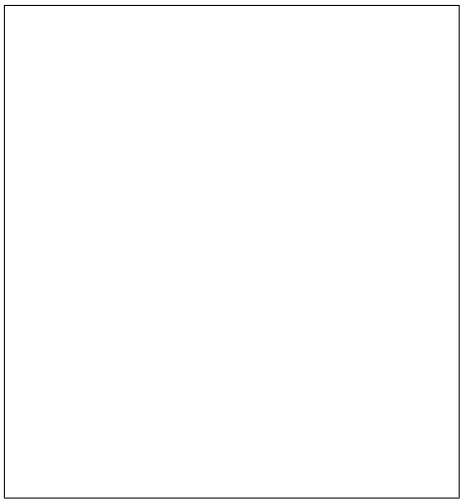
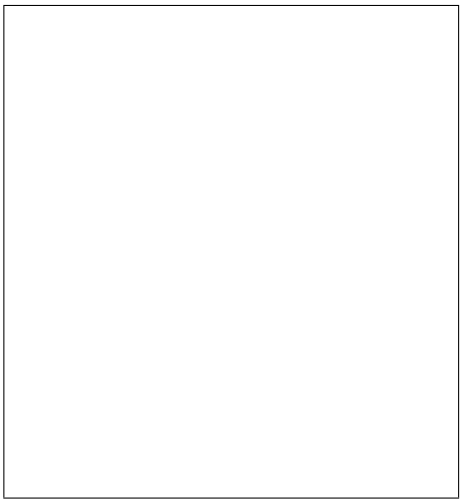
第1章　補間 (interpolation) と数値積分 (Integral)

1.1　概要:補間と近似

単純な 2 次元データについて補間と近似を考える．補間はたんに点をつなぐことを，近似はある関数にできるだけ近くなるようにフィットすることを言う．補間は Illustrator などのドロー系ツールで曲線を引くときの，ベジエやスプライン補間の基本となる．本章では補間とそれに密接に関連した積分について述べる．

表 1.1: 補間と近似の模式図.

補間	近似
	

1.2　多項式補間 (polynomial interpolation)

データを単純に多項式で補間する方法を先ず示そう．$N+1$ 点を N 次の多項式でつなぐ．この場合の補間関数は，

$$F(x) = \sum_{i=0}^{N} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_N x^N$$

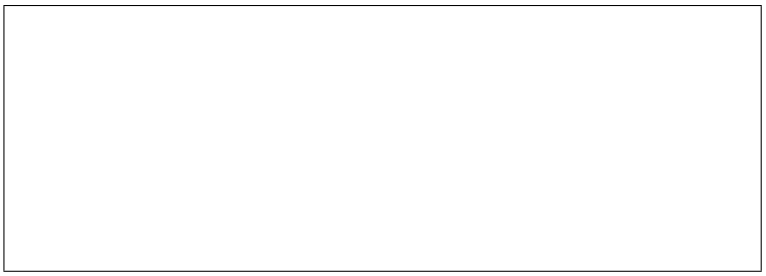
である．データの点を $(x_i, y_i), i = 0..N$ とすると

$$\begin{aligned} a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_N x_0^N &= y_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 + \cdots + a_N x_1^N &= y_1 \\ &\vdots \\ a_0 + a_1 x_N + a_2 x_N^2 + \cdots + a_N x_N^N &= y_N \end{aligned}$$

が，係数 a_i を未知数と見なした線形の連立方程式となっている．係数行列は

$$A = \begin{bmatrix} 1 & x_0 & x_0^2 & \cdots & x_0^N \\ 1 & x_1 & x_1^2 & \cdots & x_1^N \\ \vdots & & & & \vdots \\ 1 & x_N & x_N^2 & \cdots & x_N^N \end{bmatrix}$$

となる．a_i と y_i をそれぞれベクトルとみなすと



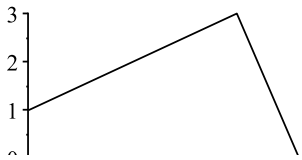
により未知数ベクトル a_i が求まる．これは単純に，前に紹介した Gauss の消去法や LU 分解で解ける．

