Maxima 入門
ver. 1.01.01

逸見 豊

2009 年 6 月 22 日

はじめに

このファイルは、逸見先生が下記にあるように作ったファイルを、つちもとがなかば強引にネットにアップしたものです。著作権はもちろん逸見先生にありますが、間違い等はつちもとまでお知らせください。ただし、本テキストを用いたことによるいかなる事象についてもつちもと、逸見先生、その属する団体はなんの責任も取りません。個人の責任でお使いください。

この小冊子のバージョン 0.10 は，高知大学理学部数理情報科学科数理科学コースにおける講義「計算機数学」のテキストとして用いるために，2007 年に作成されたものである．

その後，講義で使用しながら修正を加え，現在のバージョンは 1.00 である．対象とする Maxima は wxMaxima 0.8.1 である．

目 次

第1章　数の計算

1.1　Maximaによる計算

Maxima を起動すると，大きな窓になにやら英語が表示されます．[1]
まず最初に，

```
1+2
```

と打ち込み，(Shift) を押しながら (Enter) を入力してみたください．画面に

```
(%i1)  1+2;
(%o1)  3
```

と表示されたと思います．

最初の行は入力が `1+2` であったことを表し，2 番目の行はその出力結果である `3` を表示しています．

この様に，Maxima は命令を打ち込んだ後，(Shift) キーを押しながら (Enter) を入力することにより，その結果を表示するようになっています．(Shift) キーを押さずに (Enter) を入力すると，単に改行されるだけです．

画面をよく見ると，入力ラインは `(%i1)` のような`%i` の後に番号がついたラベルが与えられます．一方，出力ラインには `(%o1)` のような`%o` の後に番号がついたラベルが与えられます．このラベルの `i` は input の i で，`o` は output の o を意味します．一般に n 番目の入力には `(%i`n`)` というラベルが振られ，n 番目の出力には `(%o`n`)` というラベルが振られます．

1.2　加減乗除

加減乗除にはそれぞれ $+$，$-$，$*$，$/$ を用います．出力は下のようになります．Maxima は上の例のように電卓のような使い方ができます．例えば $(3 \times (24 - 8) + 4) \div 18$ は

```
(3*(24-8)+4)/18
```

のように入力します．出力は次のようになり，計算結果 $\dfrac{26}{9}$ が表示されます．

[1] 以前のバージョンでは，命令を入力するための入力窓は出力を表示する出力窓と別になっていましたが，現在のバージョンでは分かれていません

```
(%i2)  (3*(24-8)+4)/18;
(%o2)  26/9
```

さて，上の例では計算結果は分数で表示されました．これは計算式の中の数がすべて整数のためです．結果ををを小数で表示するには，この後に次のように入力します．

```
(%i3)  %, numer;
(%o3)  2.888888888888889
```

入力行の%はすぐ手前の出力，ここでは (%o2) の $\frac{26}{9}$ を表し，そのあとのコンマと numer でその値の小数近似を表示します．たとえば，5/2, numer と入力すると，次のような出力を得ます．

```
(%i4)  5/2, numer;
(%o4)  2.5
```

先ほどの計算 $(3 \times (24 - 8) + 4) \div 18$ において，ひとつでも小数が含まれれば，出力は小数になります．たとえば，(%i2) の 3 を 3.0 に変えて計算を実行すると次のようになります．

```
(%i5)  (3.0*(24-8)+4)/18;
(%o5)  2.888888888888889
```

1.3 巾，階乗

巾，階乗などの計算は^, ! を用います．例えば 3^{24}, 10! は

```
(%i6)  3^24; 10!;
(%o6)  282429536481
(%o7)  3628800
```

のようになります.

上の計算では，3^{24} と 10! が 1 行で入力されています．このように，複数の計算を 1 行で入力するときは，セミコロン ; を用いて，式を区切ってやります．出力 (%o6) が 3^{24} の計算結果で，(%o7) が 10! の計算結果になります．

この後に命令を入力すると (%i8) という番号がつき，(%i7) は一見スキップされたように見えます．実は，(%i6) で二つの計算を実行していますが，実際には (%i6) での入力は 3^{24} のみで，10! は (%i7) の入力と考えます．そのため，(%o7) が 10! の計算結果なのです．

1.4 平方根

平方根がついた数の計算もできます．$(1+\sqrt{2})^5$ を計算しましょう．$\sqrt{2}$ は sqrt(2) とします．

```
(%i8)  (1 + sqrt(2))^5;
(%o8)  (√2+1)^5
(%i9)  expand (%);
(%o9)  29*√2+41
(%i10)  %, numer;
(%o10)  82.01219330881976
```

入力行 (%i9) の expand(%) について説明します．%は前の出力 $(\sqrt{2}+1)^5$ を表します．そして expand という命令は，その式を展開しなさいというものです．展開した結果は $29\sqrt{2}+41$ であることが分かります．さらに次の行では，その値を小数で近似しています．ちなみに，窓の下に Expand と書かれたボタンがありますが，これを押すことで自動的に expand(%) が入力されます．

入力する際に%を用いると，一つ前の出力結果を参照することができますが，Maxima は一つ前ばかりでなく，行についているラベルを用いてその行の出力結果を参照することができます．たとえば (%o4) の 2.5 を 3 倍してみます．

```
(%i11)  (%o4)*3;
(%o11)  7.5
```

一度入力した命令も参照できます．たとえば，(%i6) の 10! を再実行してみます．ただし，この場合は，先ほど説明したように，(%i6) の 2 番目の入力なので，実際には (%i7) として処理します．

```
(%i12)  (%i7);
(%o12)  3628800
```

1.5 文字列への代入

Maxima では任意の文字列に数値を代入することができます．たとえば文字 x に 12 を代入するときは x:12 のようにコロン : を用います．

```
(%i13)  x:12$ y1:2$ y2:3$
(%i16)  x*y1; x*y2;
(%o16)  24
(%o17)  36
(%i18)  kill(x,y1,y2);
(%o18)  done
(%i19)  x*y1; x*y2;
(%o19)  xy1
(%o20)  xy2
```

上の例では，(%i13) で x，y1，y2 にそれぞれ 12, 2, 3 を代入し，(%i16) で，x の y1 倍と y2 倍を計算しています．その後，(%i18) の kill(x,y1,y2) で x，y1，y2 の値をクリアしています．(%i19) で同じ計算をしていますが，(\o19), (\o20) の出力では文字式が表示されるだけになっています．

ちなみに，(%i13) で$が用いられています．通常は，これはセミコロン ; が用いられるところですが，そのようにすると 12, 2, 3 がその後に表示され，多少うっとおしいです．セミコロンの代わりに$を用いると，出力結果が表示されません．

1.6 商と余り

割り算では / 以外に，商と余りを計算する方法があります．例えば $54 \div 5 = 10 \cdots 4$ の計算で，商の 10 と余りの 5 は次のように計算します．

```
(%i21)  divide(54,5);
(%o21)  [10, 4]
```

(%o21) の [10,4] は最初の 10 が商で 2 番目の 4 が余りを表します．これから商と余りを取り出すには次のようにします．

```
(%i22)  %[1];
(%o22)  10
(%i23)  (%o21)[2];
(%o23)  4
```

一般に，[a,b,...,c] のような列の第 k 番目を取り出すには [a,b,...,c][k] のようにします．上の例では，divide(54,5)[1], divide(54,5)[2] のようにしても同じ結果を得ます．

1.7 最大公約数，最小公倍数

与えられた 2 整数 a, b に対して，それらの最大公約数は gcd(a,b) で求まります．

```
(%i24)  gcd(360,945);
(%o24)  45
```

今度は 3 つの数 360,945,2310 の最大公約数を求めてみましょう

```
(%i25)  gcd(360,945,2310);
なにやら文字列
```

エラーになったようです．3 つ以上の数の最大公約数は，gcd(a,b,c,...) のようにしては求まりません．次のようにします．

```
(%i26)  gcd(gcd(360,945),2310);
(%o26)  15
```

一方，最小公倍数は lcm(a,b) で求まりますが，この関数を用いるためには，最初に functs というライブラリをロードしておかなくてはなりません．

```
(%i27)  load("functs");
(%o27)   なにやら文字列
(%i28)  lcm(20,32,86);
(%o28)  6880
```

lcm は 3 つ以上の数でも問題なかったようです.

1.8 Euclid の互助法

ここでは Euclid の互助法を用いて gcd を用いずに与えられた 2 つの正の整数の最大公約数を求めてみましょう.

補題 1.8.1. a, b は正の整数で，$a \geq b$ を満たすものとする．いま a は b の倍数でないとし，a を b で割った余りが r $(1 \leq r < b)$ であるなら，a と b の最大公約数と，b と r の最大公約数は一致する.

上の補題を用いて，与えられた正の整数 a と b の最大公約数が次のようにして求まります.

1. $a \geq b$ とし，a を b で割った余りを r_1 とする.
2. もし $r_1 = 0$ であれば，a は b で割り切れることになり，最大公約数は b になる.
3. $r_1 \neq 0$ のとき，上の補題より a と b の最大公約数と b と r_1 の最大公約数は一致するから，b と r_1 の最大公約数を求めればよい.
4. 次に上の手順を，a, b の代わりに，b, r_1 を用いて行う．すなわち，b が r_1 で割り切れれば r_1 が最大公約数になり，割り切れなければ，b を r_1 で割った余り，それを r_2 とする，を考えると，r_1 と r_2 の最大公約数が b と r_1 の最大公約数，すなわち a と b の最大公約数と一致する.
5. これを繰り返していくと，$a \geq b > r_1 > r_2 > r_3 > \cdots > r_j = 0$ という列が得られる．ただし，r_t は r_{t-2} を r_{t-1} で割った余りである．そこで r_{j-1} が最初の a, b の最大公約数になる.

1155 と 1008 の最大公約数を求めてみましょう.

```
(%i29)  a:1155$ b:1008$
(%i30)  r1:divide(a,b)[2]
(%o30)  147
(%i31)  r2:divide(b,r1)[2]
(%o31)  126
(%i32)  r3:divide(r1,r2)[2]
(%o32)  21
(%i33)  r4:divide(r2,r3)[2]
(%o33)  0
```

よって最大公約数は，r3 の 21 になることがわかります.

