変数変換によって特異性を除去してから台形則で計算したら(おお)、粗い分割で非常な高精度を達成できた、という。実質的に第 2 種完全楕円積分

$$K(k) = \int_0^{\pi/2} \frac{dt}{\sqrt{1 - k^2 \sin^2 t}}$$

の計算に帰着させたい。

⁵<http://nalab.mind.meiji.ac.jp/~mk/syori2-1992/syo2-92-06.pdf>

台形則で高精度…あれだな、きっと。ちょっとやってみよう。まずは Mathematica で。

$k = 1/2$ のときの $K(k)$ の値

```
f[t_,k_]:=1/Sqrt[1-k^2 Sin[t]^2]
```

```
k=1/2
```

```
g=Plot[f[t,k],{t,-Pi,Pi}]
```

被積分関数の周期は π か。

```
Integrate[f[t,k],{t,0,Pi/2}]
```

→ `EllipticK[1/4]` と表示されるけれど、これは Mathematica 方言で、本当は $K(1/2)$ ということだな。

```
N[%,50]
```

→ 1.6857503548125960428712036577990769895008008941411 となる。

ちなみに Mathematica で数値積分をするのは、`NIntegrate[]` だ。

```
NIntegrate[f[t, k], {t, 0, Pi/2}, AccuracyGoal -> 50, WorkingPrecision -> 60]
```

さて、それで台形則を試してみる。

```
TR[N_] := Block[{i, h, S}, h = (Pi/2)/N; S = 0;
```

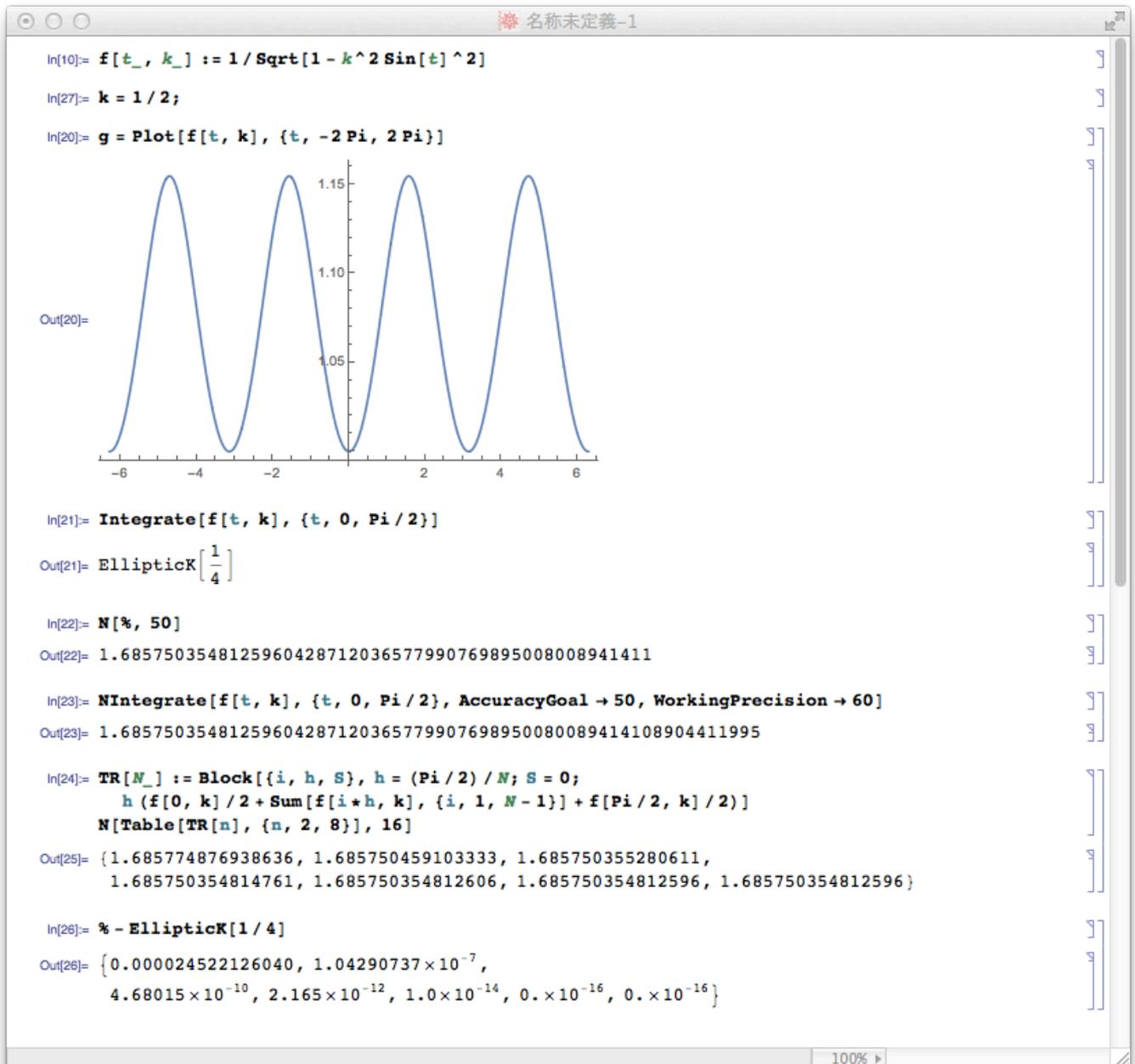
```
h (f[0, k]/2 + Sum[f[i*h, k], {i, 1, N - 1}] + f[Pi/2, k]/2)]
```

```
N[Table[TR[n], {n, 2, 8}], 16]
```

```
%-EllipticK[k^2]
```

```
{0.000024522126040, 1.04290737*10^-7, 4.68015*10^-10, 2.165*10^-12,  
1.0*10^-14, 0.*10^-16, 0.*10^-16}
```

なるほど、これは凄い。



実質的に、有名な

「解析的周期関数の1周期区間の積分は、台形則で非常に高精度に計算できる」

という数値積分の常識の実例に遭遇したわけだ。

Cのプログラムを示せば、

period.c

```
#include <stdio.h>
#include <math.h>

double k = 0.5;

double sqr(double x) { return x * x; }

double f(double x)
{
    return 1 / sqrt(1- sqr(k * sin(x)));
}

int main(void)
{
    int i, n;
    double h,s,pi;

    pi = 4 * atan(1.0);
    for (n = 2; n <= 8; n++) {
        h = pi / 2 / n;
        s = (f(0) + f(pi/2)) / 2;
        for (i = 1; i < n; i++)
            s += f(i*h);
        s *= h;
        printf("%2d %20.15f\n", n, s);
    }
}
```

```
$ gcc period.c
$ ./a.out
2    1.685774876938636
3    1.685750459103333
4    1.685750355280611
5    1.685750354814761
6    1.685750354812606
7    1.685750354812596
8    1.685750354812596
$
```

ふむふむ (区間を 7 等分で倍精度一杯の精度だ)。AGM とどっちが速いんだろう。

それにしても、自力でそういう工夫をして、高精度な結果が得られたことを不思議に思って追求して質問に来るとは、大したものだと思う。