OpenXM/Risa/Asir-Contrib

OpenXM/Risa/Asir-Contrib User's Manual (日本語版)
Edition 1.2.3 for OpenXM/Asir2000
April 2005.

by OpenXM Developing Team

1 はじめに

数式処理システム asir は OpenXM プロトコル (Open message eXchange for Mathematics, http://www.openxm.org) をサポートしたサーバをコンポーネントとして利用できる. これらのサーバを呼ぶためのインタフェース関数はファイル 'OpenXM/rc/asirrc' をロードすることによりシステムに読み込まれる. Risa/Asir (OpenXM 配布版) では起動時に自動的にこのファイルが読まれる. Risa/Asir (OpenXM 配布版) は, このマニュアルでは OpenXM/Risa/Asir と呼ぶ. このマニュアルでは asir 用のこれらの関数およびユーザ言語で書かれた数学関数およびユーティリティ関数を説明する.

OpenXM プロトコルの技術的詳細については, '$(OpenXM_HOME)/doc/OpenXM-specs' にあるファイル 'openxm-jp.tex' を見て下さい.

それでは, あなたの計算機上で数学をお楽しみ下さい.

List of contributors:

- Maekawa, Masahide (Oct., 1999 – : CVS server)
- Noro, Masayuki (Jan., 1996 – : OpenXM Protocol OXRFC-100, asir2000)
- Ohara, Katsuyoshi (Jan., 1998 – : ox_math, oxc OXRFC-101)
- Takayama, Nobuki (Jan., 1996 – : OpenXM Protocol OXRFC-100, kan/sm1, asir-contrib)
- Tamura, Yasushi (Nov., 1998 – : OpenMath proxy, tfb)
- Fujimoto, Mitsushi (Windows)
- Iwane, Hidenao (Knapsack factorizer)
- Nakayama, Hiromasa (Gaussian elimination)
- Okutani, Yukio (Oct., 1999 – Feb., 2000 : matrix, diff, ...)
- Stillman, Mike (Macaulay 2 client and server)
- Tsai, Harrison (Macaulay 2 client and server)

この Contrib パッケージの著作権については, `OpenXM/Copyright` を見て下さい.
有用だとおもいますが無保証です.

2 Asir/Contrib のロード方法.

'OpenXM/rc/asirrc' をロードすることにより Asir/Contrib の主な関数が利用可能となる. OpenXM/Risa/Asir では ASIR_CONFIG 環境変数によりこのファイルを起動時に読みこんでいる. 'names.rr' が Asir/Contrib のトップレベルのファイルである. このファイルよりその他のファイルが読み込まれている. 一部のパッケージは 'names.rr' からは読み込まれないので, 明示的に読み込む必要がある.

A sample of 'asirrc' to use Asir/Contrib.

```
load("gr")$
load("primdec")$
load("katsura")$
load("bfct")$
load("names.rr")$
load("oxrfc103.rr")$
User_asirrc=which(getenv("HOME")+"/.asirrc")$
if (type(User_asirrc)!=0)
   if (!ctrl("quiet_mode")) print("Loading ~/.asirrc")$
   load(User_asirrc)$
else $
end$
```

3 Asir Contrib の函数名について

Asir Contrib には (1) 標準的な名前で定義された数学函数 (`names.rr` および (2) Asir 標準函数以外の有用なライブラリ函数および (3) OpenXM サーバを asir から呼ぶための函数が含まれている.

Asir Contrib の函数名はモジュール化されているかまたは次の形をしている: カテゴリ名_函数名

標準的な数学函数は実体へのラッパーである. たとえば `sm1.hilbert` は OpenXM サーバ `sm1` の Hilbert 函数の計算函数を呼び出す函数である. 一方 `poly_hilbert_polynomial` は Asir Contrib の Hilbert 函数を計算するための (1) に属する標準的な函数名である. 標準函数 `poly_hilbert_polynomial` は, 現在 `sm1.hilbert` を呼び出して Hilbert 函数を計算しているが, これは将来変更されるかもしれない. たとえば, Asir 言語で記述された有用なライブラリ函数集 `commutativeRing.rr` が開発されて Hilbert 函数の計算函数 `commutativeRing_hilbert_polynomial` が含まれるようになったら, 標準函数 `poly_hilbert_polynomial` は, `commutativeRing_hilbert_polynomial` を呼び出して Hilbert 函数を計算するようになるかもしれない. したがって, ユーザプログラムは標準数学函数名を用いるのが望ましい.

標準数学函数名は, OpenXM project において, 全てのプロジェクトで共通の仕様を持つように努力している. たとえば, kan/k0 も Asir Contrib と同様の標準数学函数名を持つ予定である. 現在実験的に数学函数のカテゴリ `complex` 複体 (複素数でない) のマニュアルを kan/k0, asir/contrib で共通化を試みている.

以下の章は, 標準数学函数の解説をおこない, それからライブラリ函数, それから, OpenXM サーバのインタフェースの説明をおこなう.