1.2.1 Maple による多項式補間の実例

```
> restart; X:=[0,1,2,3]: Y:=[1,2,3,-2]:
> with(LinearAlgebra):
> list1:=[X,Y];
```

$$list1 := [[0,1,2,3],[1,2,3,-2]]$$

```
> with(plots):
  l1p:=listplot(Transpose(Matrix(list1))):
  display(l1p);
```




```
> A:=Matrix(4,4):
  for i from 1 to 4 do
    for j from 1 to 4 do
      A[i,j]:=X[i]^(j-1);
    end do;
  end do:
  A;
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 2 & 4 & 8 \\ 1 & 3 & 9 & 27 \end{bmatrix}$$

```
> a1:=MatrixInverse(A).Vector(Y);
```

$$a1 := \begin{bmatrix} 1 \\ -1 \\ 3 \\ -1 \end{bmatrix}$$

```
> f1:=unapply(add(a1[ii]*x^(ii-1),ii=1..4),x);
```

$$f1 := x \mapsto 1 - x + 3\,x^2 - x^3$$

```
> f1p:=plot(f1(x),x=0..3):
  l1p:=listplot(Transpose(Matrix(list1))):
  display(f1p,l1p);
```

1.3 Lagrange(ラグランジュ) の内挿公式

多項式補間は手続きが簡単であるため，計算間違いが少なく，プログラムとして組むのに適している．しかし，あまり”みとうし”のよい方法とはいえない．その点，Lagrange(ラグランジュ) の内挿公式は見通しがよい．これは

$$F(x) = \sum_{k=0}^{N} \frac{\prod_{j\neq k}(x - x_j)}{\prod_{j\neq k}(x_k - x_j)} y_k = \sum_{k=0}^{N} \frac{\dfrac{(x-x_0)(x-x_1)\cdots(x-x_N)}{(x-x_k)}}{\dfrac{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_N)}{(x_k-x_k)}} y_k$$

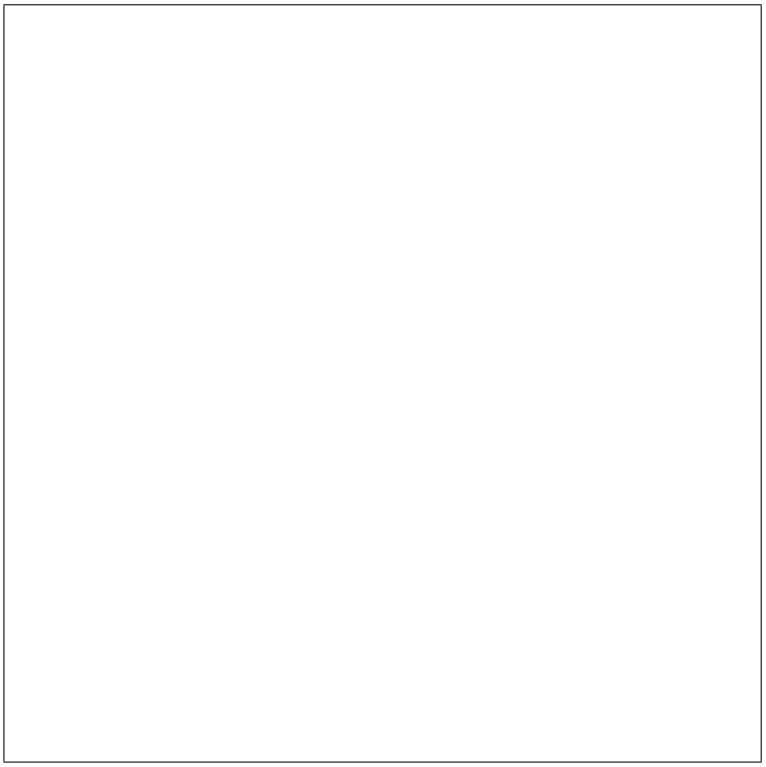
