関数名	dmat_add
概略	行列の和を計算する
ヘッダ	matrix.h
書式	double **dmat_add(double **mat1, double **mat2, double **mat3, int m, int n)
機能説明	mat1 := mat2 + mat3
引き数	mat1,mat2,mat3: 行列へのポインタ m:行数 n:列数
戻り値	mat1 へのポインタ

関数名	dmat_alloc
概略	行列の領域を確保する
ヘッダ	matrix.h
書式	double **dmat_alloc(int m, int n);
機能説明	double 型で m 行 n 列の領域を確保し, すべての要素を 0 に初期化する.
引き数	m:行数 n:列数
戻り値	行列へのポインタ
例	double **mat;　　/* 変数の宣言 */ mat = dmat_alloc(2,3);　/* 2 行 3 列の領域確保 */ /* Calculation */ dmat_free(mat,2,3);　　/* 領域の解放 */

関数名	dmat_clear
概略	行列を零行列にする
ヘッダ	matrix.h
書式	double **dmat_clear(double **mat, int m, int n)
機能説明	mat := 0
戻り値	mat へのポインタ

関数名	dmat_col_swap
概略	列要素を入れ替える
ヘッダ	matrix.h
書式	double **dmat_col_swap(double **mat, int m, int j1, int j2)
機能説明	mat の i 列と j 列を入れ替える. すなわち mat[i][j1] <-> mat[i][j2]
戻り値	mat へのポインタ

関数名	dmat_copy
概略	行列の要素を複写する
ヘッダ	matrix.h
書式	double **copy(double **mat1, double **mat2, int m, int n)
機能説明	mat1 := mat2
戻り値	mat1 へのポインタ

関数名	dmat_free
概略	行列の領域を解放する
ヘッダ	matrix.h
書式	void dmat_free(double **mat, int m, int n)
機能説明	

戻り値	なし

関数名	dmat_identity
概略	行列に単位行列を代入する
ヘッダ	matrix.h
書式	double **dmat_identity(double **mat, int n)
機能説明	mat[i][i]:=1
引き数	mat: 一次元配列へのポインタ n: 行列のサイズ
戻り値	mat へのポインタ

関数名	dmat_mul
概略	行列の積を求める
ヘッダ	matrix.h
書式	double **dmat_mul(double **mat1, double **mat2, double **mat3, int m, int p, int n)
機能説明	mat1 := mat2 * mat3
戻り値	mat1 へのポインタ

関数名	dmat_print
概略	行列を表示する
ヘッダ	matrix.h
書式	void dmat_print(double **mat, int m, int n)
機能説明	行列を表示する
戻り値	なし

関数名	dmat_sub
概略	行列の差を求める
ヘッダ	matrix.h
書式	double **dmat_sub(double **mat1, double **mat2, double **mat3, int m, int n)
機能説明	mat1 := mat2 + mat3
引き数	mat1, mat2, mat3: 行列へのポインタ m: 行数 n: 列数
戻り値	mat1 へのポインタ

関数名	dmat_mul_scalar
概略	行列をスカラー倍する
ヘッダ	matrix.h
書式	double **dmat_mul_scalar(double **mat, double c, int m, int n)
機能説明	mat := c * mat
引き数	mat: 行列　c: 実数倍　m: 行数　n: 列数
戻り値	mat へのポインタ

関数名	dmat_mul_aaT
概略	行列の積
ヘッダ	matrix.h
書式	double **dmat_mul_aaT(double **aaT, double **a, int m, int n)

機能説明	aaT[0..m-1][0..m-1]に a[0..m-1][0..n-1]の積を代入する aaT := a * aT
引き数	aaT: 2 次元配列へのポインタ a: 行列へのポインタ m:行数, n:列数
戻り値	aaT へのポインタ

関数名	dmat_mul_aTa
概略	行列の積
ヘッダ	matrix.h
書式	double **dmat_mul_aTa(double **aTa, double **a, int m, int n)
機能説明	aTa[0..n-1][0..n-1]に a[0..m-1][0..n-1]の積を代入する aTa := aT * a
引き数	aTa: 2 次元配列へのポインタ a: 行列へのポインタ m:行数, n:列数
戻り値	aTa へのポインタ

関数名	dmat_mul_aTb
概略	行列とベクトルの積
ヘッダ	matrix.h
書式	double *dmat_mul_aTb(double *aTb, double **a, double *b, int m, int n)
機能説明	aTb := a * b
引き数	aTb, b: 一次元配列へのポインタ a: 行列へのポインタ m:行数, n:列数
戻り値	a へのポインタ

関数名	dmat_mul_vector
概略	行列とベクトルの積
ヘッダ	matrix.h
書式	double *dmat_mul_vector(double *a, double **mat, double *b, int m, int n)
機能説明	a := mat * b
引き数	a,b: 一次元配列へのポインタ mat: 行列へのポインタ m:行数, n:列数
戻り値	a へのポインタ

関数名	dmat_norm
概略	行列の 2 乗ノルムを求める
ヘッダ	matrix.h
書式	double dmat_norm(double **mat, int m, int n)
機能説明	
戻り値	2 乗ノルム

関数名	dmat_row_swap
概略	ある行の中から絶対値最大の要素位置を求める
ヘッダ	matrix.h
書式	double **dmat_row_swap(double **mat, int i1, int i2, int n)