4 Windows 版 Asir-contrib

Windows でも不完全ながら asir-contrib が動作する. 現在, 外部コンポーネント sm1 および, 外部 コンポーネント を利用しない asir-contrib の関数が動作する. Cygwin 環境では外部コンポーネント sm1, phc が動作する. その他の外部コンポーネントは動作しない.

次の関数は Windows では動作しない. Windows での cygwin 環境では動作する場合がある.

- `gnuplot.*`
- `om.*`
- `mathematica.*`
- `phc.*`
- `print_dvi_form`
- `print_gif_form`
- `print_open_math_xml_form`
- `print_png_form`
- `print_xdvi_form`
- `print_xv_form`
- `tigers_xv_form`

5 基礎 (標準函数)

5.0.1 base_cancel

`base_cancel(`*S*`)`
: It simplifies S by canceling the common factors of denominators and numerators.

Example:

```
base_cancel([(x-1)/(x^2-1), (x-1)/(x^3-1)]);
```

5.0.2 base_choose

`base_choose(`*L*`,`*M*`)`
: It returns the list of the order M subsets of L.

Example:

```
base_choose([1,2,3],2);
```

It outputs all the order 2 subsets of the set $\{1, 2, 3\}$

5.0.3 base_flatten

`base_flatten(`*S*`)`
: It flattens a nested list S.

Example:

```
base_flatten([[1,2,3],4]);
```

5.0.4 base_intersection

`base_intersection(`*A*`,`*B*`)`
: It returns the intersection of A and B as a set.

Example:

```
base_intersection([1,2,3],[2,3,5,[6,5]]);
```

5.0.5 base_memberq

`base_memberq(`*A*`,`*S*`)`
: It returns 1 if A is a member of the set S else returns 0.

Example:

```
base_memberq(2,[1,2,3]);
```

5.0.6 base_permutation

`base_permutation(L)`
: It outputs all permutations of L. BUG; it uses a slow algorithm.

Example:

```
base_permutation([1,2,3,4]);
```

5.0.7 base_position

`base_position(A,S)`
: It returns the position of A in S.

Example:

```
base_position("cat",["dog","cat","monkey"]);
```

5.0.8 base_prune

`base_prune(A,S)`
: It returns a list in which A is removed from S.

Example:

```
base_prune("cat",["dog","cat","monkey"]);
```

5.0.9 base_replace

`base_replace(S,Rule)`
: It rewrites S by using the rule *Rule*

Example:

```
base_replace(x^2+y^2,[[x,a+1],[y,b]]);
```

x is replaced by a+1 and y is replaced by b in x^2+y^2.

5.0.10 base_set_minus

`base_set_minus(A,B)`
: $A \setminus B$

Example:

```
base_set_minus([1,2,3],[3,4,5]);
```

5.0.11 base_set_union

`base_set_union(A,B)`
: $A \cup B$

Example:

```
base_set_union([1,2,3],[3,4,5]);
```

5.0.12 base_subsetq

`base_subsetq(A,B)`
: if $A \subseteq B$, then it returns 1 else 0.

Example:

```
base_subsetq([1,2],[1,2,3,4,5]);
```

5.0.13 base_subsets_of_size

`base_subsets_of_size(K,S)`
: It outputs all subsets of S of the size K. BUG; it uses a slow algorithm. Do not input a large S.

Example:

```
base_subsets_of_size(2,[3,5,3,2]);
```

6 数 (標準数学函数)

6.0.1 number_abs

`number_abs(`*X*`)`
:

Example:

```
number_abs(-3);
```

6.0.2 number_ceiling

`number_ceiling(`*X*`)`
:

Example:

```
number_abs(1.5);
```

6.0.3 number_factor

`number_factor(`*X*`)`
: It factors the given integer *X*.

Example:

```
number_factor(20);
```

6.0.4 number_floor

`number_floor(`*X*`)`
:

Example:

```
number_floor(1.5);
```

6.0.5 number_imaginary_part

`number_imaginary_part(`*X*`)`
:

Example:

```
number_imaginary_part(1+2*@i);
```

6.0.6 number_is_integer

`number_is_integer(`*X*`)`
:

Example:

```
number_is_integer(2/3);
```

6.0.7 number_real_part

`number_real_part(X)`
:

Example:

```
number_real_part(1+2*@i);
```

7 微積分 (標準数学函数)

8 級数 (標準数学函数)

9 特殊函数 (標準数学函数)

まだ書いてない.

10 行列 (標準数学函数)

10.0.1 matrix_clone

`matrix_clone(`*M*`)`
: It generates the clone of the matrix *M*.

Example:

```
matrix_clone(matrix_list_to_matrix([[1,1],[0,1]]));
```

10.0.2 matrix_det

`matrix_det(`*M*`)`
: It returns the determinant of the matrix *M*.

Example:

```
poly_factor(matrix_det([[1,x,x^2],[1,y,y^2],[1,z,z^2]]));
```

10.0.3 matrix_diagonal_matrix

`matrix_diagonal_matrix(`*L*`)`
: It returns the diagonal matrix with diagonal entries *L*.