演習問題

1. 次の値を簡単にせよ．また，それぞれを小数で近似せよ．

   (1) $\dfrac{2+3^{12}-5!}{7}$　　(2) $\left(1+\sqrt{3}\right)^5$

2. 変数 x, y, z に $x=3$, $y=12$, $z=-5$ を代入し，次の値を求めよ．

   (1) x^2-3yz　　(2) $(x-2y)(5z+xy)$

3. 上で代入した変数 x, y, z の値を削除し，新たに $x=\sqrt{3}$, $y=1.2$, $z=7$ を代入し，上の (1), (2) の値を求めよ．

4. `gcd`, `lcm` を用いずに 8775, 4590 の最大公約数および最小公倍数を求めよ．（ヒント：a, b の最大公約数を d，最小公倍数を m とすると，$ab=dm$ が成り立つ．）

5. 上の問題と同様に，`gcd`, `lcm` を用いず 8775, 4590, 1970 の最大公約数および最小公倍数を求めよ．

第2章　式の計算

2.1　文字式の計算

Maxima は文字式の計算が可能です．たとえば $(x^2-3x+1)-(2x^2+4x-5)$ の計算は次のようにします．

```
(%i1)  (x^2-3*x+1)-(2*x^2+4*x-5);
(%o1)  −x^2 − 7x + 6
```

次の例は，$A = x^2 + 3y^2 - xy + 5x$, $B = -x + 3y^2 + 4xy$ と代入して計算しています．

```
(%i2)  A:x^2+3*y^2-x*y+5*x$ B:-x+3*y^2+4*x*y$
(%i4)  2*A-B;
(%o4)  2(3y^2 − xy + x^2 + 5x) − 3y^2 − 4xy + x
(%i5)  expand(%);
(%o5)  3y^2 − 6xy + 2x^2 + 11x
```

多項式を指定した文字で整理する場合は `ratsimp` を用いて次のようにします．

```
(%i6)  pol:x^2*y-3*x+5*y-x^2+1$ ratsimp(pol,x)
(%o7)  5y + x^2(y − 1) − 3x + 1
(%i8)  ratsimp(pol,y)
(%o8)  (x^2 + 5)y − x^2 − 3x + 1
```

`(%o7)` は x について整理しているのですが，多少違和感がある出力になっています．このように，必ずしも期待する結果にならないことがあります．

多項式の変数に値を代入するにはいくつかの方法があります．`pol` の x に 2 を代入してみます．

```
(%i9)  pol,x=2
(%o9)  9y − 9
(%i10)  ev(pol,x=2)
(%o10)  9y − 9
(%i11)  subst(2,x,pol);
(%o11)  9y − 9
```

次の例では，最初に多項式 pol の x に y を代入しています．次に x に a+b を，y に a-b を代入し，その後に展開してから文字 b について整理しています．

```
(%i12)  pol,x=y;
(%o12)  y^3 − y^2 + 2y + 1
(%i13)  pol,[x=a+b, y=a-b];
(%o13)  (a − b)(b + a)^2 − (b + a)^2 − 3(b + a) + 5(a − b) + 1
(%i14)  ratsimp(expand(%),b);
(%o14)  −b^3 + (−a − 1)b^2 + (a^2 − 2a − 8)b + a^3 − a^2 + 2a + 1
```

2.2 展開と因数分解

式の展開は expand を用いました．逆に与えられた式の因数分解をするには factor を用います．

```
(%i15)  expand((x^3-3*x^2+2)*(4*x+5));
(%o15)  4x^4 − 7x^3 − 15x^2 + 8x + 10
(%i16)  factor(%);
(%o16)  (x − 1)(4x + 5)(x^2 − 2x − 2)
(%i17)  (3*x^3+5*x^2-11*x+3)/(3*x-1);
                          3x^3 + 5x^2 − 11x + 3
(%o17)                    ---------------------
                                 3x − 1
(%i18)  factor(%);
(%o18)  (x − 1)(x + 3)
```

最後の例にあるように，分数に factor を適応すると，因数分解をして約分してくれます．

演習問題

1. $A = x^2 - 2x + 3$, $B = -3x^2 + 1$, $C = x^2 - 2x + 5$ のとき，次の式を計算せよ．

   (1) $(B - A) + (C - A)$　　(2) $2C - (B + C - 2A)$

2. 次の式を展開せよ．

   (1) $(x - y)(x + y)(x^2 + y^2)$ (2) $(x^2 - 1)(x^2 + x + 1)(x^2 - x + 1)$

3. 次の式を因数分解せよ．

   (1) $3x^2y - 7xy - 6y$　　(2) $(x - y)(x - y + 5) + 6$

4. $x = 2 + \sqrt{3}$, $y = 2 - \sqrt{3}$ のとき，次の式の値を求めよ．

   (1) $x^2 + y^2$　　(2) $\dfrac{y^2}{x} + \dfrac{x^2}{y}$

5. $a + b + c = 0$, $abc \neq 0$ のとき，次の等式を証明せよ．

   (1) $\dfrac{b + c}{a} + \dfrac{c + a}{b} + \dfrac{a + b}{c} = -3$

   (2) $\dfrac{b^2 - c^2}{a} + \dfrac{c^2 - a^2}{b} + \dfrac{a^2 - b^2}{c} = 0$

   （ヒント : 与えられた条件を用いて左辺を変形すると右辺になればよい.）

第3章　方程式

3.1　代数方程式の解法

Maxima を用いて方程式を解くには `solve` を用います．たとえば $x^2+x-5=0$ を解くのには次のようにします．

```
(%i1)  solve(x^2+x-5=0);
(%o1)  [x = -(sqrt(21)+1)/2, x = (sqrt(21)-1)/2]
(%i2)  ans:%$ ans[2];
(%o3)  x = (sqrt(21)-1)/2
(%i4)  x^2+x-5,%;
                    2
(%o4)  (sqrt(21) - 1)  / 4 + (sqrt(21) - 1)/2 - 5
(%i5)  expand(%);
(%o5)  0
```

最初の出力は，$x^2+x-5=0$ の解が $-\dfrac{\sqrt{21}+1}{2}$ と $\dfrac{\sqrt{21}-1}{2}$ の 2 つであることを意味しています．

2 番目の入力の `ans:%` は，前の出力を `ans` という文字に代入しているということです．つまり `ans` $= \left[x=-\dfrac{\sqrt{21}+1}{2}, x=\dfrac{\sqrt{21}-1}{2}\right]$ という意味です．そしてその後の `ans[2]` で，`ans` の 2 番目，すなわち $x=\dfrac{\sqrt{21}-1}{2}$ を取り出しています．

`(%i4)` は `ans[2]` である $\dfrac{\sqrt{21}-1}{2}$ が実際に方程式の解になっているかどうかを，元の式の x にその値を代入して調べています．そして，最後に上の式を展開し，結果の 0 が得られています．つまり $\dfrac{\sqrt{21}-1}{2}$ が実際に解になっていることがわかります．

こんどは $x^4=1$ を解きます．

```
(%i6)  solve(x^4=1);
(%o6)  [ x = %i, x = -1, x = -%i, x = 1 ]
```

上の出力で%i は虚数単位 $\sqrt{-1}$ を表しています．つまり，方程式には複素数の範囲で 4 つの解 $[i, -1, -i, 1]$ があることがわかります．

方程式を実数の範囲で解くには realroots を用います．

```
(%i7)  realroots(x^4=1);
(%o7)  [ x = -1, x = 1 ]
```

連立方程式も solve を用いて解くことができます．たとえば未知数を x と y とする連立方程式

$$\begin{cases} x^2 + y^2 = 1 \\ x + y = 1 \end{cases}$$

は，次のようにして解きます．

```
(%i8)  solve([x^2+y^2=1, x+y=1]);
(%o8)  [ [ y = 1, x = 0] , [ y = 0, x = 1] ]
(%i9)  a:%$ a[1];
(%o10)  [ y = 1, x = 0 ]
(%i11)  a[1][2];
(%o11)  x = 0
```

最初の出力は，解が $y = 1,\ x = 0$ と $y = 0,\ x = 1$ の 2 組である事を意味します．次に文字 a に最初の出力（解の集合）を代入し，a[1] で 1 番目の解を表示させています．最後の a[1][2] は，1 番目の解の 2 番目，つまり $x = 0$ を意味します．

演習問題

1. solve を用いて，2 次方程式の解の公式を求めよ．同様に，$n = 3, 4, 5$ について，n 次方程式の解の公式が求まるか試してみよ．

第4章　数列と級数

4.1　数列と数列の和

Maximaでは数列はリストとして与えます．たとえば$2,3,5,7,11,13,17,19,23,29$のような数列aは次のようにして与えます．

```
(%i1)  a: [2,3,5,7,11,13,17,19,23,29];
(%o1)  [2,3,5,7,11,13,17,19,23,29]
```

aの第4項と第9項を表示し，さらに初項から第10項までの和を求めます．

```
(%i2)  a[4]; a[9];
(%o2)  7
(%o3)  23
(%i4)  sum(a[i],i,1,10);
(%o4)  129
(%i5)  kill(a)$
```

最後の`kill(a)`で，aに代入した値（ここでは`[2,3,5,7,11,13,17,19,23,29]`）を削除しています．

数列の第n項をnの関数として定義できる場合もあります．関数の一般的な話は後で説明しますが，たとえば，第n項が$2n-5$である数列は，関数$b(x)=2x-5$のxに自然数を代入したものと考えることができます．

```
(%i6)  b(n):=2*n-5;
(%o6)  b(n) := 2n - 5
(%i7)  makelist(b(n),n,1,5);
(%o7)  [-3,-1,1,3,5]
(%i8)  sum(b(i),i,1,n);
(%o8)  sum(2i - 5, i, 1, n)
(%i9)  %,simpsum;
(%o9)  n^2 - 4n
```

(%i7) の makelist はリストを作るコマンドです．初項から第 5 項までのリストが表示されました．

また，初項から第 n 項までの和 $\sum_{i=1}^{n} b(i)$ を計算していますが，(%i9) でそれを単純な形で表示しています．

この simpsum というのは，maxima に組み込まれている変数の一つで，通常は false という値が入っています．これは，次のようにすると確かめることができます．

```
(%i10)  simpsum;
(%o10)  false
```