機能説明	mat[i1][j] <-> mat[i2][j] (j=0..n-1)
戻り値	列を返す

関数名	dmat_row_abs_max_num
概略	ある行の中から絶対値最大の要素位置を求める
ヘッダ	matrix.h
書式	int dmat_row_abs_max_num(double **mat, int i, int n)
機能説明	
戻り値	列を返す

関数名	dmat_trace
概略	行列のトレースを求める
ヘッダ	matrix.h
書式	double dmat_trace(double **mat, int n)
機能説明	mat[0..n-1][0..n-1]のトレースを求める. すなわち mat[0][0]+..+mat[n-1][n-1]
引き数	mat: 行列へのポインタ n: 行列のサイズ
戻り値	トレース値

関数名	dmat_trans
概略	行列の転置を求める
ヘッダ	matrix.h
書式	double **dmat_trans(double **mat1, double **mat2, int m, int n)
機能説明	mat[i][j] := mat[j][i] (i=0..m-1, j=0..n-1)
戻り値	mat1 へのポインタ

関数名	dvec_inner_product
概略	ベクトルの内積を求める
ヘッダ	matrix.h
書式	double dvec_inner_product(double *v1, double *v2, int n)
機能説明	v1[0..n-1]と v2[0..n-1]の内積を求める
引き数	v1, v2: 一次元配列へのポインタ n: ベクトルのサイズ
戻り値	内積値

```
#define dmat_vector(v1,mat,v2,m,n)  dmat_vector_mul(v1,mat,v2,m,n)
double **dmat_submat_assign(double **mat1, double **mat2, int sr, int sc, int m, int n);
double **dmat_submat_scalar_assign(double **mat1, double **mat2, double c, int sr, int sc, int m,
int n);
```

関数名	dmat_rx
概略	回転行列
ヘッダ	matrot.h
書式	double **dmat_rx(double **mat, double rad)
機能説明	x 軸まわりに角度 rad の回転行列を求める
引き数	mat: 2 次元配列へのポインタ rad: x 軸まわりの回転角度[単位 rad]
戻り値	mat へのポインタ

関数名	dmat_ry
概略	回転行列
ヘッダ	matrot.h
書式	double **dmat_ry(double **mat, double rad)
機能説明	y 軸まわりに角度 rad の回転行列を求める
引き数	mat: 2 次元配列へのポインタ rad: y 軸まわりの回転角度[単位 rad]
戻り値	mat へのポインタ

関数名	dmat_rz
概略	回転行列
ヘッダ	matrot.h
書式	double **dmat_rz(double **mat, double rad)
機能説明	z 軸まわりに角度 rad の回転行列を求める
引き数	mat: 2 次元配列へのポインタ rad: z 軸まわりの回転角度[単位 rad]
戻り値	mat へのポインタ

関数名	dmat_is_rot
概略	回転行列か否かを評価する
ヘッダ	matrot.h
書式	int dmat_is_rot(double **mat, int n, double eps)
機能説明	
引き数	mat: 2 次元配列へのポインタ n: 行列のサイズ eps: 評価の閾値
戻り値	TRUE / FALSE

関数名	dmat_is_orthogonal
概略	直行行列か否かを評価する
ヘッダ	matrot.h
書式	int dmat_is_orthogonal(double **u, int m, int n, double eps)
機能説明	
引き数	mat: 2 次元配列へのポインタ n: 行サイズ n: 列サイズ eps: 評価の閾値
戻り値	TRUE / FALSE

関数名	dmat_is_identity
概略	単位行列か否かを評価する
ヘッダ	matrot.h
書式	int dmat_is_identity(double **mat, int n, double eps)
機能説明	
引き数	mat: 2 次元配列へのポインタ n: 行列サイズ eps: 評価の閾値
戻り値	TRUE / FALSE

関数名	dmat_from_RollPitchYaw
概略	ロール・ピッチ・ヨー角から回転行列を求める
ヘッダ	matrot.h
書式	double **dmat_from_RollPitchYaw(double **mat, double *rad)
機能説明	
引き数	rad: ロール・ピッチ・ヨー角[rad] mat: 3 行 3 列の 2 次元配列へのポインタ
戻り値	mat へのポインタ

関数名	dmat_to_RollPitchYaw
概略	回転行列からロール・ピッチ・ヨー角を求める
ヘッダ	matrot.h
書式	double *dmat_to_RollPitchYaw(double *rad, double **mat)
機能説明	
引き数	rad: ロール・ピッチ・ヨー角[rad] mat: 3 行 3 列の 2 次元配列へのポインタ
戻り値	rad へのポインタ

関数名	dmat_to_AxisAngle
概略	回転行列から回転軸とその周りの回転角を求める
ヘッダ	matrot.h
書式	double *dmat_to_AxisAngle(double *rad, double **mat)
機能説明	
引き数	rad: ロール・ピッチ・ヨー角[rad] mat: 3 行 3 列の 2 次元配列へのポインタ
戻り値	rad へのポインタ

関数名	dmat_from_AxisAngle
概略	回転軸とその周りの回転角から回転行列を求める
ヘッダ	matrot.h
書式	double **dmat_from_AxisAngle(double **mat, double *rad)
機能説明	
引き数	rad: ロール・ピッチ・ヨー角[rad] mat: 3 行 3 列の 2 次元配列へのポインタ
戻り値	mat へのポインタ