Example:

```
matrix_diagonal_matrix([1,2,3]);
```

References:

`matrix_list_to_matrix`

10.0.4 matrix_eigenavalues

`matrix_eigenavalues(`*M*`)`
: It returns the eigenvalues of the matrix *M*.

Example:

```
matrix_eigenvalues([[x,1],[0,y]]);
```

10.0.5 matrix_identity_matrix

`matrix_identity_matrix(`*N*`)`
: It returns the identity matrix of the size *N*.

Example:

```
matrix_identity_matrix(5);
```

References:

`matrix_diagonal_matrix`

10.0.6 matrix_image

`matrix_image(`*M*`)`
: It computes the image of M. Redundant vectors are removed.

Example:

```
matrix_image([[1,2,3],[2,4,6],[1,0,0]]);
```

References:

`matrix_kernel`

10.0.7 matrix_inner_product

`matrix_inner_product(`*A*`,`*B*`)`
: It returns the inner product of two vectors A and B.

Example:

```
matrix_inner_product([1,2],[x,y]);
```

10.0.8 matrix_inverse

`matrix_inverse(`*M*`)`
: It returns the inverse of the matrix M.

Example:

```
matrix_inverse([[1,2],[0,1]]);
```

10.0.9 matrix_kernel

`matrix_kernel(`*M*`)`
: It returns the basis of the kernel of the matrix M.

Example:

```
matrix_kernel([[1,1,1,1],[0,1,3,4]]);
```

10.0.10 matrix_list_to_matrix

`matrix_list_to_matrix(`*M*`)`
: It translates the list M to a matrix.

Example:

```
print_xdvi_form(matrix_list_to_matrix([[1,1],[0,2]]));
```

References:

`matrix_matrix_to_list`

10.0.11 matrix_matrix_to_list

`matrix_matrix_to_list(`*M*`)`
: It translates the matrix *M* to a list.

References:

`matrix_list_to_matrix`

10.0.12 matrix_rank

`matrix_rank(`*M*`)`
: It returns the rank of the matrix *M*.

Example:

```
matrix_rank([[1,1,1,1],[0,1,3,4]]);
```

10.0.13 matrix_solve_linear

`matrix_solve_linear(`*M*`,`*X*`,`*B*`)`
: It solves the system of linear equations M X = B

Example:

```
matrix_solve_linear([[1,2],[0,1]],[x,y],[1,2]);
```

10.0.14 matrix_submatrix

`matrix_submatrix(`*M*`,`*Ind*`)`
: It returns the submatrix of *M* defined by the index set *Ind*.

Example:

```
matrix_submatrix([[0,1],[2,3],[4,5]],[1,2]);
```

10.0.15 matrix_transpose

`matrix_transpose(`*M*`)`
: It returns the transpose of the matrix *M*.

References:

`matrix_list_to_matrix`

11 Graphic(標準数学函数)

まだ書いてない.

12 表示 (標準数学函数)

12.0.1 print_dvi_form

`print_dvi_form(`*S*`)`
: It outputs *S* to a dvi file.

Example:

```
print_dvi_form(x^2-1);
```

References:

`print_xdvi_form` , `print_tex_form`

12.0.2 print_em

`print_em(`*S*`)`
: It outputs *S* by a font to emphasize it.

Example:

```
print_em(x^2-1);
```

12.0.3 print_gif_form

`print_gif_form(`*S*`)`
: It outputs *S* to a file of the gif format.

`print_gif_form(`*S* `|` *table*`=key0)`
: This function allows optional variables *table*

Example:

```
print_gif_form(newmat(2,2,[[x^2,x],[y^2-1,x/(x-1)]]));
```

References:

`print_tex_form`

12.0.4 print_input_form

`print_input_form(`*S*`)`
: It transforms *S* to a string which can be parsed by asir.

Example:

```
print_input_form(quote(x^3-1));
```

12.0.5 print_open_math_tfb_form

print_open_math_tfb_form(*S*)
: It transforms *S* to a tfb format of OpenMath XML.

Description:

It is experimental. You need to load taka_print_tfb.rr to call it.

Example:

```
print_open_math_tfb_form(quote(f(x,1/(y+1))+2));
```

12.0.6 print_open_math_xml_form

print_open_math_xml_form(*S*)
: It transforms *S* to a string which is compliant to OpenMath(1999).

Example:

```
print_open_math_xml_form(x^3-1);
```

References:

www.openmath.org

12.0.7 print_output

print_output(*Obj*)
: It outputs the object *Obj* to a file. If the optional variable *file* is set, then it outputs the *Obj* to the specified file, else it outputs it to "asir_output_tmp.txt". If the optional variable *mode* is set to "w", then the file is newly created. If the optional variable is not set, the *Obj* is appended to the file.

print_output(*Obj* | *file*=key0,*mode*=key1)
: This function allows optional variables *file*, *mode*

Example:

```
print_output("Hello"|file="test.txt");
```

References:

glib_tops , (,)

12.0.8 print_ox_rfc100_xml_form

print_ox_rfc100_xml_form(*S*)
: It transforms *S* to a string which is compliant to OpenXM RFC 100.