と表わされる．数学的に 2 つ目の表記は間違っているが，先に割り算を実行すると読み取って欲しい．これは一見複雑に見えるが，単純な発想から出発している．求めたい関数 $F(x)$ を

$$F(x) = y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x)$$

とすると

$$\begin{array}{lll} L_0(x_0) = 1 & L_0(x_1) = 0 & L_0(x_2) = 0 \\ L_1(x_0) = 0 & L_1(x_1) = 1 & L_1(x_2) = 0 \\ L_2(x_0) = 0 & L_2(x_1) = 0 & L_2(x_2) = 1 \end{array}$$

となるように関数 $L_i(x)$ を決めればよい．これを以下のようにとれば Lagrange の内挿公式となる．



1.4 Newton(ニュートン) の差分商公式

もう一つ有名な Newton(ニュートン) の内挿公式は，

$$F(x) = F(x_0) + (x-x_0)f_1\lfloor x_0, x_1\rfloor + (x-x_0)(x-x_1)f_2\lfloor x_0, x_1, x_2\rfloor + \cdots + \prod_{i=0}^{n-1}(x-x_i)\, f_n\lfloor x_0, x_1, \cdots, x_n\rfloor$$

となる．ここで $f_i\lfloor\ \rfloor$ は次のような関数を意味していて，

$$\begin{aligned} f_1\lfloor x_0, x_1\rfloor &= \frac{y_1 - y_0}{x_1 - x_0} \\ f_2\lfloor x_0, x_1, x_2\rfloor &= \frac{f_1\lfloor x_1, x_2\rfloor - f_1\lfloor x_0, x_1\rfloor}{x_2 - x_0} \\ &\vdots \\ f_n\lfloor x_0, x_1, \cdots, x_n\rfloor &= \frac{f_{n-1}\lfloor x_1, x_2 \cdots, x_n\rfloor - f_{n-1}\lfloor x_0, x_1, \cdots, x_{n-1}\rfloor}{x_n - x_0} \end{aligned}$$

差分商と呼ばれる．得られた多項式は，Lagrange の内挿公式で得られたものと当然一致する．Newton の内挿公式の利点は，新たなデータ点が増えたときに，新たな項を加えるだけで，内挿式が得られる点である．

1.4.1 Newton 補間と多項式補間の一致の検証

関数 $F(x)$ を x の多項式として展開．その時の，係数の取るべき値と，差分商で得られる値が一致．

