計算数理演習（担当：齊藤宣一先生）第 2 回レポート

司馬博文 05-210520

2021 年 4 月 23 日

課題.

(1) π の近似値を求める算譜の精度を改善せよ.
(2) Fourier 級数展開の収束の様子を視覚的に確かめよ.

1 精度改善

1.1 問題点

数列 $(\sigma_{2^n})_{n\in\mathbb{N}}$ の漸化式

$$\sigma_{2n} = \sqrt{\frac{1-\sqrt{1-\sigma_n^2}}{2}} \tag{1}$$

の計算の際，$\sigma_n \to 0$ より，$\sqrt{1-\sigma_n^2} \to 1$ であるから，$1-\sqrt{1-\sigma_n^2}$ での桁落ちは避けられない．したがって，このように，値が近似する 2 値の減算を避ける算譜に変えれば良い．

1.2 代替案

漸化式 1 を変形すると，

$$4\sigma_{2n}^4 - 4\sigma_{2n}^2 + 4\sigma_n^2 = 0$$

であり，この解 $0 < \sigma_{2n}^2 < 1$ を求めたい．すると，$\sigma_{2n} > 0$ より，この平方根を取れば良い．この解を求めるに当たって，桁落ちの危険のない大きい方の解

$$d^2 = \frac{1+\sqrt{1-\sigma_n^2}}{2}$$

を求めてから，解と係数の関係 $\sigma_{2n}^2 = \dfrac{\sigma_n^2}{4d^2}$ を用いて σ_{2n} を求めれば良い．なお，プログラム中では d ではなく dual とした．

算譜 1　代替案

```
function res = comp_pi3
    sig = 1/2; p = 6; n = 6; res = [];
    while sig > 1e-10
        dual = (1+sqrt(1-sig*sig))/2;
        sig = sqrt((sig*sig)/(4*dual));
        p = 4*n*sig;
        n = 2*n;
        res = [res;n,p/2,p/2-pi];
    end
end
```

算譜 2　出力

```
>> format longE
>> comp_pi3
```

```
ans =
     1.200000000000000e+01 3.105828541230249e+00 -3.576411235954424e-02
     2.400000000000000e+01 3.132628613281238e+00 -8.964040308555354e-03
     4.800000000000000e+01 3.139350203046866e+00 -2.242450542926822e-03
     9.600000000000000e+01 3.141031950890508e+00 -5.607026992846542e-04
     1.920000000000000e+02 3.141452472285461e+00 -1.401813043320210e-04
     3.840000000000000e+02 3.141557607911857e+00 -3.504567793655156e-05
     7.680000000000000e+02 3.141583892148317e+00 -8.761441475879650e-06
     1.536000000000000e+03 3.141590463228049e+00 -2.190361744425218e-06
     3.072000000000000e+03 3.141592105999270e+00 -5.475905231477896e-07
     6.144000000000000e+03 3.141592516692156e+00 -1.368976372262409e-07
     1.228800000000000e+04 3.141592619365381e+00 -3.422441174905089e-08
     2.457600000000000e+04 3.141592645033688e+00 -8.556104713619561e-09
     4.915200000000000e+04 3.141592651450765e+00 -2.139028509873242e-09
     9.830400000000000e+04 3.141592653055034e+00 -5.347589038251499e-10
     1.966080000000000e+05 3.141592653456101e+00 -1.336921684469417e-10
     3.932160000000000e+05 3.141592653556367e+00 -3.342570664699451e-11
     7.864320000000000e+05 3.141592653581434e+00 -8.358647107797879e-12
     1.572864000000000e+06 3.141592653587701e+00 -2.091660178393795e-12
     3.145728000000000e+06 3.141592653589268e+00 -5.249134460427740e-13
     6.291456000000000e+06 3.141592653589660e+00 -1.332267629550188e-13
     1.258291200000000e+07 3.141592653589758e+00 -3.552713678800501e-14
     2.516582400000000e+07 3.141592653589782e+00 -1.065814103640150e-14
     5.033164800000000e+07 3.141592653589788e+00 -4.884981308350689e-15
     1.006632960000000e+08 3.141592653589790e+00 -3.552713678800501e-15
     2.013265920000000e+08 3.141592653589790e+00 -2.664535259100376e-15
     4.026531840000000e+08 3.141592653589790e+00 -2.664535259100376e-15
     8.053063680000000e+08 3.141592653589790e+00 -2.664535259100376e-15
     1.610612736000000e+09 3.141592653589790e+00 -2.664535259100376e-15
     3.221225472000000e+09 3.141592653589790e+00 -2.664535259100376e-15
     6.442450944000000e+09 3.141592653589790e+00 -2.664535259100376e-15
     1.288490188800000e+10 3.141592653589790e+00 -2.664535259100376e-15
     2.576980377600000e+10 3.141592653589790e+00 -2.664535259100376e-15
     5.153960755200000e+10 3.141592653589790e+00 -2.664535259100376e-15
```

資料中のプログラム comp_pi2.m よりも，極めて良い精度で（ほぼ計算機イプシロンギリギリまで）計算できていることが確認できる.