Example:

```
print_ox_rfc100_xml_form(x^3-1);
```

References:

www.openxm.org

12.0.9 print_png_form

`print_png_form(`*S*`)`
: It transforms *S* to a file of the format png. dvipng should be installed.

Example:

```
print_png_form(x^3-1);
```

References:

`print_tex_form`

12.0.10 print_terminal_form

`print_terminal_form(`*S*`)`
: It transforms *S* to the terminal form???

12.0.11 print_tex_form

`print_tex_form(`*S*`)`
: It transforms *S* to a string of the LaTeX format.

`print_tex_form(`*S* `| table=key0)`
: This function allows optional variables *table*

Description:

The global variable Print_tex_form_fraction_format takes the values "auto", "frac", or "/". The global variable Print_tex_form_no_automatic_subscript takes the values 0 or 1. BUG; A large input S cannot be translated.

Example:

```
print_tex_form(x*dx+1 | table=[["dx","\\partial_x"]]);
```

The optional variable table is used to give a translation table of asir symbols and tex symbols.

References:

`print_xdvi_form`

12.0.12 print_tfb_form

`print_tfb_form(`*S*`)`
: It transforms *S* to the tfb format.

Example:

```
print_tfb_form(x+1);
```

12.0.13 print_xdvi_form

`print_xdvi_form(`*S*`)`
: It transforms S to a xdvi file and previews the file by xdvi.

Example 0:

```
print_xdvi_form(newmat(2,2,[[x^2,x],[y^2-1,x/(x-1)]]));
```

Example 1:

```
print_xdvi_form(print_tex_form(1/2));
```

References:

`print_tex_form` , `print_dvi_form`

12.0.14 print_xv_form

`print_xv_form(`*S*`)`
: It transforms S to a gif file and previews the file by xv.

`print_xv_form(`*S* `|` *input*`=key0,`*format*`=key1)`
: This function allows optional variables *input*, *format*

Example 0:

```
print_xv_form(newmat(2,2,[[x^2,x],[y^2-1,x/(x-1)]]));
```

Example 1:

```
print_xv_form(x+y | format="png");
```

If the optional variable format="png" is set, png format will be used to generate an input for xv.

References:

`print_tex_form` , `print_gif_form`

13 多項式 (標準数学函数)

13.0.1 poly_degree

poly_degree(*F*)
: It returns the degree of F with respect to the given weight vector.

poly_degree(*F* | *weight*=key0,*v*=key1)
: This function allows optional variables *weight*, *v*

Description:

The weight is given by the optional variable weight w. It returns $\mathrm{ord}_w(F)$

Example:

```
poly_degree(x^2+y^2-4 |weight=[100,1],v=[x,y]);
```

13.0.2 poly_elimination_ideal

poly_elimination_ideal(*I*,*VV*)
: It computes the ideal intersection of I and the monomial ideal generated by VV.

poly_elimination_ideal(*I*,*VV* | *grobner_basis*=key0,*v*=key1)
: This function allows optional variables *grobner_basis*, *v*

Description:

If grobner_basis is "yes", I is assumed to be a Grobner basis. The optional variable v is a list of variables which defines the ring of polynomials.

Example 0:

```
poly_elimination_ideal([x^2+y^2-4,x*y-1],[x]);
```

Example 1:

```
A = poly_grobner_basis([x^2+y^2-4,x*y-1]|order=2,v=[y,x]);
      poly_elimination_ideal(A,[x]|grobner_basis="yes");
```

References:

gr , hgr , gr_mod , dp_*

13.0.3 poly_expand

poly_expand(*F*)
: This is an alias of poly_sort.

References:

poly_sort

13.0.4 poly_factor

`poly_factor(`*F*`)`
: It factorizes the polynomial *F*.

Example:

```
poly_factor(x^10-y^10);
```

13.0.5 poly_gcd

`poly_gcd(`*F*`,`*G*`)`
: It computes the polynomial GCD of *F* and *G*.

Example:

```
poly_gcd(x^10-y^10,x^25-y^25);
```

13.0.6 poly_grobner_basis

`poly_grobner_basis(`*I*`)`
: It returns the Grobner basis of *I*.

`poly_grobner_basis(`*I* `| order=key0,`*v*`=key1)`
: This function allows optional variables *order*, *v*

Description:

The optional variable v is a list of variables which defines the ring of polynomials.

Example:

```
A = poly_grobner_basis([x^2+y^2-4,x*y-1]|order=2,v=[y,x]);
```

13.0.7 poly_hilbert_polynomial

`poly_hilbert_polynomial(`*I*`)`
: It returns the Hilbert polynomial of the ideal I.

`poly_hilbert_polynomial(`*I* `| s=key0,`*v*`=key1)`
: This function allows optional variables *s*, *v*

Description:

The optional variable v is a list of variables.