```
> restart: F:=x->f0+(x-x0)*f1p+(x-x0)*(x-x1)*f2p;
```

$$F := x \mapsto f0 + (x - x0)f1p + (x - x0)(x - x1)f2p$$

```
> F(x1);
  sf1p:=solve(F(x1)=f1,f1p);
```

$$f0 + (x1 - x0)f1p$$

$$sf1p := \frac{f0 - f1}{-x1 + x0}$$

f20 の取るべき値の導出

```
> sf2p:=solve(F(x2)=f2,f2p);
  fac_f2p:=factor(subs(f1p=sf1p,sf2p));
```

$$sf2p := -\frac{f0 + f1p\,x2 - f1p\,x0 - f2}{(-x2 + x0)(-x2 + x1)}$$

$$fac\_f2p := \frac{f0\,x1 - x2\,f0 + x2\,f1 - x0\,f1 - f2\,x1 + f2\,x0}{(-x1 + x0)(-x2 + x0)(-x2 + x1)}$$

ニュートンの差分商公式を変形

```
> ff11:=(f0-f1)/(x0-x1);
  ff12:=(f1-f2)/(x1-x2);
  ff2:=(ff11-ff12)/(x0-x2);
  fac_newton:=factor(ff2);
```

$$ff11 := \frac{f0 - f1}{-x1 + x0}$$

$$ff12 := \frac{f1 - f2}{-x2 + x1}$$

$$ff2 := \frac{\frac{f0 - f1}{-x1 + x0} - \frac{f1 - f2}{-x2 + x1}}{-x2 + x0}$$

$$fac\_newton := \frac{f0\,x1 - x2\,f0 + x2\,f1 - x0\,f1 - f2\,x1 + f2\,x0}{(-x1 + x0)(-x2 + x0)(-x2 + x1)}$$

二式が等しいかどうかを evalb で判定

