第2課（つづき）

2-5 初等関数

2-5-1　べき乗とべき乗根

x^n

x^0=1（定義）

$x^{1/n}$

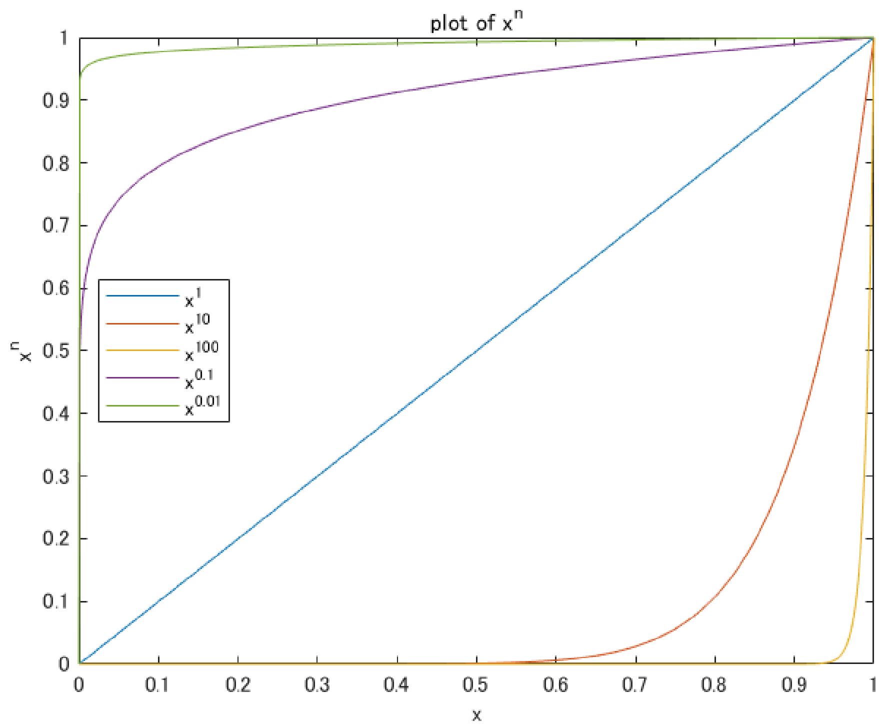
$x=y^n \to x^{1/n}=y$　　（逆関数）

x^a　ただし $a=m/n$ （有理数）$(x^m)^{1/n}$：整数べきおよびべき乗根の組み合わせ

x^a　ただしaが無理数の場合にはその最良有理数近似で置き換えると考えてよい．以下で図を描いて見るように、この関数はaに関して連続である．

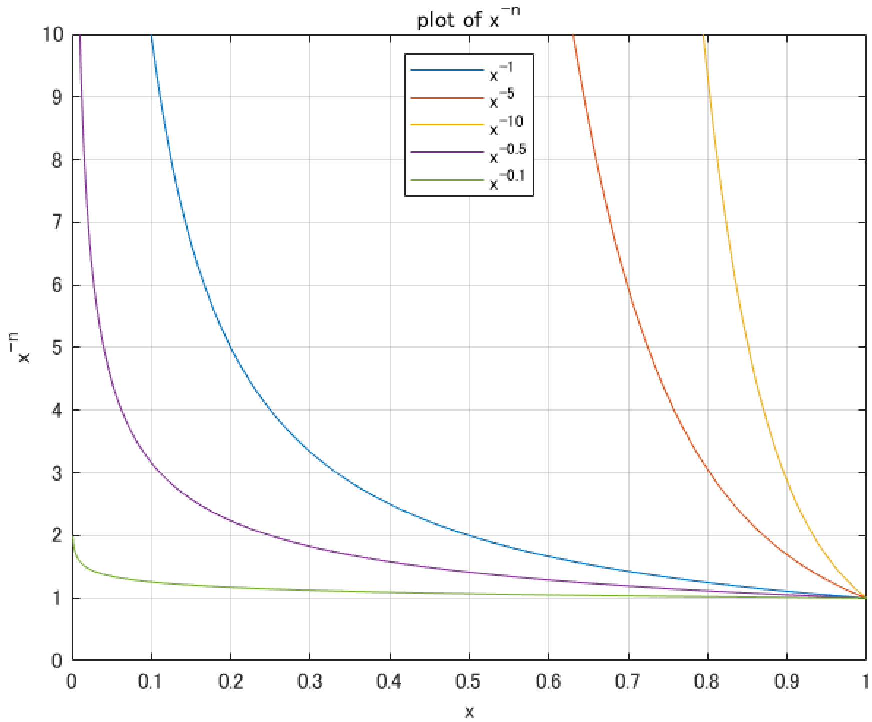
ここでは，これで充分だろう．

```
clf
x=(0:0.001:1);
y1=x.^1; plot(x,y1);hold on
y10=x.^10; plot(x,y10);hold on
y100=x.^100; plot(x,y100);hold on
y01=x.^0.1; plot(x,y01);hold on
y001=x.^0.01; plot(x,y001);hold on
xlabel('x'); ylabel('x^n');title('plot of x^n')
legend('x^1','x^{10}','x^{100}','x^{0.1}','x^{0.01}','Location','west')
```