Example:

```
poly_hilbert_polynomial([x1*y1,x1*y2,x2*y1,x2*y2]|s=k,v=[x1,x2,y1,y2]);
```

13.0.8 poly_initial

`poly_initial(`*I*`)`
: It returns the initial ideal of I with respect to the given order.

`poly_initial(`*I* `|` *order*`=key0,`*v*`=key1)`
: This function allows optional variables *order*, *v*

Description:

The optional variable v is a list of variables. This function computes $\mathrm{in}_{\prec}(I)$

Example:

```
poly_initial([x^2+y^2-4,x*y-1]|order=0,v=[x,y]);
```

13.0.9 poly_initial_coefficients

`poly_initial_coefficients(`*I*`)`
: It computes the coefficients of the initial ideal of I with respect to the given order.

`poly_initial_coefficients(`*I* `|` *order*`=key0,`*v*`=key1)`
: This function allows optional variables *order*, *v*

Description:

The optional variable v is a list of variables. The order is specified by the optional variable order

Example:

```
poly_initial_coefficients([x^2+y^2-4,x*y-1]|order=0,v=[x,y]);
```

13.0.10 poly_initial_term

`poly_initial_term(`*F*`)`
: It returns the initial term of a polynomial F with respect to the given weight vector.

`poly_initial_term(`*F* `|` *weight*`=key0,`*order*`=key1,`*v*`=key2)`
: This function allows optional variables *weight*, *order*, *v*

Description:

The weight is given by the optional variable weight w. It returns $\mathrm{in}_w(F)$

Example:

```
poly_initial_term( x^2+y^2-4 |weight=[100,1],v=[x,y]);
```

13.0.11 poly_solve_linear

`poly_solve_linear(`*Eqs*`,`*V*`)`
: It solves the system of linear equations *Eqs* with respect to the set of variables *V*.

Example:

```
poly_solve_linear([2*x+3*y-z-2, x+y+z-1], [x,y,z]);
```

13.0.12 poly_sort

`poly_sort(`*F*`)`
: It expands *F* with a given variables *v*=V and a given weight *w*=W. It returns a quote object. If *trucate* option is set, the expansion is truncated at the given degree.

`poly_sort(`*F* `| v=key0,w=key1,truncate=key2)`
: This function allows optional variables *v*, *w*, *truncate*

Example:

```
poly_sort((x-y-a)^3 | v=[x,y], w=[-1,-1])
   returns a series expansion in terms of x and y.
```

14 複体 (標準数学函数)

15 グラフィックライブラリ (2 次元)

16 Graphic Library (2 dimensional)

ライブラリ glib は, Risa/Asir の グラフィック基本関数 (draw_obj) に対する, 昔の BASIC のような単純なインタフェースを提供する.

16.0.1 glib_clear

`glib_clear()`
: Clear the screen.

16.0.2 glib_flush

`glib_flush()`
: ; Flush the output. (Cfep only. It also set initGL to 1.).

16.0.3 glib_line

`glib_line(`*X0*`,`*Y0*`,`*X1*`,`*Y1*`)`
: It draws the line [*X0*, *Y0*]– [*X1*, *Y1*] with *color* and *shape*

`glib_line(`*X0*`,`*Y0*`,`*X1*`,`*Y1* `| color=key0,shape=key1)`
: This function allows optional variables *color*, *shape*

Example:

```
glib_line(0,0,5,3/2 | color=0xff00ff);
          glib_line(0,0,10,0 | shape=arrow);
```

16.0.4 glib_open

`glib_open()`
: It starts the ox_plot server and opens a canvas. The canvas size is set to *Glib_canvas_x* X *Glib_canvas_y* (the default value is 400). This function is automatically called when the user calls glib functions.

16.0.5 glib_plot

`glib_plot(`*F*`)`
: It plots an object F on the glib canvas.

Example 0:

```
glib_plot([[0,1],[0.1,0.9],[0.2,0.7],[0.3,0.5],[0.4,0.8]]);
```

Example 1:

```
glib_plot(tan(x));
```

16.0.6 glib_print

`glib_print(X,Y,Text)`
: It put a string *Text* at $[X, Y]$ on the glib canvas.

`glib_print(X,Y,Text | color=key0)`
: This function allows optional variables *color*

Example:

```
glib_print(100,100,"Hello Worlds" | color=0xff0000);
```

16.0.7 glib_ps_form

`glib_ps_form(S)`
: It returns the PS code generated by executing S (experimental).

Example 0:

```
glib_ps_form(quote( glib_line(0,0,100,100) ));
```

Example 1:

```
glib_ps_form(quote([glib_line(0,0,100,100),glib_line(100,0,0,100)]));
```

References:

glib_tops

16.0.8 glib_putpixel

`glib_putpixel(X,Y)`
: It puts a pixel at $[X, Y]$ with *color*

`glib_putpixel(X,Y | color=key0)`
: This function allows optional variables *color*

Example:

```
glib_putpixel(1,2 | color=0xffff00);
```

16.0.9 glib_remove_last

`glib_remove_last()`
: Remove the last object. glib_flush() should also be called to remove the last object. (cfep only).