(%i9) は，simpsum の値を一時的に true にして，和を表示させていることになります．

常に，単純化した形で和を求めたいときは，次のようにします．

```
(%i11)  simpsum:true;
(%o11)  true
(%i12)  sum(b(i),i,1,n);
(%o12)  n^2 - 4n
```

4.2 無限級数

無限級数の和も求めることができます．たとえば $\sum_{i=1}^{\infty} \frac{1}{i^2}$ は

```
(%i13)  sum(1/i^2, i,1,inf);
(%o13)  %pi^2/6
(%i14)  %, numer:
(%o14)  1.644934066848226
```

ここで，inf は ∞ を，%pi は円周率 π を表します．また，すでに simpsum の値を true にしてあるので，和は単純な形で表されています．

4.3 数列の極限

今度は，極限 $\lim_{n\to\infty} (1+1/n)^n$ を計算してみましょう．

```
(%i15)  limit((1+1/n)^n,n,inf);
(%o15)  %e
(%i16)  %, numer;
(%o16)  %e
(%i17)  float(%);
(%o17)  2.718281828459045
```

ここで%e はネピアの数（自然対数の底）です．(%i16) で小数近似を試みましたが失敗したようです．それで，(%i17) で float という関数を用いてみたら，今度はうまくいきました．

次に，極限 $\lim_{n\to\infty} a^n$ を計算します．この極限は a の値により違ってきます．すなわち

$$\lim_{n\to\infty} a^n = \begin{cases} \infty & a > 1 \\ 1 & a = 1 \\ 0 & -1 < a < 1 \\ \text{不定} & a = -1 \\ \text{複素無限大} & a < -1 \end{cases}$$

となります．最後の複素無限大の意味は，複素数における無限大で，細かい話は避けますが，ここでは単に絶対値が無限大に発散すると考えていいと思います．

最初に $a > 1$ の仮定の下で計算してみます．仮定には assume を用います．

```
(%i18)  assume(a>1);
(%o18)  [a > 1]
(%i19)  limit(a^n,n,inf);
(%o19)  ∞
```

(%i18) では $a > 1$ という仮定を a に与えています．

次に $a = 1$ のときは自明なので，$-1 < a < 1$ で計算します．そのためには，先ほどの仮定 $a > 1$ を削除する必要があります．仮定の削除は forget を用います．

```
(%i20)  forget(a>1)
(%o20)  [a > 1]
(%i21)  facts(a);
(%o21)  [ ]
(%i22)  assume(a<1,a>-1);
(%o22)  [a < 1, a > -1]
(%i23)  limit(a^n,n,inf);
Is |a| - 1 positive, negative, or zero? n;
Is a positive, negative, or zero? p;
(%o23)  0
```

facts(a) は，現在 a に与えている仮定を表示させるものです．前の行で forget をして，仮定を削除してしまっているので，ここではブランクになっています．

(%i23) の後に「Is $|a|-1$ positive, negative, or zero?」という質問が表示されます．これは $|a|-1$ が正 (positive) なのか負 (negative) なのか，それとも 0 (zero) なのかという問い合わせです．$-1 < a < 1$ を仮定しているわけですから，当然負に決まっているんですが，Maxima はその辺がちょっとお馬鹿なようです．そこで，負になるので「n」と答えます．正であれば「p」で，0 であれば当然「0」です．次に a についても聞いてきますが，ここではとりあえず正の場合を考え「p」と答えました．ちなみに，a が負の場合でも結果は同じになります．ここでは省略します．

最後に $a = -1$ の場合と $a < -1$ の場合を考えて見ます．

```
(%i24)  limit((-1)^n,n,inf);
(%o24)  ind
(%i25)  forget(a<1,a>-1)
(%o25)  [a < 1, a > -1]
(%i26)  assume(a<-1);
(%o26)  [a < -1]
(%i27)  limit(a^n,n,inf);
(%o27)  infinity
```

(%o24) の ind は不定を表しています．そして，最後の (%o27) の $infinity$ が複素無限大で，通常の正の実無限大 inf と区別して下さい．

演習問題

1. 次の和を求めよ.（結果は因数分解すること）

   (1) $\sum_{i=1}^{n} i$　　(2) $\sum_{i=1}^{n} i^2$　　(3) $\sum_{i=1}^{n} i^{10}$　　(4) $\sum_{i=1}^{n} 2i - 3i^2 + 4i^3$

2. 初項から第 n 項までの和が $-n^2 + 24n$ で与えられる数列の一般項を求めよ.

3. すべての項が 1 である数列を $(a1_n)_{n=1,2,\ldots}$ とおく.（つまり，任意の n に対して $a1_n = 1$）次に数列 $\{ak_n\}$, $(k \geq 1)$ を帰納的に

   $$a2_n = \sum_{i=1}^{n} a1_i, \quad a3_n = \sum_{i=1}^{n} a2_i, \quad a4_n = \sum_{i=1}^{n} a3_i, \quad \ldots$$

   のように定めたとき，$\{ak_n\}$ の一般項を $k = 1, 2, 3, 4, 5, 6$ について求めよ．また，その結果から，一般に ak_n がどのようになるかを推測せよ．
   ヒント:ak_n は Maxima では `a[k](n)` とする．すなわち，$a1_n$ は `a[1](n):=1` と定める.

4. 第 n 項が次で与えられる数列の極限を求めよ.

   (1) $\dfrac{1^2 + 3^2 + \cdots + (2n-1)^2}{n^3}$　　(2) $n^{1/n}$　　(3) $\left(1 - \dfrac{1}{n}\right)^{-n}$

5. $a > 0$ のとき次の極限値を求めよ.

   (1) $\lim_{n\to\infty} a^n$　　(2) $\lim_{n\to\infty} a^{1/n}$

6. 初項が 1 で正の公比を持つ等比数列 (f_n) で漸化式 $f_{n+2} = f_n + f_{n+1}$ $(n \geq 1)$ を満たすものを求めよ.

7. 次の等式が成立することを確かめよ.

   $$\left(1 - \frac{1}{2^2}\right)\left(1 - \frac{1}{3^2}\right)\cdots\left(1 - \frac{1}{(n+1)^2}\right) = \frac{n+2}{2(n+1)}$$

   ヒント：数学的帰納法を用いて確かめる.

第5章　関数

5.1　関数の定義

関数の定義はすでに数列のところで説明しましたが，実は2通りの定義の方法があります．例えば $f(x)=x^2$ で定義される関数 f は次のように与えることができます．

```
(%i1)  f(x):=x^2;
(%o1)  f(x) := x^2
(%i2)  define(f(x),x^2);
(%o2)  f(x) := x^2
```

ただし，この2つは完全に同じというわけではなく，微妙な違いがるため，状況によって使い分ける必要があります．特に問題がなければ，前者の方が簡単だし，記憶しやすいと思います．

上の関数の x にさまざまな値を代入してみます．

```
(%i3)  f(-1); f(0); f(1/2); f(a+b); f(f(x))
(%o3)  1
(%o4)  0
(%o5)  1/4
(%o6)  (b+a)^2
(%o7)  x^4
```

最後の `f(f(x))` は合成関数 $f\circ f(x)$ をあらわす事になります．

5.2　組み込み関数

Maxima には，三角関数や指数関数のような基本的な関数が組み込まれています．三角関数およびその逆関数については次のようなものが組み込まれ

ています.

$\sin x$	`sin(x)`	$\cos x$	`cos(x)`	$\tan x$	`tan(x)`
$\csc x$	`csc(x)`	$\sec x$	`sec(x)`	$\cot x$	`cot(x)`
$\sin^{-1} x$	`asin(x)`	$\cos^{-1} x$	`acos(x)`	$\tan^{-1} x$	`atan(x)`
$\csc^{-1} x$	`acsc(x)`	$\sec^{-1} x$	`asec(x)`	$\cot^{-1} x$	`acot(x)`

また，指数関数 e^x は `exp(x)` または`%e^x` で，対数関数 $\log x$ は `log(x)` で表されます.

$\sin\pi$, $\cos\frac{\pi}{4}$, $\tan\frac{\pi}{3}$, $\arcsin 1$, e^0, $\log 1$ の値を計算してみます.

```
(%i8)  sin(%pi); cos(%pi/4); tan(%pi/3); asin(1);
(%o8)  0
(%o9)  1/sqrt(2)
(%o10) sqrt(3)
(%o11) %pi/2
(%i12) exp(0); log(1);
(%o12) 1
(%o13) 0
```

加法定理を用いて，式に含まれる三角関数をできるだけやさしい形に展開するには `trigexpand` を用います．その逆に，与えられた式を，ただ一つの sin または cos のみを含む式の和として表すには `trigreduce` を用います.

```
(%i14) sin(x+y)*cos(x)^3;
(%o14) cos(x)^3*sin(y+x)
(%i15) trigexpand(%);
(%o15) cos(x)^3*(cos(x)*sin(y)+sin(x))*cos(y)
(%i16) trigreduce(%);
(%o16) sin(y+4*x)+sin(y-2*x)/8+3*sin(y+2*x)+3*sin(y)/8
```

5.3 関数の極限

関数の極限は，数列の極限と同様に求めることができます．たとえば

$$\lim_{x\to 0}\frac{\sin x}{x}, \qquad \lim_{x\to\infty}(\sqrt{x^2+x+1}-x)$$

は次のように計算します.

```
(%i17)  limit(sin(x)/x,x,0);
(%o17)  1
(%i18)  limit(sqrt(x^2+x+1)-x, x, inf);
(%o18)  1/2
```

左極限や右極限も求めることができます．$f(x) = \dfrac{x-1}{|x-1|}$ を考えて見ます．この関数は

$$f(x) = \begin{cases} -1 & x < 1 \\ 1 & x > 1 \end{cases}$$

であり，$x = 1$ では定義されません．さらに，$\lim_{x\to 1} f(x)$ は存在せず，$\lim_{x\to 1+0} f(x) = 1$, $\lim_{x\to 1-0} f(x) = -1$ になります．実際に Maxima で計算してみると次のようになります．

```
(%i19)  f(x):=(x-1)/abs(x-1);
(%o19)  f(x) := (x-1)/|x-1|
(%i20)  limit(f(x),x,1);
(%o20)  und
(%i21)  limit(f(x),x,1,plus); limit(f(x),x,1,minus);
(%o21)  1
(%o22)  -1
```

最初の行で，$f(x)$ を定義しています．絶対値 $|x-1|$ は `abs(x-1)` とします．その後で，$\lim_{x\to 1} f(x)$ を調べていますが，不定の *und* が出力されています．さらに，$\lim_{x\to 1+0} f(x)$ と $\lim_{x\to 1-0} f(x)$ を調べたところ，きちんと 1 と -1 という答えが返ってきました．

第6章　微分

6.1　常微分

微分には diff を用います．たとえば $\sin x$, e^x, $\log x$ の微分は次のようにします.

```
(%i1)  diff(sin(x),x); diff(exp(x),x); diff(log(x),x);
(%o1)  cos(x)
(%o2)  %e^x
(%o3)  1/x
```

自分で定義した関数も微分できます.

```
(%i4)  f(x):=x^2+3*x-5$
(%i5)  diff(f(x),x);
(%o5)  2x+3
```

以前に関数の定義が 2 通りあり，微妙に違っていると書きました．その違いは次のような場面であらわれてきます.

```
(%i6)  g(x):=diff(f(x),x)$
(%i7)  define(h(x),diff(f(x),x))$
(%i8)  g(x);
(%o8)  2x+3
(%i9)  h(x);
(%o9)  2x+3
(%i10)  g(2);
エラーメッセージ
(%i11)  h(2);
(%o11)  7
```

上のように g(x) も h(x) も一見同じであるように見えますが，x に値を代入したときに違いが表れます.