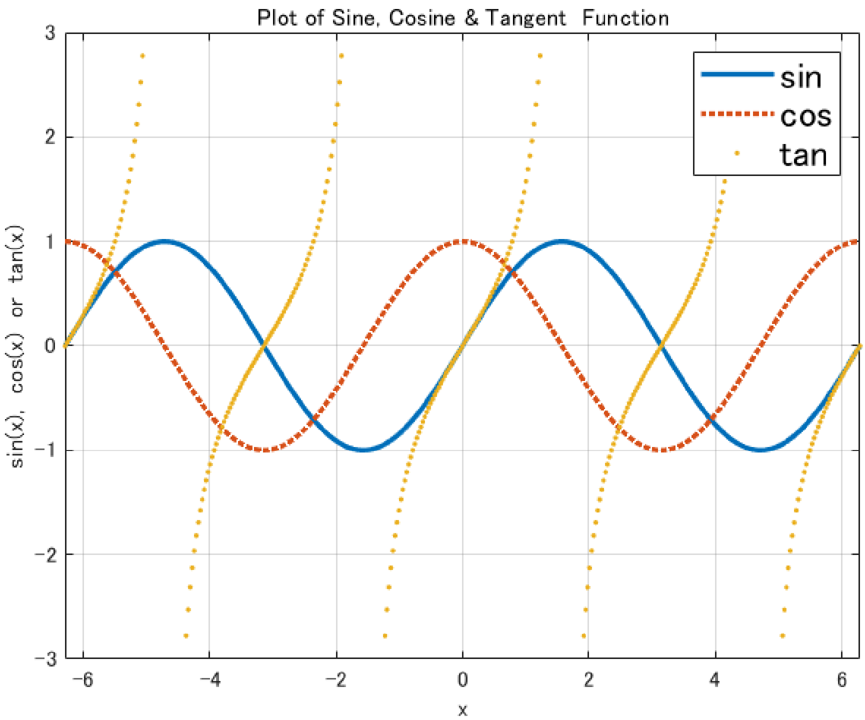
$x^{-n} = 1/x^n$

```
clf
x=(0:0.001:1);
ym1=x.^(-1); plot(x,ym1);hold on
ym5=x.^(-5); plot(x,ym5);hold on
ym10=x.^(-10); plot(x,ym10);hold on
ym05=x.^(-0.5); plot(x,ym05);hold on
ym01=x.^(-0.1); plot(x,ym01);hold on
ylim([0 10])
xlabel('x'); ylabel('x^{-n}');title('plot of x^{-n}')
legend('x^{-1}','x^{-5}','x^{-10}','x^{-0.5}','x^{-0.1}','Location','north')
grid on
```



