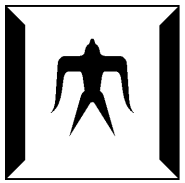
Matlib Reference Manual



東京工業大学　古賀 雅伸

© Copyright 1989 – 1995 Masanobu KOGA
All Rights Reserved.

No part of this publication may be reproduced, transmitted, or translated, in any form or by any means, electronic, mechanical, manual, optical or otherwise, without prior permission of Masanobu Koga.

© 著作権　古賀 雅伸　1989 – 1995

本説明書の一部または全部を古賀雅伸の許可なく，複写，複製することは，その形態・手段を問わず禁じます。

Contents

1 ユーティリティ **9**

MatInit . . . 9
MatCopy . . . 11
MatDup . . . 12
MatSwap . . . 13
MatAssign . . . 14
MatChangeColumn . . . 15
MatChangeRow . . . 16
MatFileRead . . . 17
MatFileWrite . . . 18
MatInput . . . 19
MatEdit . . . 20
MatSearch . . . 21
MatPrint . . . 22
MatAllPrint . . . 23
MatFree . . . 24
MatFrees . . . 25
MatAllFree . . . 26
MatDestroy . . . 27
MatDestroys . . . 28
MatAllDestroy . . . 29
MatTmpDestroy . . . 30
MatError . . . 31

MatRows . . . 32
MatCols . . . 33

2 実行列メモリ割り付け **35**

MatDef . . . 35
MatNonNameDef . . . 37
MatDiagDef . . . 38
MatVander . . . 39
MatIDef . . . 40
MatIDef2 . . . 41
MatOneDef . . . 42
MatFillDef . . . 43
MatZDef . . . 44
MatZDef2 . . . 45
MatSameDef . . . 46
MatSameClassDef . . . 47
MatTransDef . . . 48
MatMulDef . . . 49
MatEigValDef . . . 50
MatEigVecDef . . . 51
MatColumnVec . . . 52
MatRowVec . . . 53

3 高レベル行列演算 **55**

MatAdd . . . 55
MatSub . . . 57
MatMul . . . 58
MatMulElem . . . 59
MatInv . . . 60
MatInvElem . . . 61
MatPseudoInv . . . 62
MatPow . . . 63

MatScale . . . 64
MatScaleC . . . 65
MatScaleP . . . 66
MatScaleR . . . 67
MatTrans . . . 68
MatFlipLR . . . 69
MatFlipUD . . . 70
MatNegate . . . 71
MatExp . . . 72
MatConj . . . 73
MatConjTrans . . . 74
MatVectorProduct . . . 75
MatScalarProduct . . . 76
C_MatScalarProduct . . . 77
MatVecToDiag . . . 78
MatDiagToVec . . . 79
MatEig . . . 80
MatHermitEig . . . 81
MatEigVal . . . 82
MatEigVec . . . 83
MatEigValVec . . . 84
MatSingVal . . . 85
MatSingVec . . . 86
MatSingValVec . . . 87
MatSingLeftVec . . . 88
MatSingRightVec . . . 89
MatSqrt . . . 90
MatDeterm . . . 91
MatFrobNorm . . . 92
MatInfNorm . . . 93
MatTrace . . . 94
MatMaxElem . . . 95

MatMinElem . . . 96
MatSumElem . . . 97
MatMeanElem . . . 98
MatMaxSingVal . . . 99
MatMinSingVal . . . 100
MatGetValue . . . 101
MatSetValue . . . 102
MatGetPtr . . . 103
MatRank . . . 104
MatKernel . . . 105

4 行列の合成・分解・変換 107
MatCatColumn . . . 107
MatCatColumns . . . 109
MatCatRow . . . 110
MatCatRows . . . 111
MatCat4 . . . 112
MatCut . . . 113
MatColumnVector . . . 114
MatRowVector . . . 115
MatPut . . . 116
MatRealPart . . . 117
MatImagPart . . . 118
MatRealAndImag . . . 119
MatRealToComp . . . 120
MatRealToPoly . . . 121
MatRealToRat . . . 122
MatCompToPoly . . . 123
MatCompToRat . . . 124
MatPolyToRat . . . 125
MatBlockDiag . . . 126
MatCompanion . . . 127