16.0.10 glib_set_pixel_size

`glib_set_pixel_size(P)`
: Set the size of putpixel to P. 1.0 is the default. (cfep only).

16.0.11 glib_tops

`glib_tops()`
: If Glib_ps is set to 1, it returns a postscript program to draw the picture on the canvas.

References:

`print_output`

16.0.12 glib_window

`glib_window(`*`Xmin`*`,`*`Ymin`*`,`*`Xmax`*`,`*`Ymax`*`)`
: It generates a window with the left top corner $[Xmin, Ymin]$ and the right bottom corner $[Xmax, Ymax]$. If the global variable *Glib_math_coordinate* is set to 1, mathematical coordinate system will be employed, i.e., the left top corner will have the coordinate $[Xmin, Ymax]$.

Example:

```
glib_window(-1,-1,10,10);
```

17 OpenXM-Contrib 一般函数

17.1 函数一覧

17.1.1 ox_check_errors2

ox_check_errors2(*p*)
:: 識別番号 p のサーバのスタック上にあるエラーオブジェクトをリストで戻す.

return リスト

p 数

- 識別番号 p のサーバのスタック上にあるエラーオブジェクトをリストで戻す.
- エラーオブジェクトのポップはしない.

```
[219] P=sm1.start();
0
[220] sm1.sm1(P," 0 get ");
0
[221] ox_check_errors2(P);
[error([7,4294967295,executeString: Usage:get])]
Error on the server of the process number = 1
To clean the stack of the ox server,
type in ox_pops(P,N) (P: process number, N: the number of data you need to pop)
out of the debug mode.
If you like to automatically clean data on the server stack,
set XM_debug=0;
```

18 OXshell の関数

OXshell はシステムのコマンドを ox server より実行する仕組みである. 詳しくは OpenXM/src/kan96xx/Doc/oxshell.oxw および OpenXM/doc/Papers/rims-2003-12-16-ja.tex を見よ.

18.0.1 oxshell.get_value

`oxshell.get_value(`*NAME*`,`*V*`)`
: It get the value of the variable *NAME* on the server ox_shell.

Example:

```
oxshell.set_value("abc","Hello world!");
          oxshell.oxshell(["cp", "stringIn://abc", "stringOut://result"]);
          oxshell.get_value("result");
```

References:

`oxshell.oxshell` , `oxshell.set_value`

18.0.2 oxshell.oxshell

`oxshell.oxshell(`*L*`)`
: It executes command L on a ox_shell server. L must be an array. The result is the outputs to stdout and stderr.

Example:

```
oxshell.oxshell(["ls"]);
```

References:

`ox_shell` , `oxshell.set_value` , `oxshell.get_value`

18.0.3 oxshell.set_value

`oxshell.set_value(`*NAME*`,`*V*`)`
: It set the value V to the variable *Name* on the server ox_shell.

Example:

```
oxshell.set_value("abc","Hello world!");
          oxshell.oxshell(["cat", "stringIn://abc"]);
```

References:

`oxshell.oxshell` , `oxshell.get_value`

19 便利な関数

システムの資源にアクセスするためおよび文字列処理の便利な関数を集めてある.

19.0.1 util_filter

`util_filter(`*Command*`,`*Input*`)`
: It executes the filter program *Command* with the *Input* and returns the output of the filter as a string.

`util_filter(`*Command*`,`*Input* `|` *env*`=key0)`
: This function allows optional variables *env*

Example:

```
util_filter("sort","cat\ndog\ncentipede\n");
```

19.0.2 util_find_and_replace

`util_find_and_replace(`*W*`,`*S*`,`*Wnew*`)`
: It replaces W in S by *Wnew*. Arguments must be a list of ascii codes.

19.0.3 util_find_substr

`util_find_substr(`*W*`,`*S*`)`
: It returns the position of W in S. If W cannot be found, it returns -1. Arguments must be a list of ascii codes.

19.0.4 util_index

`util_index(`*V*`)`
: It returns the name part and the index part of V.

Example:

```
util_index(x_2_3)
```

References:

`util_v`

19.0.5 util_load_file_as_a_string

`util_load_file_as_a_string(`*F*`)`
: It reads a file F as a string.

19.0.6 util_part

`util_part(`*S*`,`*P*`,`*Q*`)`
: It returns from Pth element to Qth element of S.

19.0.7 util_read_file_as_a_string

`util_read_file_as_a_string(`*F*`)`
: It reads a file F as a string.

19.0.8 util_remove_cr

`util_remove_cr(`*S*`)`
: It removes cr/lf/tabs from S. Arguments must be a list of ascii codes.

19.0.9 util_timing

`util_timing(`*Q*`)`
: Show the timing data to execute Q.

Example:

```
util_timing( quote( fctr(x^50-y^50) ));
```

19.0.10 util_v

`util_v(`*V*`,`*L*`)`
: It returns a variable indexed by L.

Example:

```
util_v("x",[1,3]);
```

References:

`util_index`

19.0.11 util_write_string_to_a_file

`util_write_string_to_a_file(`*Fname*`,`*S*`)`
: It writes a string S to a file *Fname*.