2-5-2　三角関数

```
clf
x=-2*pi:pi/100:2*pi;
y1=sin(x);  plot(x,y1,'LineWidth',2)
hold on;
y2=cos(x);  plot(x,y2,':','LineWidth',2)
hold on;
y3=tan(x); plot(x,y3,'.','LineWidth',2)
ylim([-3 3]); xlim([-2*pi 2*pi]);grid on
xlabel('x');ylabel('sin(x),  cos(x)  or  tan(x)');
title('Plot of Sine, Cosine & Tangent  Function')
legend('sin','cos', 'tan','FontSize',15)
```



2-5-3　指数関数と対数関数

指数関数と**Napier**数

べき乗の関数は（意外と？）重要だが、特に 10^x （こちらは特に説明は要らない？）や 或る数 e のべき乗は重要.

$e^x, \exp(x)$

e をNapier数という．その値は

```
e=exp(1)
```

```
e = 2.7183
```

```
vpa(e)
```

ans = $2.7182818284590455348848081484903$

これは無理数．以下のように定義される．

$$e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n$$

```
format long
(1+1/2)^2
```

```
ans =
   2.250000000000000
```

```
(1+1/10)^10
```

```
ans =
   2.593742460100002
```

```
(1+1/100)^100
```

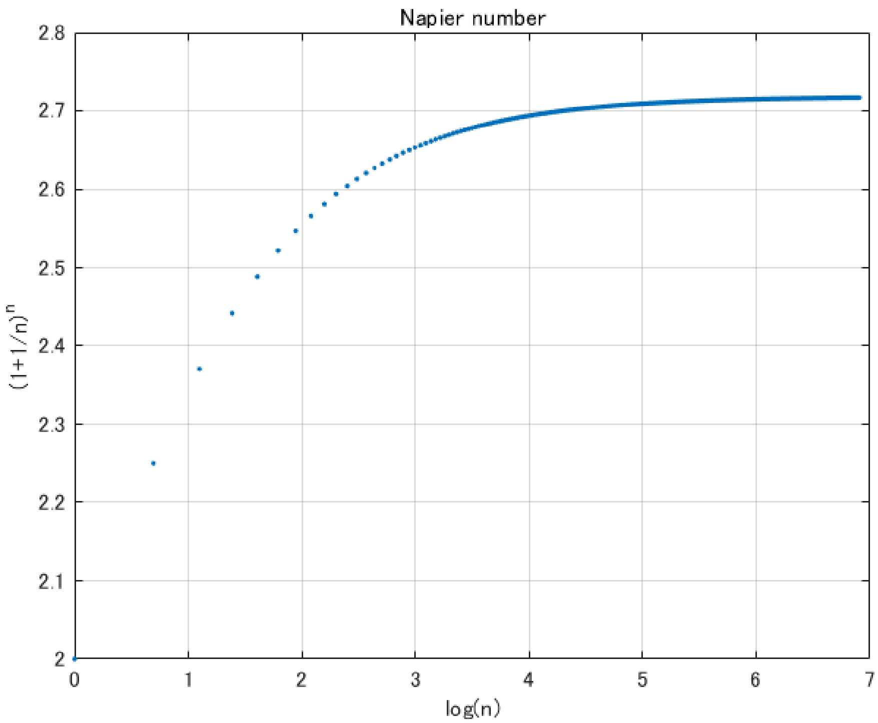
```
ans =
   2.704813829421528
```

```
(1+1/10000)^10000
```

```
ans =
   2.718145926824926
```

Napier数への収束が大変ゆっくりしていることが見てとれる.

```
format
%
z=(1:1000);
u=(1:1000);
f=(1:1000);
for n=1:1000
 u(n)=n;
 zp=log(n);
 fp=(1.0+1.0./n).^n;
 z(n)=zp;
 f(n)=fp;
end
clf
plot(z,f,'.'); hold on
xlabel('log(n)');ylabel('(1+1/n)^n')
title('Napier number')
grid on
```