MatCompanion2 128

5 行列の特性 129

MatGetClass 129
MatSetClass 131
MatGetType 132
MatSetType 133
MatGetName 134
MatSetName 135
MatRows 136
MatCols 137
MatIsReal 138
MatIsComplex 139
MatIsVar 140
MatIsTmp 141
MatIsFullRank 142
MatIsSingular 143
MatIsNonSingular 144
MatIsSquare 145
MatIsNonSquare 146
MatIsPositive 147
MatIsPositiveSemi 148
MatIsNonPositive 149
MatIsNegative 150
MatIsNegativeSemi 151
MatIsNonNegative 152
MatIsZero 153
MatIsNonZero 154
MatIsEqual 155
MatIsNonEqual 156
MatIsColumnVec 157
MatIsRowVec 158

6 低レベル実行列演算 159

Mat_Add 159
Mat_Sub 161
Mat_Mul 162
Mat_MulElem 163
Mat_Inv 164
Mat_InvElem 165
Mat_PseudoInv 166
Mat_Pow 167
Mat_Scale 168
Mat_Trans 169
Mat_FlipLR 170
Mat_FlipUD 171
Mat_Negate 172
Mat_CatColumn 173
Mat_CatRow 174
Mat_Cat4 175
Mat_Cut 176
Mat_Put 177
Mat_VectorProduct 178
Mat_Eig 179
Mat_EigVal 180
Mat_EigVec 181
Mat_EigValVec 182
Mat_SingVal 183

7 複素行列ユーティリティー 185

C_MatInput 185
C_MatCopy 187
C_MatRead 188
C_MatPrint 189
C_MatSwap 190

8 複素行列メモリ割り付け **191**
C_MatDef . . . 191
C_MatNonNameDef . . . 193
C_MatDiagDef . . . 194
C_MatVander . . . 195
C_MatIDef . . . 196
C_MatIDef2 . . . 197
C_MatOneDef . . . 198
C_MatFillDef . . . 199
C_MatZDef . . . 200
C_MatZDef2 . . . 201
C_MatSameDef . . . 202
C_MatTransDef . . . 203
C_MatMulDef . . . 204
C_MatColumnVec . . . 205
C_MatRowVec . . . 206

9 高レベル複素行列演算 **207**
C_MatDeterm . . . 207
C_MatTrace . . . 209
C_MatMaxElem . . . 210
C_MatMinElem . . . 211
C_MatSumElem . . . 212
C_MatMeanElem . . . 213
C_MatGetValue . . . 214
C_MatSetValue . . . 215
C_MatGetPtr . . . 216
C_MatRealPart . . . 217
C_MatImagPart . . . 218

10 低レベル複素行列演算 **219**
C_Mat_Add . . . 219

C_Mat_Sub . . . 221
C_Mat_Mul . . . 222
C_Mat_MulElem . . . 223
C_Mat_Inv . . . 224
C_Mat_InvElem . . . 225
C_Mat_PseudoInv . . . 226
C_Mat_Pow . . . 227
C_Mat_Scale . . . 228
C_Mat_Scale2 . . . 229
C_Mat_Trans . . . 230
C_Mat_FlipLR . . . 231
C_Mat_FlipUD . . . 232
C_Mat_Conj . . . 233
C_Mat_ConjTrans . . . 234
C_Mat_Negate . . . 235
C_Mat_Cut . . . 236
C_Mat_Put . . . 237
C_Mat_VectorProduct . . . 238
C_Mat_Eig . . . 239
C_Mat_HermitEig . . . 240
C_Mat_EigVal . . . 241
C_Mat_EigVec . . . 242
C_Mat_EigValVec . . . 243
C_Mat_SingVal . . . 245
C_Mat_CatColumn . . . 246
C_Mat_CatRow . . . 247
C_Mat_RealPart . . . 248
C_Mat_ImagPart . . . 249
C_Mat_RealAndImag . . . 250
C_Mat_RealToComp . . . 251

索 引 **253**

Chapter 1

ユーティリティ

Name

MatInit — 行列ライブラリの初期化

Synopsis

void MatInit()

Arguments

なし

Description

行列ライブラリの初期化を行なう。

Return

なし

See Also

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatPrint(A);
}
```

Name

MatCopy — 行列の成分のコピー

Synopsis

```
Matrix    MatCopy(to, from)
Matrix    to, from;
```

Arguments

to	代入先の行列
from	被代入行列

Description

from 行列の成分を to 行列にコピーする。

Return

コピーされた結果の行列

See Also

MatAssign, C_MatCopy

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   MatCopy(B, A);
   MatPrint(B);
}
```


Name

MatDup — 行列の複製

Synopsis

```
Matrix    MatDup(org)
Matrix    org;
```

Arguments

org	複製元の行列

Description

org 行列の複製を作る

Return

複製された行列

See Also

MatCopy

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDup(A);
   MatPrint(B);
}
```

Name

MatSwap — 2 つの同じ大きさの行列の成分の交換

Synopsis