関数の積や合成関数の微分もできます.

```
(%i12)  diff(sin(x)*cos(x),x);
(%o12)  cos(x)^2 - sin(x)^2
(%i13)  diff(sin(log(x)),x);
(%o13)  cos(log(x))/x
```

高階の微分も計算できます．たとえば $\sin(x^2)$ の 3 階微分は次のようにします.

```
(%i14)  diff(sin(x^3),x,3);
(%o14)  -54*x^3*sin(x^3)-27*x^6*cos(x^3)+6*cos(x^3)
```

6.2 偏微分

偏微分もほとんど同様です.

```
(%i15)  f(x,y,z):=sin(x)*cos(y)*z$
(%i16)  diff(f(x,y,z),x); diff(f(x,y,z),y);
diff(f(x,y,z),z);
(%o16)  cos(x)*cos(y)*z
(%o17)  -sin(x)*sin(y)*z
(%o18)  sin(x)*cos(y)
(%i19)  diff(diff(f(x,y,z),x),y);
(%o19)  -cos(x)*sin(y)*z
```

最後は，x で偏微分した後 y で偏微分しています．つまり $\dfrac{\partial^2 f(x,y,z)}{\partial y \partial x}$ です.

演習問題

1. 関数 $f(x) = xe^{-x^2}$ $(-5 \leq x \leq 5)$ の極値を求めよ.

第7章　積分

積分には integrate を用います．不定積分 $\int x^2 \sin x \, dx$，定積分 $\int_0^{\pi/2} \sin x \, dx$ および広義積分 $\int_0^{\infty} e^{-x^2} \, dx$ は次のように計算します．

```
(%i1)  integrate(x^2*sin(x)^2,x);
```

$$(\%o1) \quad -\frac{(6x^2-3)\sin(2x)+6x\cos(2x)-4x^3}{24}$$

```
(%i2)  integrate(sin(x), x, 0, %pi/2);
(%o2)  1
(%i3)  integrate(exp(-x^2), x, 0, inf);
```

$$(\%o3) \quad \frac{\sqrt{\%pi}}{2}$$

第8章　関数のグラフ

8.1　1変数関数のグラフ

1変数の関数のグラフは wxplot2d または plot2d を用いて描くことができます．wxplot2d を用いると，グラフがインラインで描かれます．一方，plot2d を用いると，デフォルトでは別プログラムの gnuplot が立ち上がり，別の窓にグラフを描いてくれます．gnuplot 以外にも openmath を用いてグラフを表示することもできます．その場合は [plot_format, openmath] というオプションを付けることになります．

たとえば $y = \sin x$ のグラフを $-2\pi \leq x \leq 2\pi$ の範囲で描くには，次の3種類の方法があります．

```
wxplot2d(sin(x), [x,-2*%pi,2*%pi]);
plot2d(sin(x), [x,-2*%pi,2*%pi]);
plot2d(sin(x), [x,-2*%pi,2*%pi], [plot_format, openmath]);
```

openmath で表示するとき，いちいち{[plot_format, openmath] というオプションを付けるのが面倒だと思うなら，

```
set_plot_option([plot_format, openmath]);
```

としておくと，それ以後はオプションを指定しなくても openmath で表示されます．gnuplot に戻す場合は，set_plot_option([plot_format, gnuplot] とします．

ちなみに，3D グラフィックスを表示する場合は openmath が圧倒的にきれいですが，2D グラフィックスではさほどの違いはありません．

同じことなので，以下では，gnuplot による方法を用いて記述します．

$y = \sin x$ のグラフを表示したとき，見慣れたサインカーブとは異なっていると思われたかもしれません．それは x 軸と y 軸のスケールが異なっているためです．wxplot2d や gnuplot による plot2d では [gnuplot_preamble, "set size ratio -1"] というオプションを付けてやると x 軸と y 軸のスケールを同じに表示してくれます．

```
(%i1)  plot2d(sin(x), [x,-2*%pi,2*%pi], [gnuplot_preamble,
"set size ratio -1"]);
```

入力窓の下にある Plot 2D.. というボタンをクリックして関数を描くこともできます．ボタンをクリックすると Plot 2D 窓が開きますので，Expression(s): に `sin(x)` と入力し，2 つある Variable:の上のほうが `x` になっていることを確認し，From:に`-2*%pi` を To:に `2*%pi` を入力し OK をクリックすると同じグラフが表示されます．この窓を用いると，Format:で `inline`, `gnuplot`, `openmath` を選ぶことができ，Options:に `[gnuplot_preamble, "set size ratio -1"]` のようにオプションを指定することができます．このようにしておくと，一度指定したオプションを記憶しておいてくれるので，何度も利用したいときには便利です．

Ticks:の値を変えることにより，`nticks` を変更できます．グラフによっては，ぎざぎざが大きくきれいに表示されないときがありますが，それは描画するために用いられる点の数が少なすぎるためです．そこで，`plot2d` のオプションで `nticks` の値を大きく指定し，描画の点を増やすときれいになります．ちなみに，`nticks` のディフルト値は 10 です．

Plot 2D 窓でなく，インラインで，`nticks` の値を 20 にするには，たとえば

```
plot2d(sin(x), [x,-2*\%pi,2*\%pi], [nticks, 20]);
```

のようにします．

複数の関数を一緒に表示してみます．たとえば $y = \sin x$ と $y = \cos x$ のグラフを $-2\pi \leq x \leq 2\pi$ の範囲で描くには，

```
(%i2)  plot2d([sin(x),cos(x)], [x,-2*%pi,2*%pi]);
```

のように入力します．先ほどの Plot 2D 窓を用いる場合は，Expression(s) に `sin(x),cos(x)` と 2 つ並べて入力します．

8.2 2 変数関数のグラフ

2 変数の関数を描く場合は `wxplot3d` または `plot3d` を使います．1 変数のときと同様に `wxplot3d` を用いると，グラフがインラインで描かれ，`plot3d` を用いると別の窓に描かれます．また，別窓で描く場合，`gnuplot` や `openmath` を用いることができることも同様です．3D グラフィックスでは，`plot3d` を用いると，表示されたグラフを回転させることができます．しかし `wxplot3d` を用いるとインラインで表示されるためグラフを回転させることができません．そのため，`plot3d` の方がベターといえます．

たとえば，$z = \sin(x^2 + y^2)$ を $0 \leq x \leq \pi/2$, $0 \leq y \leq \pi/2$ の範囲で描くには次のようにします．

```
(%i3)  plot3d(sin(x^2+y^2), [x,0,%pi/2], [y,0,%pi/2]);
```

また，1 変数のときのように，入力窓の下にある Plot 3D.. をクリックしても同様のことができます．ただし，描画する点の個数を指定するには，先ほどの Ticks:ではなく Grid:の値を変更します．

8.3 応用

Maxima を用いて高校数学の問題を解いてみましょう．

例題. $y = x^2$ と $y = 2x + 5$ で囲まれる部分の面積を求めよ．

まず $f(x) = x^2$, $g(x) = 2x + 5$ とおき，この 2 つの関数のグラフの交点の座標を求めます．

```
(%i4)  f(x):=x^2$ g(x):=2*x+5$
(%i6)  solve(g(x)-f(x)=0,x);
(%o6)  [x = 1 - sqrt(6), x = sqrt(6) + 1]
```

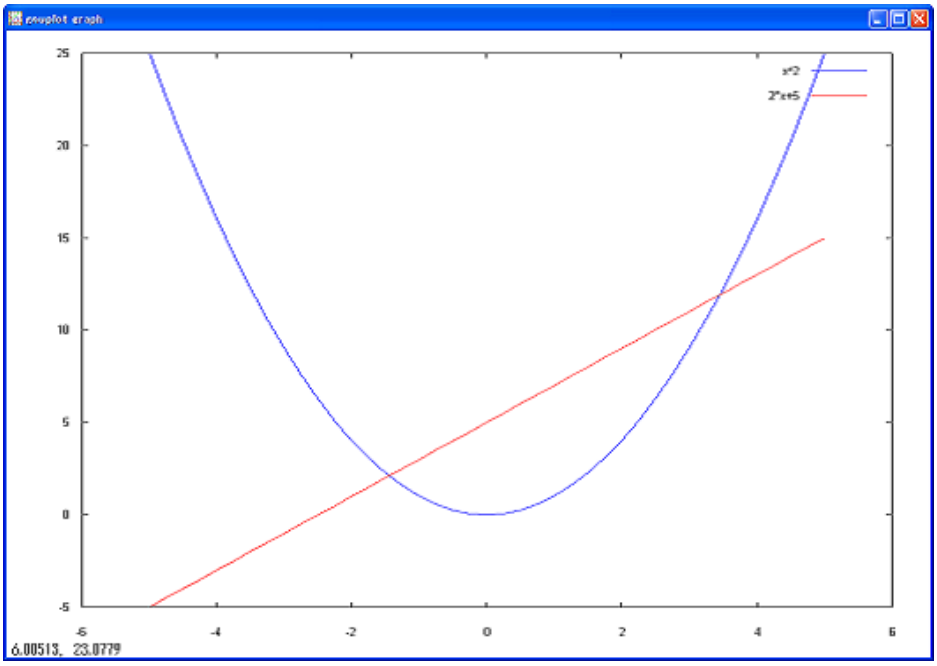
交点の x 座標は $1-\sqrt{6}$ と $\sqrt{6}+1$ であることが分かりましたが，このままでは扱いにくいので，上のリストから x=という部分を削除して，$[1-\sqrt{6}, \sqrt{6}+1]$ というリストを作り kai という変数に代入することにします．