20 その他のマニュアル

この節では asir-contrib のその他のマニュアルを紹介する.

それからまだ分類がおわっていない関数を解説する. これらの関数は将来は別の独立した節へ移す予定である.

20.0.1 [[dsolv]] (Solving the initial ideal for holonomic systems)

dsolv

20.0.2 [[ok_diff]] (Okutani's library for differential operators)

ok_diff

20.0.3 [[ok_dmodule]] (Okutani's library for D-modules)

ok_dmodule

20.0.4 [[plucker]] (Plucker relations)

plucker

20.0.5 [[pfpcoh]] (Ohara's library for homology/cohomology groups for p F q)

pfpcoh

20.0.6 [[gnuplot]] (gnuplot ox server for graphics)

gnuplot

20.0.7 [[mathematica]] (Mathematica (TM) ox server)

mathematica

20.0.8 [[om]] (om (java) ox server for translating CMO and OpenMath)

om

20.0.9 [[phc]] (PHC ox server for solving systems of algebraic equations by the homotopy method)

phc

20.0.10 [[sm1]] (Kan/sm1 ox server for the ring of differential operators)

sm1

20.0.11 [[tigers]] (tigers ox server for toric universal Grobner bases)

tigers

20.0.12 [[f_res]] (Comuting resultant)

f_res

20.0.13 [[mt_graph]] (3D grapher)

mt_graph (oxmgraph)

20.0.14 [[todo_parametrize]]

todo_parametrize

パッケージ todo_parametrize/todo_parametrize.rr をロードすることにより, 有理曲線のパラメータ表示を見付ける関数である, paramerize が利用できるようになる. 詳しくは See section “概要” in *Risa/Asir 代数曲線論用パッケージ説明書* を見よ (Web 版 Risa/Asir 代数曲線論用パッケージ説明書 (http://www.math.kobe-u.ac.jp/OpenXM/Current/doc/asir-contrib/html-ja/todo_parametrize/todo_parametrize_ja_toc.html)). このパッケージのマニュアルへの統合はまだできていない. このパッケージはまだ module 構造を利用していないので, 既存のライブラリと名前の衝突の可能性がある.

```
[1205] load("todo_parametrize/todo_parametrize.rr");
1
[1425] parametrize(y^2-x^3);
[155*t^2+20*t+1,720*t^4+1044*t^3+580*t^2,155*t^4+20*t^3+t^2,(-x)/(y)]
[1426] parametrize(y^2+x^3);
[-t,1,t^3,(-x)/(y)]
```

20.0.15 [[taji_alc]]

taji_alc

パッケージ taji_alc.rr をロードすることにより, 一変数代数的コホモロジ群に関連する関数をロードできる. (Web 版 Risa/Asir 一変数代数的局所コホモロジー類に関する Risa/Aisr パッケージ説明書 (http://www.math.kobe-u.ac.jp/OpenXM/Current/doc/asir-contrib/ja/taji_alc-html/taji_alc-ja_toc.html)). このパッケージの asir-contrib マニュアルへの統合はまだできていない.

```
import("taji_alc.rr");
taji_alc.laurent_expansion(x,(x-1)^3);
```

20.0.16 Texinfo でないマニュアル, 論文等.

OpenXM documents (`http://www.math.kobe-u.ac.jp/OpenXM/Current/doc/index-doc-ja.html`). には texinfo で書かれていない asir-contrib のファイルや関数のマニュアル, および関連論文へのリンクがある.

(yang, fj_curve, nk_mora 等)

索引

(Index is nonexistent)

(Index is nonexistent)

Short Contents

1 はじめに 1
2 Asir/Contrib のロード方法. 3
3 Asir Contrib の函数名について 4
4 Windows 版 Asir-contrib 5
5 基礎 (標準函数) 6
6 数 (標準数学函数) 9
7 微積分 (標準数学函数) 11
8 級数 (標準数学函数) 12
9 特殊函数 (標準数学函数) 13
10 行列 (標準数学函数) 14
11 Graphic(標準数学函数) 17
12 表示 (標準数学函数) 18
13 多項式 (標準数学函数) 22
14 複体 (標準数学函数) 26
15 グラフィックライブラリ (2 次元) 27
16 Graphic Library (2 dimensional) 28
17 OpenXM-Contrib 一般函数 31
18 OXshell の関数 32
19 便利な関数 33
20 その他のマニュアル 35
索引 38

Table of Contents

1 はじめに 1

2 Asir/Contrib のロード方法. 3

3 Asir Contrib の函数名について 4

4 Windows 版 Asir-contrib 5

5 基礎 (標準函数) 6

5.0.1 base_cancel 6
5.0.2 base_choose 6
5.0.3 base_flatten 6
5.0.4 base_intersection 6
5.0.5 base_memberq 6
5.0.6 base_permutation 6
5.0.7 base_position 7
5.0.8 base_prune 7
5.0.9 base_replace 7
5.0.10 base_set_minus 7
5.0.11 base_set_union 7
5.0.12 base_subsetq 7
5.0.13 base_subsets_of_size 8