```
void    MatSwap(mat1, matg2)
Matrix  mat1, mat2;
```

Arguments

mat1	成分交換を行なう行列 1
mat2	成分交換を行なう行列 2

Description

mat1 と mat2 の成分を交換する。

Return

なし

See Also

C_MatSwap

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   MatSwap(A, B);
   MatPrint(A);
   MatPrint(B);
}
```


Name

MatAssign — 行列の代入 (行列リストへの登録)

Synopsis

```
Matrix  MatAssign(to, from)
Matrix  to, from;
```

Arguments

to	代入先の行列
from	被代入行列

Description

from 行列を to 行列 に代入し，行列リストに登録する。

Return

登録した行列

See Also

MatCopy

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatIDef(2);
   MatAssign(B, A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatChangeColumn — 行の交換

Synopsis

```
Matrix  MatChangeColumn(mat, col1, col2)
Matrix  mat;
int     col1, col2;
```

Arguments

mat	対象行列
col1	交換する行番号 1
col2	交換する行番号 2

Description

mat 行列の col1 行と col2 行を交換する。

Return

行の交換を行なった行列

See Also

MatChangeRow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatChangeColumn(A, 1, 2);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatChangeRow — 列の交換

Synopsis

```
Matrix  MatChangeRow(mat, row1, row2)
Matrix  mat;
int     row1, row2;
```

Arguments

mat	対象行列
row1	交換する列番号 1
row2	交換する列番号 2

Description

mat 行列の row1 列と row2 列を交換する。

Return

列の交換を行なった行列

See Also

MatChangeColumn

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatChangeRow(A, 1, 2);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatFileRead — ファイルからの読み込み

Synopsis

```
Matrix  MatFileRead(filename)
char    *filename;
```

Arguments

filename	読み込むファイルの名前

Description

ファイルから行列を読み込む。

Return

読み込んだ行列

See Also

MatFileWrite

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatFileRead("A.mat");
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

MatFileWrite — ファイルへの書き込み

Synopsis

```
Matrix  MatFileWrite(mat, filename)
Matrix  mat;
char    *filename;
```

Arguments

mat	書き込む行列の名前
filename	書き込むファイルの名前

Description

ファイルへ行列を書き込む

Return

書き込んだ行列

See Also

MatFileRead

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatFileWrite(A, "A.mat");
}
```

Name

MatInput — キーボードからの行列の入力

Synopsis

```
Matrix    MatInput(name)
char      *name;
```

Arguments

name	入力する行列の名前

Description

キーボードから行列を入力、名前を name とする。

Return

入力された行列

See Also

MatEdit, C_MatInput, C_MatRead

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   MatInput("A");
   MatPrint(A);
}
```


Name

MatEdit — 行列の編集

Synopsis

```
Matrix    MatEdit(mat)
Matrix    mat;
```

Arguments

mat	編集を行なう行列

Description

行列の編集を行なう。

Return

編集された結果の行列

See Also

MatInput

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = MatDef("A", 2, 2);
   MatEdit(A);
   MatPrint(A);
}
```

Name

MatSearch — 行列の名前による検索

Synopsis

```
Matrix  MatSearch(name)
char    *name;
```

Arguments

name	検索する行列の名前

Description

行列を名前で検索する。

Return

検索された行列

See Also

MatGetName, MatName

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatSearch("A");
    MatPrint(B);
}
```


Name

MatPrint — 行列の表示

Synopsis

```
void    MatPrint(mat)
Matrix  mat;
```

Arguments

mat	画面に表示する行列

Description

mat 行列を画面に表示する。

Return

なし

See Also

MatAllPrint, C_MatPrint

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatPrint(A);
}
```

Name

MatAllPrint — すべての行列の表示

Synopsis

void MatAllPrint()

Arguments

なし

Description

行列リストに登録されている行列をすべて表示する。

Return

なし

See Also

MatPrint, C_MatPrint

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    MatAllPrint();
}
```


Name

MatFree — 行列リストからの削除

Synopsis

void MatFree(mat)
Matrix mat;

Arguments

mat 行列リストから削除を行なう行列

Description

mat 行列を行列リストから削除する。

Return

なし

See Also

MatAllFree, MatDestroy

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    MatFree(A);
    MatAllPrint();
}
```

Name

MatFree — 複数の行列の行列リストからの削除

Synopsis

```
void        MatFrees(mat)
Matrix      mat;
```

Arguments

mat	行列リストから削除を行なう先頭行列

Description

mat 行列以降の行列を行列リストから削除する。

Return

なし

See Also

MatFree, MatAllFree, MatDestroy

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    MatFrees(B);
    MatAllPrint();
}
```