一般に式 a=b から右辺 b を取り出すのは rhs(a=b) とします．また，左辺は lhs(a=b) です．しかし，この場合は rhs(%) のようにしてもうまくいきません．それは出力 (%o6) がリストであるためで，このような場合は map という命令を用いて次のようにします．

```
(%i7)  kai:map(rhs,%);
(%o7)  [1 - sqrt(6), sqrt(6) + 1]
```

そこで，$f(x)$ と $g(x)$ のグラフを同じ平面上に描いてみます．x の範囲は交点の，$x = 1 - \sqrt{6}$ と $x = \sqrt{6} + 1$ が含まれるように選びます．とりあえず $-5 \leq x \leq 5$ の範囲にして見ます．

```
(%i8)  plot2d([f(x),g(x)], [x,-5,5]);
```



図を見ると，2 つの曲線で囲まれる部分では $g(x) \geq f(x)$ なので，面積は $g(x) - f(x)$ を $1 - \sqrt{6}$ から $\sqrt{6} + 1$ まで積分することによって求まることがわかります．

```
(%i9)  integrate(g(x)-f(x),x, kai[1], kai[2]);
```

$$\text{(\%o9)} \quad \frac{12\sqrt{6}+17}{3} + \frac{12\sqrt{6}-17}{3}$$

```
(%i10)  factor(%)
```

$$\text{(\%o10)} \quad 8\sqrt{6}$$

以上で，面積が $8\sqrt{6}$ であることがわかりました．

演習問題

1. 関数 $f(x) = 1 - |2x - 1|$ $(0 \leq x \leq 1)$ に対して，f, f^2, f^3, f^4 のグラフを，1 つの座標平面に描け．ただし f^n は f の n 回の合成関数 $\underbrace{f \circ \cdots \circ f}_{n}$ を表す．また，次で与えられる関数 g のグラフを描け．

$$g = \frac{f}{2} + \frac{f^2}{2^2} + \cdots + \frac{f^{10}}{2^{10}}$$

注意：(1) f, f^2, f^3 などは，たとえば `f[1]`, `f[2]`, `f[3]` などのように，リストを用いて定義してやるとよいかもしれない．具体的には，次のように帰納的に定義する．

```
f[1](x):=1-abs(2*x-1); f[n](x):=f[1](f[n-1](x));
```

(2) 出来上がりがきれいになるように，`plot2d` のオプションをうまく設定して書くこと．

(3) 次の無限級数により定義される関数 T は高木関数とよばれている．

$$T = \frac{f}{2} + \frac{f^2}{2^2} + \cdots + \frac{f^n}{2^n} + \cdots$$

この関数は，区間 $[0, 1]$ で連続であるが，いたるところ微分不可能になっている．また，この関数のグラフはフラクタルの一つの例になっている．

2. 2 つの曲線 $y = x^3 - 4x$, $y = -x^2 + 4$ で囲まれる領域の面積を求めよ．

3. 関数 $f(x) = x \sin \frac{x}{2}$ を考える．点 $(\pi, f(\pi))$ における曲線 $y = f(x)$ の接線 ℓ の方程式を求めよ．また ℓ とこの曲線 $y = f(x)$ とで囲まれた部分の面積を求めよ．（愛媛大学 1999 年度入試問題）

4. k を正の定数とし，$g(x)$ を連続な関数とする．関数 $h(x)$ を

$$h(x) = \int_0^x g(t) e^{k(x-t)}\, dt$$

とすると，$h'(x) = kh(x) + g(x)$ であることを示せ．また，$f(x)$ を

$$f(x) = \sin kx + \frac{1}{k} \int_0^x g(t) \sin(kx - kt)\, dt$$

としたとき，$f''(x)$ を $f(x)$ 及び $g(x)$ を用いて表せ．（山口大学 1999 年度入試問題）

第9章　行列と行列式

9.1　行列と行列式の計算

行列は matrix を用いて定義します．行列

$$P=\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\end{bmatrix},Q=\begin{bmatrix}1&-1&2\\-2&3&-1\\-1&4&7\end{bmatrix},R=\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}$$

を定義して，$R(2P-Q)$ を計算してみます．行列のスカラー倍は数の積と同じ $*$ を用いますが，行列の積にはピリオッドを用います．

```
(%i1)  P:matrix([0,1,1],[1,0,1],[1,1,0]);
```

$$\text{(\%o1)}\quad\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\end{bmatrix}$$

```
(%i2)  Q:matrix([1,-1,2],[-2,3,-1],[-1,4,7]);
```

$$\text{(\%o2)}\quad\begin{bmatrix}1&-1&2\\-2&3&-1\\-1&4&7\end{bmatrix}$$

```
(%i3)  R: matrix([1,2,3],[4,5,6]);
```

$$\text{(\%o3)}\quad\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix}$$

```
(%i4)  R.(2*P-Q);
```

$$\text{(\%o4)}\quad\begin{bmatrix}16&-9&-15\\34&-15&-27\end{bmatrix}$$

行列 A の n 乗は，A^n でなく A^^n として求めます．ちなみに A^n とすると，各成分が n 乗されてしまいます．

```
(%i5)  A:matrix([1,2,3],[-1,2,-5],[0,2,0]);
```

$$(\%o5)\quad \begin{bmatrix} 1 & 2 & 3 \\ -1 & 2 & -5 \\ 0 & 2 & 0 \end{bmatrix}$$

```
(%i6)  A^^4;
```

$$(\%o6)\quad \begin{bmatrix} -21 & -136 & -79 \\ 53 & -24 & 255 \\ 10 & -96 & 62 \end{bmatrix}$$

```
(%i7)  A^4;
```

$$(\%o7)\quad \begin{bmatrix} 1 & 16 & 81 \\ 1 & 16 & 625 \\ 0 & 16 & 0 \end{bmatrix}$$

行列の列ベクトルや行ベクトルは col や row を用いて取り出すことができます．また addcol や addrow で行列に列や行を追加することができます．

```
(%i8)  B:matrix([2,-3,-1],[3,3,10]);
```

$$(\%o8)\quad \begin{bmatrix} 2 & -3 & -1 \\ 3 & 3 & 10 \end{bmatrix}$$

```
(%i9)  u:col(B,2);
```

$$(\%o9)\quad \begin{bmatrix} -3 \\ 3 \end{bmatrix}$$

```
(%i10)  row(B,1);
```

$$(\%o10)\quad \begin{bmatrix} 2 & -3 & -1 \end{bmatrix}$$

```
(%i11)  C:addcol(B,u,matrix([0],[1]));
```

$$(\%o11)\quad \begin{bmatrix} 2 & -3 & -1 & -3 & 0 \\ 3 & 3 & 10 & 3 & 1 \end{bmatrix}$$

```
(%i12)  addrow(C,[1,2,3,4,5]);
```

$$(\%o12)\quad \begin{bmatrix} 2 & -3 & -1 & -3 & 0 \\ 3 & 3 & 10 & 3 & 1 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

ident は単位行列を，zeromatrix は零行列を生成します．

```
(%i13)  ident(4);
```

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
(%i14)  zeromatrix(2,3);
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

ematrix はある成分ひとつだけ値を持ち，他はすべて 0 である行列を生成します．たとえば，$(2,3)$ 成分が -4 である 3×5 行列を生成するには次のようにします．

```
(%i15)  ematrix(3,5,-4,2,3);
```

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

対角成分がすべて同じ対角行列は diagmatrix を用いて簡単に与えることができます．

```
(%i16)  diagmatrix(3,5);
```

$$\begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

行列の各成分が関数で与えられる様な場合は genmatrix を用いて与えます． たとえば (i,j) 成分が $i+j$ であるような 2×3 行列は次のようにして与えます．

```
(%i17)  h[i,j]:=i+j;
```

$$h_{i,j} := i+j$$

```
(%i18)  genmatrix(h,2,3);
```

$$\begin{bmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{bmatrix}$$

(%i17) で h を定義するとき，関数の定義の h(i,j):=i+j ではなく，リストの定義の h[i,j]:=i+j であることに注意してください．

今度は，先ほど定義した行列 P, Q の逆行列を求めてみます．逆行列を求めるには invert を用います．

```
(%i19)  invert(P); invert(Q);
(%o19)  [ -1/2   1/2   1/2 ]
        [                  ]
        [ 1/2   -1/2   1/2 ]
        [                  ]
        [ 1/2    1/2  -1/2 ]
Division by 0
 -- an error. Quitting.  To debug this try debugmode(true);
```

Q の逆行列を求めるのは失敗しました．そこで P, Q の行列式を調べてみます．行列式には determinant を用います．

```
(%i21)  determinant(P); determinant(Q);
(%o21)  2
(%o22)  0
```

P の行列式は 2 であるため P は正則行列であることが分かります．一方 Q の行列式は 0 であり，Q は逆行列を持たないことが分かります．

転置行列は transpose を用います．ここでは，P, Q, R の転置行列を，map を用いていっぺんに求めてみます．

```
(%i23)  map(transpose,[P,Q,R]);
        [ [ 0  1  1 ]  [  1  -2  -1 ]  [ 1  4 ] ]
        [ [ 1  0  1 ]  [ -1   3   4 ]  [ 2  5 ] ]
(%o23)  [ [ 1  1  0 ] , [  2  -1   7 ] , [ 3  6 ] ]
```

この map は与えられた関数の複数の変数に対しての値を求めるとき便利です．たとえば，次のようにして用います．

```
(%i24)  f(x):=x^2$
(%i25)  map(f,[1,2,3]);
(%o25)  [1, 4, 9]
```