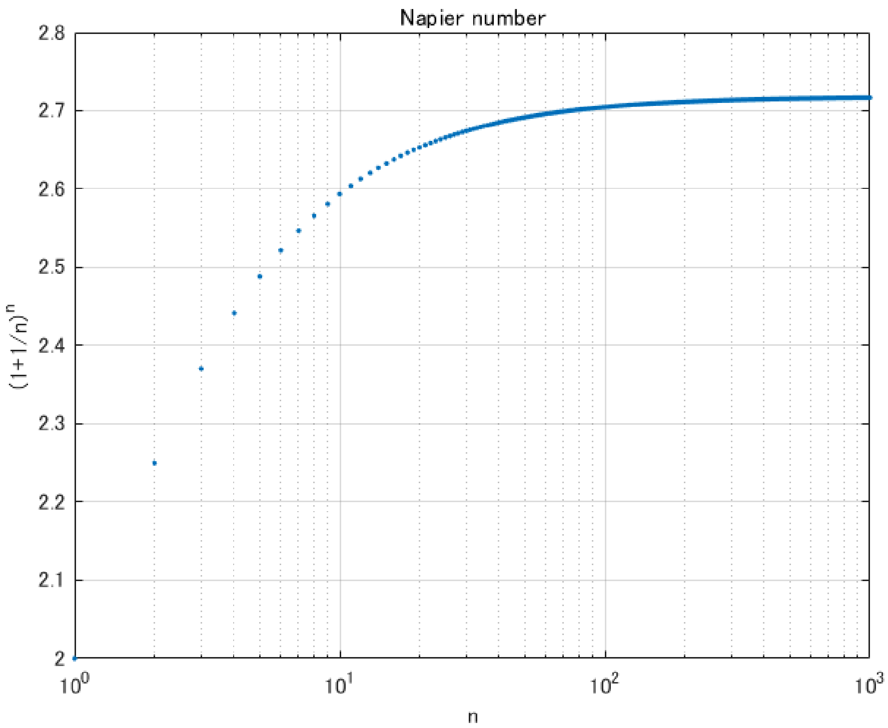


ここで横軸に $\log(n)$ なるものをとった（対数関数一後述）．横軸に n をとると，横に長い長いグラフとなるからである．

これを片対数グラフと呼んでいる．

片対数グラフを用いると便利．横軸は 10^x の x をとっている．

```
clf
semilogx(u,f,'.'); hold on
xlabel('n'); ylabel('(1+1/n)^n')
title('Napier number')
grid on
```



なぜNapier数が重要か．それは複利計算において発見されたと言われる．

「元金1を年利1、付利期間を1/n 年で1年預金すれば、1/n 年ごとに利子1/n で元利合計が増えていき、1年経つと右辺の式になる。$n \to +\infty$ とした極限は連続複利の元利合計となる。」（Wikipedia）