2 Fourier 級数の収束

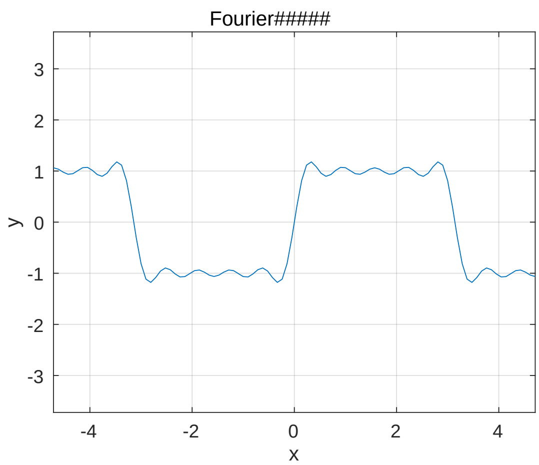
2.1 プログラム内容

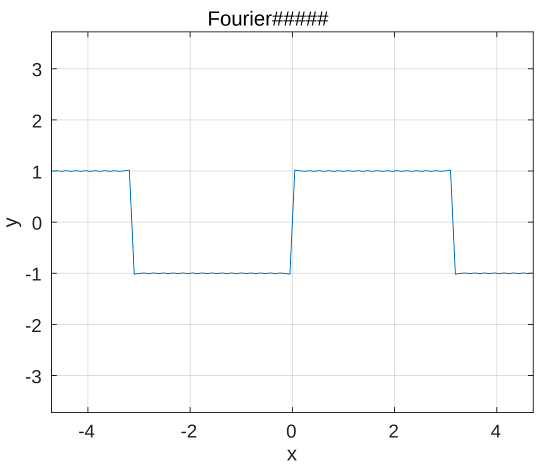
算譜 3　プログラム

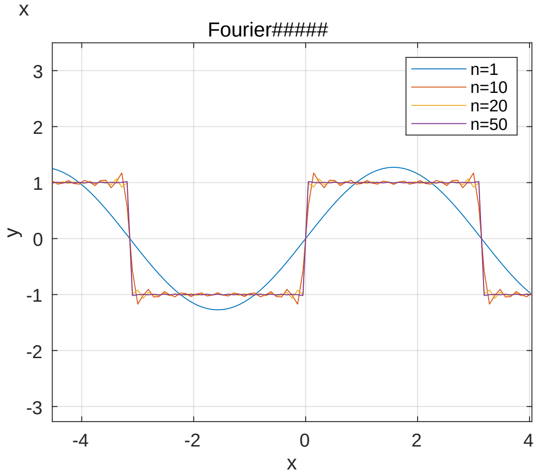
```
four_draw1(5, @four2, 100, "four1")
four_draw1(50, @four2, 100, "four2")
four_draw2(@four2, 100, "four3")

function y = four2(x,n)
    y = sin(x);
    for i=2:n
        y = y + sin((2*i-1)*x)/(2*i-1);
    end
    y = 4/pi * y;
end
```

```
function four_draw1(n, fun, m, filename)
    x = linspace(-1.5*pi, 1.5*pi, m)';
    y = fun(x, n);
    figure(1); plot(x,y);
    axis equal;
    grid on; title("級数の収束 Fourier");
    xlabel('x'); ylabel('y');
    % saveas(1,"../"+filename+'.pdf');
end

function four_draw2(fun, m, filename)
    x = linspace(-1.5*pi, 1.5*pi, m)';
    yy = [];
    y = fun(x,1); yy = [yy,y];
    y = fun(x,10); yy = [yy,y];
    y = fun(x,20); yy = [yy,y];
    y = fun(x,50); yy = [yy,y];
    figure(1); plot(x,yy);
    axis equal;
    grid on; title("級数の収束 Fourier");
    xlabel("x"); ylabel("y");
    legend("n=1","n=10","n=20","n=50");
    % saveas(1,filename+".pdf");
end
```

2.2 出力結果







$n = 5$ の時点ですでに相当収束しており，$n = 50$ となると $x = -\pi, 0, \pi$ 周りでの傾きが有限であることは視認できるが，ほとんど三角関数の振動は見えない．