```
> evalb(fac_f2p=fac_newton);
```

$$true$$

1.5 数値積分 (Numerical integration)

積分,

$$I = \int_a^b f(x)dx$$

を求めよう．1 次元の数値積分法では連続した領域を細かい短冊に分けて，それぞれの面積を寄せ集めることに相当する．分点の数を N とすると，

$$x_i = a + \frac{b - a}{N} i = a + h \times i$$

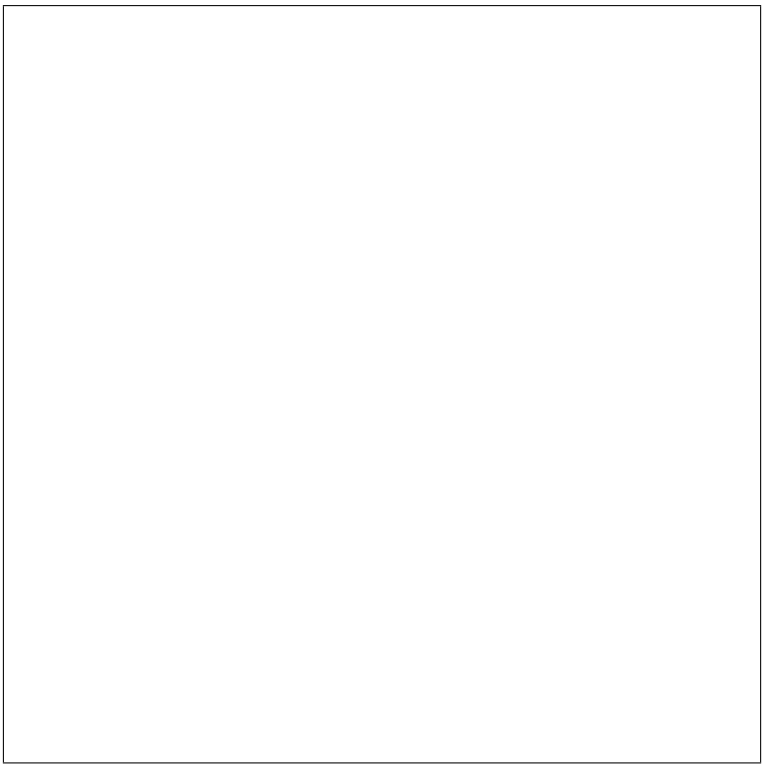
$$h = \frac{b - a}{N}$$

ととれる．そうすると，もっとも単純には，

$$I_N = \left\{ \sum_{i=0}^{N-1} f(x_i) \right\} h = \left\{ \sum_{i=0}^{N-1} f(a + i \times h) \right\} h$$

となる．

表 1.2: 数値積分の模式図.



1.5.1 中点則 (midpoint rule)

中点法 (midpoint rule) は，短冊を左端から書くのではなく，真ん中から書くことに対応し，

$$I_N = \left\{ \sum_{i=0}^{N-1} f\left(a + \left(i + \frac{1}{2} \right) \times h \right) \right\} h$$

となる.

1.5.2 台形則 (trapezoidal rule)

さらに短冊の上側を斜めにして，短冊を台形にすれば精度が上がりそうに思う．その場合は，短冊一枚の面積 S_i は,

$$S_i = \frac{f(x_i) + f(x_{i+1})}{2} h$$

で求まる．これを端から端まで加えあわせると,

$$i_N = \sum_{i=0}^{N-1} S_i = h \left\{ \frac{1}{2} f(x_0) + \sum_{i=1}^{N-1} f(x_i) + \frac{1}{2} f\left(x_N\right) \right\}$$

が得られる.

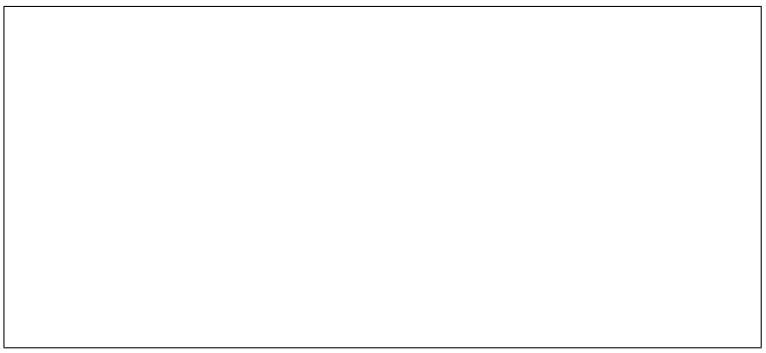
1.5.3 Simpson(シンプソン) 則

Simpson(シンプソン) 則では，短冊を 2 次関数,

$$f(x) = ax^2 + bx + c$$

で近似することに対応する．こうすると，

$$S_i = \int_{x_i}^{x_{i+1}} f(x)\,dx = \int_{x_i}^{x_{i+1}} (ax^2 + bx + c)\,dx$$



$$\frac{h}{6}\left\{f(x_i) + 4f\left(x_i + \frac{h}{2}\right) + f(x_i + h)\right\}$$

となる．これより，

$$I_N = \frac{h}{6}\left\{f(x_0) + 4\sum_{i=0}^{N-1} f\left(x_i + \frac{h}{2}\right) + 2\sum_{i=1}^{N-1} f(x_i) + f(x_N)\right\}$$

として計算できる．ただし，関数値を計算する点の数は台形則などの倍となっている．

教科書によっては，分割数 N を偶数にして，点を偶数番目 (even) と奇数番目 (odd) に分けて，

$$I_N = \frac{h}{3}\left\{f(x_0) + 4\sum_{i=even}^{N-2} f\left(x_i + \frac{h}{2}\right) + 2\sum_{i=odd}^{N-1} f(x_i) + f(x_N)\right\}$$

としている記述があるが，同じ計算になるので誤解せぬよう．

1.6　数値積分のコード

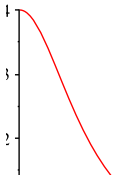
次の積分を例に，Maple のコードを示す．

$$\int_0^1 \frac{4}{1+x^2}\,dx$$

先ずは問題が与えられたらできるだけ Maple で解いてしまう．答えをあらかじめ知っておくと間違いを見つけるのが容易．プロットしてみる．