Wikipediaの「連続複利」の項には以下のように記述されている．

元本を a，年利を p とする．

1) まず、預金期間 n 年、年利 p とし，預金期間後の元利合計を考える．

元利合計は、 $a(1+p)^n$

2) 付利期間を1/2年（半年毎）とし、各期の利率を $p/2$ にしてみる。

預金期間 n 年の元利合計は、 $a((1+p/2)(1+p/2))^n = a\left(1+\frac{p}{2}\right)^{2n}$

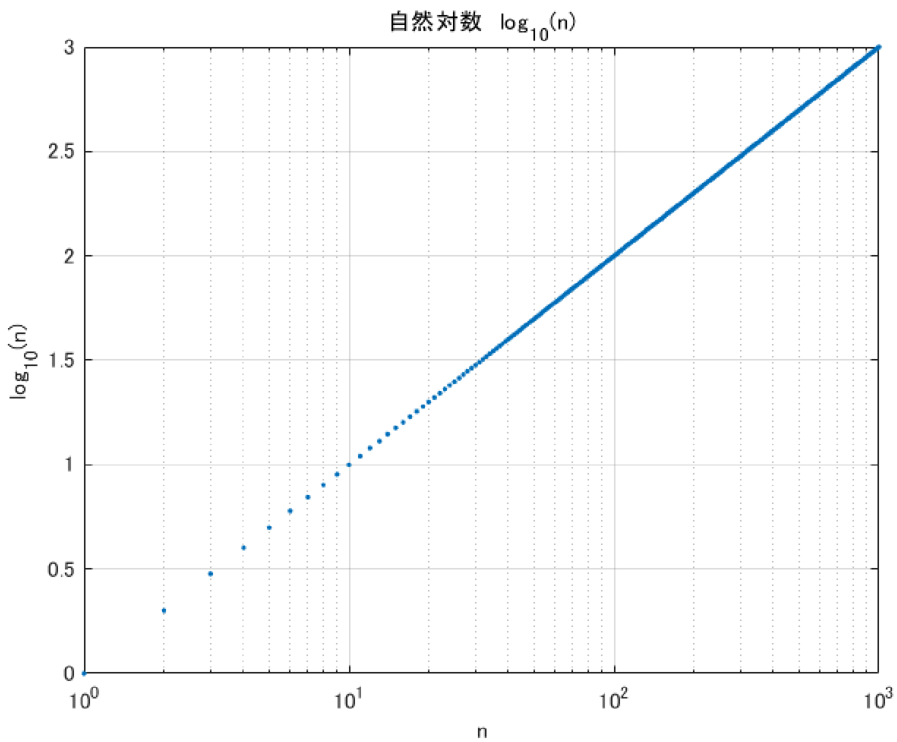
3) 同様に付利期間を $\frac{1}{k}$ 年、各期の利率 $\frac{p}{k}$ の預金期間 n 年の元利合計は $a\left(1+\frac{p}{k}\right)^{kn}$.

4) $\frac{k}{p} = K$ とおくと、この式は $a\left(1+\frac{1}{K}\right)^{npK} \to ae^{np}$

（片）対数グラフ

```
%対数グラフ semilogx
clf
%
z=(1:1000);
```

```
f=(1:1000);
for n=1:1000
 z(n)=n;
 f(n)=log10(n);
end
clf
semilogx(z,f,'.'); hold on
xlabel('n');ylabel('log_{10}(n)')
title('自然対数  log_{10}(n)')
grid on
```




```
%対数グラフsemilogy
clf
x=(0.01:0.01:5);
y0=10.^x;
semilogy(x,y0,'.');xlabel('x');ylabel('10^x');legend('10^x');grid on
```

指数関数

```
% 指数関数
clf
x=(0.01:0.01:5);
y1=exp(x);
plot(x,y1,'.');legend('exp(x)');grid on
```

対数関数

対数関数は指数関数の「逆関数」として定義される．

Napier数のベキ乗として定義した

$y \to x$ の関係（x を y の「関数」として） $x = \exp(y)$

を，逆に $x \to y$ の関係としてみたときこれを

$y = \log(x)$

と書く．これを自然対数あるいは単に対数という．

```
% 対数関数
clf
x=(0.01:0.01:5);
xlog=log(x);x2=x.^2;x5=x.^5;
xm2=-x.^(-2);xm5=-x.^(-5);
plot(x,xlog,'.');hold on
plot(x,x2,'x-');plot(x,x5);hold on
plot(x,xm2,'x-');plot(x,xm5);hold on
legend('log(x)','x^2','x^5','-x^{-2}','-x^{-5}','Location','south');hold on
xlim([0,0.1]);ylim([-10,10]);grid on
```

- 指数関数 $\log(x)$ はいかなるべき関数 x^n よりゆっくり$\cdots$する．
- 指数関数 $\log(x)$ はいかなるべき関数 x^{-n} より早く$\cdots$する．
- $a^0 = 1$

課題

（1）対数関数の振舞$(x \to 0, x = 0, x \to \infty$ などを調べよ．

（2）それぞれの図の体裁を考え，線の太さや色、線種，凡例の書き方，大きさなどを工夫せよ．

（3）対数関数の最後に書いた事項を確かめよ．