6 数 (標準数学函数) 9

6.0.1 number_abs 9
6.0.2 number_ceiling 9
6.0.3 number_factor 9
6.0.4 number_floor 9
6.0.5 number_imaginary_part 9
6.0.6 number_is_integer 9
6.0.7 number_real_part 9

7 微積分 (標準数学函数) 11

8 級数 (標準数学函数) 12

9 特殊函数 (標準数学函数) 13

10 行列 (標準数学函数) 14

10.0.1 matrix_clone 14
10.0.2 matrix_det 14
10.0.3 matrix_diagonal_matrix 14
10.0.4 matrix_eigenavalues 14
10.0.5 matrix_identity_matrix 14
10.0.6 matrix_image 14
10.0.7 matrix_inner_product 15
10.0.8 matrix_inverse 15
10.0.9 matrix_kernel 15
10.0.10 matrix_list_to_matrix 15
10.0.11 matrix_matrix_to_list 15
10.0.12 matrix_rank 16
10.0.13 matrix_solve_linear 16
10.0.14 matrix_submatrix 16
10.0.15 matrix_transpose 16

11 Graphic(標準数学函数) 17

12 表示 (標準数学函数) 18

12.0.1 print_dvi_form 18
12.0.2 print_em 18
12.0.3 print_gif_form 18
12.0.4 print_input_form 18
12.0.5 print_open_math_tfb_form 18
12.0.6 print_open_math_xml_form 19
12.0.7 print_output 19
12.0.8 print_ox_rfc100_xml_form 19
12.0.9 print_png_form 19
12.0.10 print_terminal_form 20
12.0.11 print_tex_form 20
12.0.12 print_tfb_form 20
12.0.13 print_xdvi_form 20
12.0.14 print_xv_form 21

13 多項式 (標準数学函数) 22
13.0.1 poly_degree 22
13.0.2 poly_elimination_ideal 22
13.0.3 poly_expand 22
13.0.4 poly_factor 22
13.0.5 poly_gcd 23
13.0.6 poly_grobner_basis 23
13.0.7 poly_hilbert_polynomial 23
13.0.8 poly_initial 23
13.0.9 poly_initial_coefficients 24
13.0.10 poly_initial_term 24
13.0.11 poly_solve_linear 24
13.0.12 poly_sort 24

14 複体 (標準数学函数) 26

15 グラフィックライブラリ (2 次元) 27

16 Graphic Library (2 dimensional) 28
16.0.1 glib_clear 28
16.0.2 glib_flush 28
16.0.3 glib_line 28
16.0.4 glib_open 28
16.0.5 glib_plot 28
16.0.6 glib_print 28
16.0.7 glib_ps_form 29
16.0.8 glib_putpixel 29
16.0.9 glib_remove_last 29
16.0.10 glib_set_pixel_size 29
16.0.11 glib_tops 29
16.0.12 glib_window 30

17 OpenXM-Contrib 一般函数 31
17.1 函数一覧 31
17.1.1 ox_check_errors2 31

18 OXshell の関数 32
18.0.1 oxshell.get_value 32
18.0.2 oxshell.oxshell 32
18.0.3 oxshell.set_value 32

19 便利な関数 33

19.0.1 util_filter 33
19.0.2 util_find_and_replace 33
19.0.3 util_find_substr 33
19.0.4 util_index 33
19.0.5 util_load_file_as_a_string 33
19.0.6 util_part 33
19.0.7 util_read_file_as_a_string 33
19.0.8 util_remove_cr 34
19.0.9 util_timing 34
19.0.10 util_v 34
19.0.11 util_write_string_to_a_file 34

20 その他のマニュアル 35

20.0.1 [[dsolv]] (Solving the initial ideal for holonomic systems) 35
20.0.2 [[ok_diff]] (Okutani's library for differential operators) 35
20.0.3 [[ok_dmodule]] (Okutani's library for D-modules) 35
20.0.4 [[plucker]] (Plucker relations) 35
20.0.5 [[pfpcoh]] (Ohara's library for homology/cohomology groups for p F q) 35
20.0.6 [[gnuplot]] (gnuplot ox server for graphics) 35
20.0.7 [[mathematica]] (Mathematica (TM) ox server) . . 35
20.0.8 [[om]] (om (java) ox server for translating CMO and OpenMath) 35
20.0.9 [[phc]] (PHC ox server for solving systems of algebraic equations by the homotopy method) 35
20.0.10 [[sm1]] (Kan/sm1 ox server for the ring of differential operators) 35
20.0.11 [[tigers]] (tigers ox server for toric universal Grobner bases) 36
20.0.12 [[f_res]] (Comuting resultant) 36
20.0.13 [[mt_graph]] (3D grapher) 36
20.0.14 [[todo_parametrize]] 36
20.0.15 [[taji_alc]] 36
20.0.16 Texinifo でないマニュアル, 論文等 36

索引 38