x	in	values at which to evaluate the Gaussian Allowed:double Default: no default
y	in	values at which to evaluate the Gaussian Allowed:numeric Default: no default
height	in	maximum value of the Gaussian Allowed:double Default: 1.0
center	in	location of the maximum value of the Gaussian Allowed:double vector Default: [0.0,0.0]
fwhm	in	width of the gaussian Allowed:double vector Default: [1.0,1.0]/fwhm_to_natural
pa	in	position angle (from y-axis to major-axis) (radians) Allowed:double Default: 0.0

Returns
The "z" values or fail

Example
First evaluate a 2-D Gaussian onto a `[100,100]` array.

```
include 'gaussian.g';
z := array(0, 100, 100);
x := 1:100
for (i in 1:100) {
  y := array(x[i], 100);
  z[,i] := gaussian2d(x, y, center=[50, 50], fwhm=[30, 15], pa=30*pi/180)
}
```

And then show it in a plotter:

```
include 'pgplotter.g';
pg := pgplotter();
pg.env(1, 100, 1, 100, 1, 0);
pg.gray(z, 1, 0, [0, 1, 0, 0, 0, 1])
```

See Also
gaussian1d

1.5.29 matrix - Tool

Package utility
Module mathematics

A collection of matrix/vector routines
include "matrix.g"

Constructors

Functions

angle	Find the angle between two vectors
cosangle	Find the cosine of the angle between two vectors
crossproduct	vector cross-product
diagonalmatrix	make an $n \times n$ diagonal matrix
directproduct	Direct Matrix Product
dotproduct	vector dot-product
help	print a summary of matrix functions
invert	Invert a matrix
mean	the mean of the elements
mult	matrix multiplication
norm	the 'norm' of a vector or matrix
print	print a matrix/vector
rms	the root mean square (rms) of the elements
rotate	rotate a 2D (plain) vector
rotationmatrix	make a 2×2 rotation matrix
symbolic	Make a symbolic matrix/vector
test	a test of most matrix functions
transpose	Transpose a matrix
unitmatrix	make an $n \times n$ unit matrix

Description

The functions in this tool have not been tested as rigorously as other parts of AIPS++. This tool can be considered "alpha" code.
`matrix` is a collection of common matrix and vector operations in Glish. It includes the Direct Matrix Product (used in the AIPS++ Measurement Equation), and a simple matrix inversion which also works with rectangular matrices. The latter should in time be replaced by a C++ executable (DO), which should then implement a choice of more sophisticated techniques than Gauss-Jordan inversion. All operations also work with complex numbers.
NB: Glish itself provides a number of element-by-element vector and array

operations like addition etc. These have not been duplicated here.

Example
The functions are encapsulated in a separate tool, and take matrices and vectors as arguments:

```
- include "matrix.g"
- mx.help();
```

Now, let us do a matrix multiplication:

```
- m1 := array([1:6],2,3);
- print m1;
- m2 := array([1:12],3,4);
- print m2;
- m3 := mx.mult(m1,m2);
- print m3;
```

matrix.unitmatrix - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

make an $n \times n$ unit matrix

Synopsis
unitmatrix(n)

Arguments

n	the length of the diagonal (Input)

---

matrix.diagonalmatrix - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

make an $n \times n$ diagonal matrix

Synopsis
diagonalmatrix(diag)

Arguments

diag	the matrix diagonal (vector) (Input)

---

matrix.rotationmatrix - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

make a 2×2 rotation matrix

Synopsis
rotationmatrix(angle, deg)

Arguments

angle	The rotation angle ϕ (rad) (Input)
deg	If T, the angle is in degrees (Input) Default: F

Description
A 2×2 rotation matrix can be used to rotate a 2D vector over the given angle ϕ.

---

matrix.rotate - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

rotate a 2D (plain) vector

Synopsis
rotate(v, angle, deg)

Arguments

v	The 2x2 vector to be rotated (Input)
angle	The rotation angle ϕ (rad) (Input)
deg	If T, the angle is in degrees (Input) Default: F

Returns
rotated vector

Description
Rotates the 2D vector v over the given angle ϕ.