```
> restart;
  f1:=x->4/(1+x^2);
  plot(f1(x),x=0..5);
```

$$f1 := x \mapsto \frac{4}{1+x^2}$$



Maple で解いてみる．

```
>int(f1(x),x=0..1);
```

$$\pi$$

えっと思うかも知れないが,

```
>int(1/(1+x^2),x);
```

$$arctan(x)$$

となるので，納得できるでしょう.

具体的に Maple でコードを示す．先ずは初期設定.

```
>N:=8: x0:=0: xn:=1: Digits:=20:
```

Midpoint rule(中点法)

```
> h:=(xn-x0)/N: S:=0:
  for i from 0 to N-1 do
    xi:=x0+(i+1/2)*h;
    dS:=h*f1(xi);
    S:=S+dS;
  end do:
  evalf(S);
```

3.1428947295916887799

Trapezoidal rule(台形公式)

```
> h:=(xn-x0)/N: S:=f1(x0)/2:
  for i from 1 to N-1 do
    xi:=x0+i*h;
    dS:=f1(xi);
    S:=S+dS;
  end do:
  S:=S+f1(xn)/2:
  evalf(h*S);
```

3.1389884944910890093

Simpson's rule(シンプソンの公式)

```
> M:=N/2: h:=(xn-x0)/(2*M): Seven:=0: Sodd:=0:
  for i from 1 to 2*M-1 by 2 do
    xi:=x0+i*h;
    Sodd:=Sodd+f1(xi);
  end do:
  for i from 2 to 2*M-1 by 2 do
    xi:=x0+i*h;
    Seven:=Seven+f1(xi);
  end do:
  evalf(h*(f1(x0)+4*Sodd+2*Seven+f1(xn))/3);
```

3.1415925024587069144

1.7 課題

1. 補間と近似の違いについて，適切な図を描いて説明せよ．
2. 次の 4 点

```
x y
0 1
1 2
2 3
3 -2
```

   を通る多項式を以下のそれぞれの手法で求めよ．(a) 逆行列, (b) ラグランジュ補間, (c) ニュートンの差分商公式
3. $\tan(5°) = 0.08748866355$, $\tan(10°) = .1763269807$, $\tan(15°) = .2679491924$ の値を用いて，ラグランジュ補間法により，$\tan(17°)$ の値を推定せよ．(2008 年度期末試験)
4. exp(0)=1.0, exp(0.1)=1.1052, exp(0.3)=1.3499 の値を用いて，ラグランジュ補間法により，exp(0.2) の値を推定せよ．(2009 年度期末試験)
5. 次の関数

$$f(x) = \frac{4}{1+x^2}$$

   を $x = 0..1$ で数値積分する．

   (a) N を 2,4,8,…256 ととり，N 個の等間隔な区間にわけて中点法で求めよ．(15)

   (b) 小数点以下 10 桁まで求めた値 3.141592654 との差を dX とする．dX と分割数 N とを両対数プロット (loglogplot) して比較せよ (10)

   (2008 年度期末試験)
6. 次の関数

$$y = \frac{1}{1+x^2}$$

   を $x = 0..1$ で等間隔に $N+1$ 点とり，N 個の区間にわけて数値積分で求める．N を 2, 4, 8, 16, 32, 64, 128, 256 と取ったときの (a) 中点法, (b) 台形公式, (c) シンプソン公式それぞれの収束性を比較せよ．

   ヒント：Maple script にあるそれぞれの数値積分法を関数 (procedure) に直して，for-loop で回せば楽．出来なければ，一つ一つ手で変えても OK. 両対数プロット (loglogplot) すると見やすい．

1.8 解答例

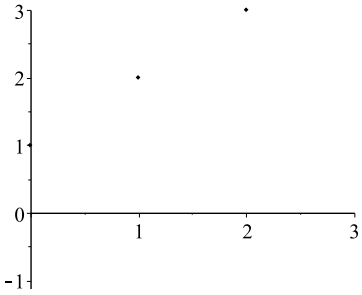
2.
```
> restart;

X:=[0,1,3,2];
Y:=[1,2,-2,3];
with(LinearAlgebra):
list1:=[X,Y];
```

$$X := [0, 1, 3, 2]$$

$$Y := [1, 2, -2, 3]$$

$$list1 := [[0, 1, 3, 2], [1, 2, -2, 3]]$$

```
> with(plots):
l1p:=pointplot(Transpose(Matrix(list1))):
display(l1p);
```




```
> F2:=y0+(x-x0)*f1_01+(x-x0)*(x-x1)*f2_012;
F3:=y0+(x-x0)*f1_01+(x-x0)*(x-x1)*f2_012+(x-x0)*(x-x1)*(x-x2)*f3_0123;
```

$$F2 := y0 + (x - x0)\, f1\_01 + (x - x0)\,(x - x1)\, f2\_012$$

$$F3 := y0 + (x - x0)\, f1\_01 + (x - x0)\,(x - x1)\, f2\_012 + (x - x0)\,(x - x1)\,(x - x2)\, f3\_0123$$