数ベクトルは，列がただ一つの特別な行列と考えることができるので，たとえばベクトル $\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ は matrix([1],[0],[-1]) として与えることができます．しかし，もっと簡単にリスト [1,0,-1] の転置行列として与えることもできます．

```
(%i26)  v:transpose([1,0,-1]);
(%o26)  [ 1 ]
        [ 0 ]
        [-1 ]
```

連立一次方程式の係数行列は coefmatrix で得られます．また，拡大係数行列は augcoefmatrix で得られます．

```
(%i27)  eq:[2*x-3*y-1=0,3*x+3*y=10];
(%o27)  [-3*y+2*x-1=0,3*y+3*x=10]
(%i28)  coefmatrix(eq,[x,y]);
(%o28)  [ 2  -3 ]
        [ 3   3 ]
(%i29)  augcoefmatrix(eq,[x,y]);
(%o29)  [ 2  -3  -1  ]
        [ 3   3  -10 ]
```

実を言うと，正確には augcoefmatrix は拡大係数行列にはなっていません．上の例だと，拡大係数行列は $\begin{bmatrix} 2 & -3 & 1 \\ 3 & 3 & 10 \end{bmatrix}$ でなくてはなりません．でもいいでしょう．

正方行列の余因子行列を求めるには adjoint を用います．

```
(%i30)  B:matrix([1,2,3],[0,5,6],[7,8,9]);
                                [1  2  3]
                                [        ]
(%o30)                          [0  5  6]
                                [        ]
                                [7  8  9]
(%i31)  adjoint(B);
                                [ -3   6   -3 ]
                                [             ]
(%o31)                          [ 42  -12  -6 ]
                                [             ]
                                [-35   6    5 ]
```

行列の階数を求めるには rank を用います.

```
(%i32)  rank(B);
(%o32)  3
```

トレースは mattrace を用いますが，この場合は前もって nchrpl をロードしておく必要があります.

```
(%i33)  load("nchrpl");
(%o33)  C:/PROGRA~1/MAXIMA~1.1/share/maxima/5.17.1/share/matrix/nchrpl.mac
(%i34)  mattrace(B);
(%o34)  15
```

今度は行列 P の固有値および固有ベクトルを求めましょう．ここで P の固有値 λ と，λ の固有ベクトル $\boldsymbol{v}$ ($\neq \mathbf{0}$) とは

$$P\boldsymbol{v} = \lambda\boldsymbol{v}$$

を満たすものです.

例えば，先ほどの行列 $P = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$ とベクトル $v = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$ に対して，次のように $Pv = (-1)v$ となるのがわかるので，-1 は P の固有値で，v は固有値 -1 に対する固有ベクトルとなっていることがわかります.

```
(%i35)  P.v;
                                  [-1]
                                  [ ]
(%o35)                            [ 0 ]
                                  [ ]
                                  [ 1 ]
```

固有値を求めるためには，最初に eigen というマクロを読み込む必要があります．

```
(%i36)  load("eigen");
(%o36)  C:/PROGRA~1/MAXIMA~1.1/share/maxima/5.17.1/share/matrix/eigen.mac
(%i37)  eivects(P);
(%o37)  [[[2, -1], [1, 2]], [1, 1, 1], [1, 0, -1], [0, 1, -1]]
```

最後の出力について説明します．出力は，第 1 要素が [[2,-1],[1,2]], 第 2 要素から第 4 要素が [1,1,1], [1,0,-1], [0,1,-1] のリストになっています．各要素はリストになっていますが，第 1 要素は他の要素と形が違って，リストを要素としたリストになっています．一般に出力の第 1 要素は

$$[[e_1, e_2, \ldots, e_k], [m_1, m_2, \ldots, m_k]]$$

という形をしており，e_i が固有値で m_i は固有値 e_i の重複度を表しています．第 2 要素以降が固有空間の基底である固有ベクトルの列になり，最初の m_1 個は e_1 の固有空間の基底，次の m_2 個が e_2 の固有空間の基底といった具合です．

上の例では，最初の要素から，2 と -1 が固有値であり，2 の重複度は 1, -1 の重複度は 2 であることが分かります．

その次から，ベクトル $\begin{bmatrix}1\\1\\1\end{bmatrix}$ が固有値 2 に対する 1 つの固有ベクトルであり，ベクトル $\begin{bmatrix}1\\0\\-1\end{bmatrix}, \begin{bmatrix}0\\1\\-1\end{bmatrix}$ が固有値 -1 に対する線形独立な 2 つの固有ベクトルになることが分かります．

そこで P を対角化しましょう．一般に n 次正方行列 A の固有値が $e_1, \ldots, e_n$ で，それらに関する固有ベクトルの線形独立な組 $v_1, \ldots, v_n$ が与えられたと

き，これらのベクトルを 列とする行列 $X = [v_1 \dots v_n]$ により

$$X^{-1}AX = \begin{bmatrix} e_1 & 0 & \dots & 0 \\ 0 & e_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_n \end{bmatrix}$$

となることがわかります.

今の場合は

$$X = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \quad \text{とすると} \quad X^{-1}PX = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

が得られることになります.

まず，`eivects(P)` の出力から，固有ベクトルだけからなるリストを作り，それを `x` とします．リストの 1 番目を削除したリストは `rest` を用いて求めることができます.

```
(%i38)  x:rest(%);
(%o38)  [[1,1,1],[1,0,−1],[0,1,−1]]
```

次に `x` の各要素を各行にした行列を考え，その転置行列を X とし，$X^{-1}PX$ を計算します.

```
(%i39)  X:transpose(matrix(x[1],x[2],x[3]));
(%o39)  [1  1  0 ]
        [1  0  1 ]
        [1 −1 −1 ]
(%i40)  invert(X).P.X;
(%o40)  [2  0  0 ]
        [0 −1  0 ]
        [0  0 −1 ]
```

固有値を対角成分とする対角行列が得られました.

これを用いて P^n を次のようにして計算することができます.

$$\begin{aligned} P^n &= (XX^{-1})P(XX^{-1})P\cdots P(XX^{-1}) \\ &= X(X^{-1}PX)^nX^{-1} \\ &= X\begin{bmatrix} 2^n & 0 & 0 \\ 0 & (-1)^n & 0 \\ 0 & 0 & (-1)^n \end{bmatrix}X^{-1} \end{aligned}$$

Maxima では行列 `A` に対して `A^n` は行列の n 乗でなく，各成分の n 乗を与えます．ここで出力 (`%o40`) は対角行列なので，その n 乗は各成分の n 乗になるので，次のようにして P^n が求まります.

```
(%i41)  %^n;
```

$$\begin{bmatrix} 2^n & 0 & 0 \\ 0 & (-1)^n & 0 \\ 0 & 0 & (-1)^n \end{bmatrix}$$

```
(%i42)  X.%.invert(X);
```

$$\begin{bmatrix} \frac{2^n}{3}+\frac{2(-1)^n}{3} & \frac{2^n}{3}-\frac{(-1)^n}{3} & \frac{2^n}{3}-\frac{(-1)^n}{3} \\ \frac{2^n}{3}-\frac{(-1)^n}{3} & \frac{2^n}{3}+\frac{2(-1)^n}{3} & \frac{2^n}{3}-\frac{(-1)^n}{3} \\ \frac{2^n}{3}-\frac{(-1)^n}{3} & \frac{2^n}{3}-\frac{(-1)^n}{3} & \frac{2^n}{3}+\frac{2(-1)^n}{3} \end{bmatrix}$$

```
(%i43)  factor(%)
```

$$\begin{bmatrix} \frac{2^n+2(-1)^n}{3} & \frac{2^n-(-1)^n}{3} & \frac{2^n-(-1)^n}{3} \\ \frac{2^n-(-1)^n}{3} & \frac{2^n+2(-1)^n}{3} & \frac{2^n-(-1)^n}{3} \\ \frac{2^n-(-1)^n}{3} & \frac{2^n-(-1)^n}{3} & \frac{2^n+2(-1)^n}{3} \end{bmatrix}$$

9.2 応用

漸化式で与えられる数列の一般項を行列を用いて解いてみます．フィボナッチ数列と呼ばれる有名な数列があります．これは

$$f_1 = f_2 = 1, \qquad f_{n+2} = f_n + f_{n+1}\ (n \geq 1)$$

で与えられる数列 $\{f_n\}$ です．ここで 2 次元ベクトル $\boldsymbol{v}_n$ を $\boldsymbol{v}_n = \begin{bmatrix} f_{n+1} \\ f_n \end{bmatrix}$ $(n \geq 1)$ で定めます．このとき，行列 $A = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ を考えると，条件式は次

のようになります.

$$\boldsymbol{v}_1 = \begin{bmatrix} f_2 \\ f_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \qquad Av_{n-1} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} f_n \\ f_{n-1} \end{bmatrix} = \begin{bmatrix} f_n + f_{n-1} \\ f_n \end{bmatrix} = \begin{bmatrix} f_{n+1} \\ f_n \end{bmatrix} = \boldsymbol{v}_n$$

よって，上のようにして A^{n-1} を求めると，一般項 f_n は

$$\boldsymbol{v}_n = A^{n-1}\boldsymbol{v}_1$$

から求まります.

演習問題

1. 行列 $A = \begin{pmatrix} a & 1 & 0 \\ 0 & a & 1 \\ 0 & 0 & a \end{pmatrix}$ に対して，$A^n = \begin{pmatrix} a^n & na^{n-1} & \dfrac{n(n-1)a^{n-2}}{2} \\ 0 & a^n & na^{n-1} \\ 0 & 0 & a^n \end{pmatrix}$ であることを帰納法で示せ.

   ヒント： $\begin{pmatrix} a^n & na^{n-1} & \dfrac{n(n-1)a^{n-2}}{2} \\ 0 & a^n & na^{n-1} \\ 0 & 0 & a^n \end{pmatrix}$ を n の関数として定義します.

   つまり行列 $A(n)$ を次のように定義します.

   ```
   A(n):=matrix([a^n,n*a^(n-1),n*(n-1)*a^(n-2)/2],[0,a^n,n*a^(n-1)],[0,0,a^n])
   ```

   このとき $A^n = A(n)$ を帰納的に示せばよいのです.