---

matrix.mult - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

matrix multiplication

Synopsis
mult(m1, m2)

Arguments

m1	The first $(m \times k)$ matrix (Input)
m2	The second $(k \times n)$ matrix (Input)

Returns
an $m \times n$ matrix

Description
Returns the product of the input matrices. Glish vectors are interpreted as matrices with one of their dimensions equal to one.

---

matrix.directproduct - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

Direct Matrix Product

Synopsis
directproduct(m1, m2)

Arguments

m1	The first $(n \times m)$ matrix (Input)
m2	The second $(p \times q)$ matrix (Input)

Returns
an $(n.p \times m.q)$ matrix

Description
Returns the Direct Matrix Product (or tensor product, or Kronecker product) of two rectangular matrices of arbitrary shape. It is used in the AIPS++ Measurement Equation, where the (4×4) matrix that describes the instrumental effects of an interferometer is the Direct Product of two (2×2) antenna 'Jones' matrices.

---

matrix.crossproduct - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

vector cross-product

Synopsis
crossproduct(v1, v2)

Arguments

v1	The first 3D vector (Input)
v2	The second 3D vector (Input)

Returns
vector

Description
Returns a vector which is the cross-product (or outer product) of two 3D input vectors. The output vector is perpendicular to the plane of the two input vectors.

---

matrix.dotproduct - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

vector dot-product

Synopsis
dotproduct(v1, v2)

Arguments

v1	The first vector (Input)
v2	The second vector (Input)

Returns
scalar

Description
Returns the dot-product (or inner product) of two vectors of the same length. The result is proportional to the cosine of the angle between them (see also the functions .angle() and .cosangle() below).

---

matrix.angle - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

Find the angle between two vectors

Synopsis
angle(v1, v2, deg)

Arguments

v1	The first vector (Input)
v2	The second vector (Input)
deg	If T, the result is in degrees (Input) Default: F

Returns
angle

Description
Returns the angle (in rad or deg) between the two input vectors, which must be of equal length.

---

matrix.cosangle - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

Find the cosine of the angle between two vectors

Synopsis
cosangle(v1, v2)

Arguments

v1	The first vector (Input)
v2	The second vector (Input)

Returns
angle

Description
Returns the cosine of the angle between the two input vectors, which must of course be of equal length.

---

matrix.transpose - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

Transpose a matrix

Synopsis
transpose(m)

Arguments

m	The given $n \times m$ matrix (Input)

Returns
An $m \times n$ matrix

---

matrix.invert - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

Invert a matrix

Synopsis

invert(m)

Arguments

m	The matrix to be inverted (Input)

Returns
matrix

Description
The inversion function is Gauss-Jordan substitution with full pivoting. The input matrix may be rectangular. If the number of rows (equations) is greater then the number of columns (unknowns), the inverted matrix represents a least-squares fit. The input matrix is not affected. If the input matrix is singular, a fail is returned.

---

matrix.norm - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

the 'norm' of a vector or matrix

Synopsis
norm(m)

Arguments

m	array (Input)

Description
Returns the square root of the sum of the elements. For a vector this is called the norm, i.e. the square of its length.

---

matrix.mean - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

the mean of the elements

Synopsis
mean(m)

Arguments

m	array (Input)

---

matrix.rms - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

the root mean square (rms) of the elements

Synopsis
rms(m)

Arguments

m	array (Input)

Description
Returns the rms of the elements (w.r.t. the mean).

---

matrix.symbolic - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

Make a symbolic matrix/vector

Synopsis
symbolic(name, nrow, ncol)

Arguments

name	The 'root' name of an element (Input)
nrow	The nr of rows (Input)
ncol	Optional: The nr of columns (Input) Default: F

Returns
a string matrix or vector

Description
This function looks ahead to the possibility of symbolic computation in Glish. It returns a $nrow \times ncol$ matrix with string elements of the form 'a12' etc. If the ncol argument is omitted, the result is a string vector: ['a1','a2','a3','a4'].

---

matrix.print - Function

Package	**utility**
Module	**mathematics**
Tool	**matrix**

print a matrix/vector

Synopsis
print(m, name, prec)

Arguments

m	The vector/matrix to be printed (Input)
name	Optional: a header text (Input) Default: ' '
prec	Optional: the printing precision (Input) Default: 0

Description
Prints the matrix m in an organised way. Useful for seeing what is going on. Not very well developed at the moment.