```
> f1_01:=(y1-y0)/(x1-x0);
f1_12:=(y2-y1)/(x2-x1);
f1_23:=(y3-y2)/(x3-x2);
```

$$f1\_01 := \frac{y1 - y0}{x1 - x0}$$

$$f1\_12 := \frac{y2 - y1}{x2 - x1}$$

$$f1\_23 := \frac{y3 - y2}{x3 - x2}$$

```
> f2_012:=(f1_12-f1_01)/(x2-x0);
f2_123:=(f1_23-f1_12)/(x3-x1);
```

$$f2\_012 := \frac{1}{x2 - x0}\left(\frac{y2 - y1}{x2 - x1} - \frac{y1 - y0}{x1 - x0}\right)$$

$$f2\_123 := \frac{1}{x3 - x1}\left(\frac{y3 - y2}{x3 - x2} - \frac{y2 - y1}{x2 - x1}\right)$$

```
> f3_0123:=(f2_123-f2_012)/(x3-x0);
```

$$f3\_0123 := \frac{1}{x3 - x0}\left(\frac{1}{x3 - x1}\left(\frac{y3 - y2}{x3 - x2} - \frac{y2 - y1}{x2 - x1}\right) - \frac{1}{x2 - x0}\left(\frac{y2 - y1}{x2 - x1} - \frac{y1 - y0}{x1 - x0}\right)\right)$$

```
F2;
```

$$y0 + \frac{(x - x0)\,(y1 - y0)}{x1 - x0} + \frac{(x - x0)\,(x - x1)}{x2 - x0}\left(\frac{y2 - y1}{x2 - x1} - \frac{y1 - y0}{x1 - x0}\right)$$

```
> for i from 1 to 4 do
  x||(i-1):=X[i];
  y||(i-1):=Y[i];
end:
F2;
F3;
```

$$1 + x - x\,(x - 1)$$

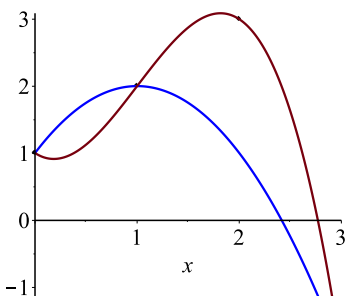
$$1 + x - x\,(x - 1) - x\,(x - 1)\,(x - 3)$$

```
> eq2:=expand(F2);
eq3:=expand(F3);
```

$$eq2 := -x^2 + 2\,x + 1$$

$$eq3 := -x^3 + 3\,x^2 - x + 1$$

```
> with(plots):
l1p:=pointplot(Transpose(Matrix(list1))):
pf2:=plot(eq2,x=0..3,color=blue):
pf3:=plot(eq3,x=0..3):
display(l1p,pf2,pf3);
```



5.
```
> restart;
f1:=unapply(4/(1+x^2),x);
```

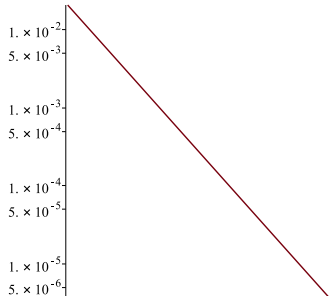
$$x \to \frac{4}{x^2+1}$$

```
> plot(f1(x),x);
```


```
> int(f1(x),x=0..1);
```

$$\pi$$

```
> N:=8: x0:=0: xn:=1: Digits:=20:
> mid:=proc(N)
  global x0,xn;
  local S,i,dS,h,xi;
  h:=(xn-x0)/N:
  S:=0:
  for i from 0 to N-1 do
    xi:=x0+(i+1/2)*h;
    dS:=h*f1(xi);
    S:=S+dS;
  end do:
  evalf(S);
end;

> results:=[];
for i from 1 to 8 do
 results:=[op(results),[2^i,evalf(abs(mid(2^i)-Pi))]];
end:

with(plots):
loglogplot([results]);
```


1. × 10-2
5. × 10-3
1. × 10-3
5. × 10-4
1. × 10-4
5. × 10-5
1. × 10-5
5. × 10-6