2. $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ を

$$f\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x\cos t - y\sin t \\ x\sin t + y\cos t \end{pmatrix}$$

   と定める．このとき，$\mathbb{R}^2$ の標準基底 $\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ に関する f の表現行列 M を求めよ．また $M^n = \begin{pmatrix} \cos nt & -\sin nt \\ \sin nt & \cos nt \end{pmatrix}$ $(n \geq 1)$ となることを帰納法で示せ.

   ヒント：三角関数を含む式を加法定理を用いてただ一つの sin または cos のみを含む式の和として表すには `trigreduce` を用います．たとえば `trigreduce(2*sin(t)*cos(t))` とすると，`sin(2t)` が帰ってきます.

第10章　平面図形

10.1　陽関数のグラフ

平面図形を描くには，関数のグラフと同様に wxplot2d または plot2d を用います．ここでも gnuplot を用いて記述します．

関数 $y = f(x), y = g(x), \ldots$ のグラフを $a \leq x \leq b$ の範囲で描くには，すでにやったように

```
plot2d([f(x),g(x),...],[x,a,b])
```

のようにします．しかし，必ずしもうまくいかないことがあります．たとえば，$y = \dfrac{1}{x}$ を $-5 \leq x \leq 5$ の範囲で描いてみます．

```
(%i1)  plot2d(1/x,[x,-5,5]);
```

表示されるグラフはめちゃめちゃ変になったと思います．これは x が 0 に近づくと，関数 $y = 1/x$ の値の絶対値が限りなく大きくなる（つまり直線 $x = 0$ がこのグラフの漸近線になる）ためです．実際，グラフをよく見ると，画面に現れている x 軸の範囲と y 軸の範囲が大きく異なっていることが分かります．そこで，今度は y の範囲も指定してやると，そこからはみ出す部分は描画しなくなり，きれいなグラフが表示できます．

```
(%i2)  plot2d(1/x,[x,-5,5],[y,-5,5]);
```

ちなみに，Plot 2D 窓を用いる場合は，下の Variable: で y の範囲を指定します．

10.2　媒介変数表示される曲線

媒介変数表示 $(x(t), y(t))$ $(a \leq t \leq b)$ で与えられている平面曲線を描くには

```
plot2d([parametric,x(t),y(t)],[t,a,b]);
```

のように parametric というオプションを用います．

半径1の円を描いてみます．半径1の円は，媒介変数表示で $(x, y) = (\cos t, \sin t)$ $(0 \leq t \leq 2\pi)$ なので，次のようにしてみます．

```
(%i3)  plot2d([parametric,cos(t),sin(t)],[t,0,2*%pi]);
```

Plot 2D 窓を用いる場合は，Expression(s): に，そのままずらずら

```
parametric,cos(t),sin(t)
```

と書き込めばよいです.

上の円は，よく見るときちんとした円でなく，多少ぎざぎざの曲線で結ばれていることがわかります．これは以前にも書きましたが，Maxima が区間 $0 \leq t \leq 2\pi$ をいくつかの点で等分し，各分点の値を代入してできる平面上の点を順番に結ぶことにより図形を描いているからです．区間の分割数は，nticks という変数に格納されていますので，nticks の値を大きくしてやると（たとえば 300 くらいに），よりきれいな円が描かれることになります.

逆に nticks の値を小さくすると，正多角形を描くことができます．つまり正 n 角形を描くには nticks を $n+1$ にして円を表示すればよいことになります．たとえば，正 5 角形なら

```
(%i4)  plot2d([parametric,cos(t),sin(t)],[t,0,2*%pi],[nticks,
6]);
```

のようにします.

さらに，t の範囲を $0 \leq t \leq 4\pi$ にすると，今度は星型が表示されます.

```
(%i5)  plot2d([parametric,cos(t),sin(t)],[t,0,4*%pi],[nticks,
6]);
```

一般に n と m が互いに素な自然数のとき，nticks を $n+1$ にして $0 \leq t \leq 2m\pi$ の範囲で描いてできる星型を正 n/m 角形といいます．上の星型は正 $5/2$ 角形です.

通常の陽関数のグラフとパラメータオプションの図の混じった描画も可能です．たとえば，$y = x^3 + 2$ $(-3 \leq x \leq 3)$ と単位円を同時に描きたいときは次のようにします.

```
(%i6)  plot2d ([x^3 + 2, [parametric, cos(t), sin(t)]],
[x, -3, 3], [y, -3, 3], [t, 0, 2*%pi];
```

上の例では nticks を共通にとっていますが，複数のパラメーターオプションの図で，パラメーターの範囲や nticks の値を個別に設定したいときは

```
plot2d ([[parametric, x(t), y(t), [nticks, n], [t, a, b]],
[parametric, x'(t), y'(t), [nticks, n'],[t, a', b']]]);
```

のようにします.

最後にスピノグラフを描いて見ましょう.

```
(%i7)  a:96$ b:45$
(%i9)  load("functs")$
(%i10)  c:lcm(a,b);
(%o10)  1440
(%i11)  A:a/(2*%pi)$ B:b/(2*%pi)$ C:B*0.8$
(%i14)  plot2d([parametric, (A-B)*cos(t)-C*cos((a-b)/b*t),
(A-B)*sin(t)+C*sin((a-b)/b*t)], [t,0,2*%pi*c/a],[nticks,
1000]);
```

演習問題

1. サイクロイドを描け．サイクロイドとは，円が直線の上を滑らずに転がるときに，円の周上の 1 点が描く軌跡をいう．転がる円の半径が 1 のとき，サイクロイドは t を媒介変数として $(x, y) = (t - \sin t, 1 - \cos t)$ で与えられる．

2. アステロイド：$(x, y) = (\cos^3 t, \sin^3 t)$ $(0 \leq t \leq 2\pi)$ を描け．

3. 極方程式 $r = \cos 3t$ $(0 \leq t \leq 2\pi)$ で与えられる三葉線を描け．また，$r = \cos 2t$ $(0 \leq t \leq 2\pi)$ で与えられる四葉線を描け．ただし極方程式 $r = r(t)$ $(a \leq t \leq b)$ で与えられる曲線とは，$\{\, (r(t)\cos t, r(t)\sin t) \mid a \leq t \leq b \,\}$ と媒介変数表示される曲線である．

4. 極方程式 $r = 1 + \cos t$ $(0 \leq t \leq 2\pi)$ で与えられるカージオイドを描け．

5. 極方程式 $r = e^{\sin t} - 2\cos(4t) + \sin(t/12)$ $(0 \leq t \leq 24\pi)$ で与えられるフェイの蝶を描け．

6. 頂点を共有する正 5 角形と正 5/2 角形を同一の平面に描け．

7. 平面内の x 軸上の線分 $\{\, (x, 0) \mid 1 \leq x \leq 3 \,\}$ を，点 $(2, 0)$ を中心に角 $2\pi/7$ だけ回転した線分を，正方形の領域 $\{(x, y) \mid -4 \leq x \leq 4,\ -4 \leq y \leq 4\}$ 内に描け．

第11章　空間図形

11.1　陽関数のグラフ

空間図形を描くには wxplot3d または plot3d を用いますが，表示した図形を回転させることができる plot3d がベターです．関数 $z = f(x, y)$ のグラフを $a \leq x \leq b$, $c \leq y \leq d$ の範囲で描くには，すでにやったように

```
plot3d(f(x,y),[x,a,b],[y,c,d])
```

のようにします．ただし，平面図形のように複数の図形を同時に表示することはできません．めちゃめちゃ残念です．

関数 $z = \dfrac{x^2 y}{x^4 + y^2}$ を $-1 \leq x \leq 1$, $-1 \leq y \leq 1$ の範囲で描くには次のようにします．

```
(%i1)   plot3d(x^2*y/(x^4+y^2),[x,-1,1],[y,-1,1]);
```

図形のぎざぎざが気になるときは，すでに書いたように grid というオプションを大きくしてやります．ちなみに grid の初期値は 30×30 です．

```
(%i2)   plot3d(x^2*y/(x^4+y^2),[x,-1,1],[y,-1,1],[grid,50,50]);
```

11.2　媒介変数表示される曲面

媒介変数 $(x(s,t), y(s,t), z(s,t))$ $(a \leq s \leq b,\ c \leq t \leq d)$ で与えられている曲面を描くには，単に

```
plot3d([x(s,t),y(s,t),z(s,t)],[s,a,b],[t,c,d]);
```

のようにします．平面曲線のときのように parametric というオプションは必要ありません．たとえば，半径 1 の球面は，媒介変数で

$$(x, y, z) = (\cos s \cos t, \cos s \sin t, \sin s) \quad (-\pi/2 \leq s \leq \pi/2,\ 0 \leq t \leq 2\pi)$$

なので次のようになります．

```
(%i3)   plot3d([cos(s)*cos(t),cos(s)*sin(t),sin(s)],[s,-%pi/2,%pi/2],
[t,0,2*%pi]);
```

平面図形のときと同様にgridの値を小さくすると，多面体が描けます．残念ながら，正多面体になるわけではありませんが，たとえば球面を[grid,2,4]で描くと正八面体になります．

```
(%i4)  plot3d([cos(s)*cos(t),cos(s)*sin(t),sin(s)],[s,-%pi/2,%pi/2],
[t,0,2*%pi],[grid, 2, 4]);
```

演習問題

1. xz 平面の線分 $\{\,(3,0,t) \mid -1 \leq t \leq 1\,\}$ を z 軸を中心に 1 回転したときできる曲面を描け．この曲面をアニュラスとよぶ．

2. 次をヒントにしてトーラスを描け．

   (a) まず，x 軸上の点 $(3,0,0)$ を中心に xz 平面内で半径 1 の円を考える．

   (b) この円を媒介変数表示で $(x(t), y(t), z(t))$ $(a \leq t \leq b)$ のように表す．

   (c) この円を 1 のように z 軸を中心に 1 回転したとき，その軌跡として得られるのがトーラスである．

3. 次をヒントにしてメビウスの帯を描け．トーラスを描いたときと同様に考えると，xz 平面内の線分を線分の中心を軸に自転させながら，z 軸を中心に 1 回転させてやる．その際 z 軸中心に 1 回転する間に，線分が $1/2$ 回転自転するようにすれば，出来上がりはメビウスの帯になる．より詳しくは，次のようにする．

   (a) まず，xz 平面の長さ 1 の線分で $(3,0,0)$ を中点とし，x 軸とのなす角が θ $(0 \leq \theta \leq \pi)$ であるものを l_θ とする．

   (b) l_θ を媒介変数表示で $(x_\theta(t), y_\theta(t), z_\theta(t))$ $(a \leq t \leq b)$ のように表す．

   (c) l_θ を z 軸を中心に角 2θ 回転して得られる線分を l'_θ とすると，線分の束 $\{\,l'_\theta \mid 0 \leq \theta \leq \pi\,\}$ 全体からなる曲面がメビウスの帯になる．

第12章　プログラム

12.1　条件分岐

Maxima では条件分岐のための `if` 文が用意されています．書式は

`if` 条件式 `then` 真の場合の処理 `else` 偽の場合の処理;

のようになります．条件式には，等号や不等号 $=, <, >, >=, <=$，あるいは等しくないかを問う $\#$ などを使うことができます．

次のような関数 $f(x)$ を考えます．

$$f(x) = \begin{cases} x^2 & x > 0 \text{ のとき} \\ 0 & x \leq 0 \text{ のとき} \end{cases}$$

この関数は，次のように条件分岐を使って定義できます．

```
(%i1)  f(x):=if x>0 then x^2 else 0;
```

条件分岐は入れ子にもできます．たとえば，上の関数を次のように変更したいとします．

$$f(x) = \begin{cases} x^2 & x > 0 \text{ のとき} \\ 0 & 0 \geq x > -2 \text{ のとき} \\ x^3 + 8 & x \leq -2 \text{ のとき} \end{cases}$$

この場合，次のように定義します．

```
(%i2)  f(x):=if x>0 then x^2 else (if x> -2 then 0 else
x^3+8);
```

関数を再帰的に定義することもできます．たとえば，$f(x) = x!$ を定義するには次のようにします．

```
(%i3)  f(x):=if x=0 then 1 else x*f(x-1);
(%o3)  f(x) := if x = 0 then 1 else xf(x − 1)
(%i4)  f(5);
(%o4)  120
```

上では，$x = 0$ に対し $f(0) = 0! = 1$ と定義し，$x > 1$ に対しては $x! = x \times (x-1)!$ を用いて，$f(x) = xf(x-1)$ で定義しています．

ある数が正または 0 であるかどうかをチェックするためのプログラム sei を作ってみます．

```
(%i5)  sei(x):= if x > 0 then "正 の 数" else (if x=0 then "
ゼロ" else "負の数");
(%i6)  sei(0.5);
(%o6)  正の数
(%i7)  sei(0);
(%o7)  ゼロ
(%i8)  sei(-3);
(%o8)  負の数
```

上のプログラムで引用符" で囲まれた"正の数"などは，引用符内の文字（この場合は正の数）を印字させるための表現です．

等号や不等号以外にも，変数の属性を問うための関数を用いることができます．変数が整数かを問う integerp，偶数かをを問う evenp，奇数かを問う oddp，素数かを問う primep，有理数かを問う ratnump，リストかを問う listp，行列かを問う matrixp などを用いることもできます．

たとえば，引数として与えられた正の整数が素数かどうかを判定するために用いられる関数 primep を用いてみます．

```
(%i9)  primep(2);
(%o9)  true
(%i10)  primep(4);
(%o10)  false
```

自然数 n が与えられたとき，n 以下の素数の個数を求める関数 primen を作ってみます．

```
(%i11)  primen(n):=if n=2 then 1 else (if primep(n) then
primen(n-1)+1 else primen(n-1))$
(%i12)  primen(100)
(%o12)  25
(%i13)  primen(1000)
Maxima encountered a Lisp error:
Error in PROGN [or a callee]: Bind stack overflow.
Automatically continuing.
To reenable the Lisp debugger set *debugger-hook* to nil.
```

100 以下の素数の個数は 25 個であることが分かりますが，次に 1000 以下の素数の数を求めようとしてエラーになりました．スタックがオーバーフローした様です．

x と $x+2$ が共に素数であるとき，これらを双子素数といいます．そこで，数 x が与えられたとき x と $x+2$ が双子素数かどうかを問う関数 `twinprime` は次の様に定めることができます．

```
(%i14)  twinprime(x):=if primep(x) and primep(x+2) then
          print(x,"と",x+2,"は双子素数") else print
          (x,"または",x+2,"は素数ではない");
(%o14)  twinprime(x):=if primep(x) and primep(x+2) then print
(x, と,x+2, は双子素数) else print(x, または,x+2, は素数ではない)
(%i15)  twinprime(3)$
3 と 5 は双子素数
(%i16)  twinprime(7)$
7 または 9 は素数ではない
(%i17)  twinprime(11777)$
11777 と 11779 は双子素数
```

上の例で `twinprime(7)$`のように`$`を用いていますが，`$`を用いないと次のようになります．

```
(%i18)  twinprime(3);
3 と 5  は双子素数
(%o18)  は双子素数
```

12.2　繰り返し操作

繰り返し操作のためには for 文が用意されています．書式は

for カウンタ名:初期値 step 増分 thru 終了値 do (反復実行手続き);
for カウンタ名:初期値 step 増分 while 条件式 do (反復実行手続き);

の 2 種類があります．この step は省略することができ，その場合は 1 が指定されたものとなります．そして，do の括弧の中身は，複数の作業を指定するときは，順番にコンマで区切って列挙します．

1000 以下の素数の個数を繰り返し操作を用いて求めてみましょう．

```
(%i19)  j:0;
(%o19)  0
(%i20)  for i:2 thru 1000 do (if primep(i) then j:j+1);
(%o20)  done
(%i21)  j;
(%o21)  168
```

今度は 1000 以下の素数の個数も，きちんと 168 個と計算できたようです．

方程式を数値計算によって解くための反復法によるアルゴリズムの 1 つにニュートン法と呼ばれるものがあります．これは，$f(x) = 0$ で与えられる方程式に対して，解の近くの値 x_0 を適当に決めてやり，漸化式

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad (n \geq 1)$$

により数列 x_n を定めると，この数列が $f(x) = 0$ の解に収束することが多いということによります．よって，十分大きな n をとれば，x_n は $f(x) = 0$ の解の一つの近似値を与えることになります．

たとえば，$f(x) = x^2 - 2 = 0$ の解の近似値をニュートン法で求めてみます．(もちろんこの解は $\sqrt{2}$ です．)

```
(%i22)  f(x):=x^2-2$
(%i23)  define(g(x),x-f(x)/diff(f(x),x));
(%o23)  g(x) := (x^2 - 2)/(2x)
(%i24)  a:4;
(%o24)  4
(%i25)  for i:1 thru 10 do(a:g(a));
(%o25)  done
(%i26)  a,numer;
(%o26)  1.414213562373095
(%i27)  a:-1;
(%o27)  -1
(%i28)  for i:1 thru 10 do(a:g(a));
(%o28)  done
(%i29)  a,numer;
(%o29)  -1.414213562373095
```

関数の定義に define を用いていることに注意してください．

上の例で，初期値を 4 にとると，$\sqrt{2}$ の近似値である 1.414213562373095 を得ることができ，-1 にすると，今度は $-\sqrt{2}$ の近似値である -1.414213562373095 を得ます．

12.3 ブロック化

既存の関数に条件分岐や繰り返し操作を加えると，様々な処理を行うことができます．れらの手順をまとめて一つの関数にする方法がこのブロック化です．ブロック化するときの書式は，

block([局所変数のリスト], 一連の手続き, return(計算結果));

のようになります．

前節の繰り返し操作を用いた素数の個数の求め方を一般化して，n 以下の素数の個数を求める関数 primen2 を作ってみましょう．

```
(%i30)  primen2(n):=block([j:0], for i:2 thru n do (if
primep(i) then j:j+1), return(j))$
(%i31)  primen2(100)
(%o31)  25
(%i32)  primen2(1000)
(%o32)  168
```

上のプログラムでは block の中で，まず局所変数 j を与え，その初期値として 0 を代入しています．その後に，2 から n までの整数 i が素数かどうかをチェックし，素数であれば j に 1 を加えてゆき，最後まで調べ終わったら，j の値を返しています．

あるクラスの全員の試験の点数のリストが与えられたとき，合格者の人数を求める関数 goukaku を作ってみます．ここで，成績のリストとは，全員の点数を並べた [70,45,82,100,13,0,68,93,48] のようなものとします

```
(%i33)  goukaku(lis):= block([ninzuu:0], for i:1 thru
length(lis) do (if lis[i] >= 60 then ninzuu:ninzuu+1),
return(ninzuu))$
(%i34)  goukaku([70,45,82,100,13,0,68,93,48]);
(%o34)  5
```

上のプログラムでは block の中で，まず局所変数 ninzuu を与え，初期値として 0 を代入しています．その後に，リスト lis の要素を 1 番目から順に 60 以上かを調べ，60 以上の点数があるたびに，変数 ninzuu に 1 を加えてゆき，最後まで調べ終わったら，最後に ninzuu の値を返しています．ちなみに length はリストの要素の数を与える関数です．

演習問題

1. 自然数 n に対する値 $f(n)$ がフィボナッチ数列の第 n 項を与える関数 f を定義し，初項から第 20 項までのリストを作れ．ここでフィボナッチ数の第 n 項 $f(n)$ は，$n = 1, 2$ のとき，$f(n) = 1$，$n \geq 3$ のとき，$f(n) = f(n-1) + f(n-2)$ で与えられる．

2. 自然数 n が与えられたとき，n 番目の素数を求める関数 nprime を作れ．また，1000 番目の素数を求めよ．

3. すべての正の整数上で定義された関数 $f(n)$ を次で定める．

$$f(n) = \begin{cases} n/2 & n \text{ が偶数のとき} \\ 3n+1 & n \text{ が奇数のとき} \end{cases}$$

このとき Collatz の予想と呼ばれるものは，$f(n), f^2(n), f^3(n), \ldots$ と繰り返していけば，どのような正の整数 n に対しても必ず 1 に到達するというものである．この予想は $3 \times 253 = 27,021,597,764,222,976$ 以下で正しいことが知られている．それでは，この予想が正しいとして，1 より大きな整数 n が与えられたとき，$f^k(n) = 1$ となる最小の $k > 1$ を

求める関数 nankai を作れ．また，n が 2 から 100 までの，nankai(n) の値のリストを作れ．