Name

MatAllFree — すべての行列の行列リストから削除

Synopsis

```
void        MatAllFree()
```

Arguments

なし

Description

すべての行列を行列リストから削除する。

Return

なし

See Also

MatFree, MatTmpFree, MatAllDestroy

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    MatAllFree();
}
```

Name

MatDestroy — 行列のメモリの解放

Synopsis

```
void    MatDestroy(mat)
Matrix  mat;
```

Arguments

mat	メモリの解放を行なう行列

Description

MatDef() で確保した mat 行列のメモリを解放する。

Return

なし

See Also

MatDef, MatDestroys, MatAllDestroy, MatTmpDestroy, MatFree

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    MatDestroy(A);
    MatAllPrint();
}
```


Name

MatDestroys — 複数の行列のメモリの解放

Synopsis

```
void    MatDestroys(mat)
Matrix  mat;
```

Arguments

mat	メモリの解放を行なう先頭行列

Description

mat 行列以降の行列のメモリを解放する。

Return

なし

See Also

MatDestroy, MatAllDestroy, MatTmpDestroy, MatFree

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    MatDestroys(B);
    MatAllPrint();
}
```

Name

MatAllDestroy — すべての行列メモリの解放

Synopsis

void MatAllDestroy()

Arguments

なし

Description

行列リストに登録されているすべての行列のメモリを解放する。

Return

なし

See Also

MatDestroy, MatTmpDestroy, MatAllFree

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    MatAllDestroy();
}
```


Name

MatTmpDestroy — テンポラリー行列のメモリを解放する。

Synopsis

void MatTmpDestory()

Arguments

なし

Description

テンポラリー行列のメモリを解放する。

Return

なし

See Also

MatDestroy, MatAllDestroy, MatTmpFree

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatMul(MatAdd(A, A), MatSub(A, A));
    MatSetName(B, "B");
    MatTmpDestroy();
    MatAllPrint();
}
```

Name

MatError — エラーメッセージの表示

Synopsis

```
void    MatError(func, stmt, a)
char    *func;
char    *stmt;
Matrix  a;
```

Arguments

func	エラーの発生した関数名
stmt	エラーメッセージ
a	エラーに関係する行列

Description

エラーメッセージを表示する。

Return

なし

See Also

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);

    if (MatIsComplex(A))
        MatError("main()", "Not a real matrix", A);
}
```


Name

MatRows — 行列の行の数

Synopsis

```
void    MatRows(mat)
Matrix  mat;
```

Arguments

mat	行の数を求める行列

Description

行列の数を求める。

Return

行列の行の数

See Also

Rows, Cols, MatCols

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    int rows;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    rows = MatRows(A);
    printf("MatRows(A) = %d“n", rows);
}
```

Name

MatCols — 行列の列の数を返す。

Synopsis

```
void    MatCols(mat)
Matrix  mat;
```

Arguments

mat	列の数を求める行列

Description

行列の列の数を求める。

Return

行列の列の数

See Also

Cols, Rows, MatRows

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    int cols;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    cols = MatCols(A);
    printf("MatCols(A) = %d“n", cols);
}
```

Chapter 2

実行列メモリ割り付け

Name

MatDef — 実行列の名前付きメモリ割り付け

Synopsis

```
Matrix  MatDef(name, rows, cols)
char    *name;
int     rows, cols;
```

Arguments

name	設定する実行列の名前
rows	行の大きさ
cols	列の大きさ

Description

実行列のメモリ割り付けを行ない、名前をつける。ただし，成分の初期化 (ゼロ化) は行なわれない。

Return

メモリ割り付けを行なった行列

See Also

MatNonNameDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatPrint(A);
}
```

Name

MatNonNameDef — 実行列の名前なしメモリ割り付け

Synopsis

```
Matrix  MatNonNameDef(rows, cols)
int     rows, cols;
```

Arguments

rows	行の大きさ
cols	列の大きさ

Description

実行列のメモリ割り付けを行なう。ただし，成分の初期化 (ゼロ化) は行なわれない。

Return

メモリ割り付けを行なった行列

See Also

MatDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatNonNameDef(2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

MatDiagDef — 実対角行列の定義

Synopsis

```
Matrix  MatDiagDef(size, elm1, elm2, ...)
int     size;
double  elm1, elm2, ...;
```

Arguments

size	対角成分の数
elm1	第 (1、 1) 成分
elm2	第 (2、 2) 成分

Description

実対角行列を定義する。

Return

定義された行列

See Also

MatDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDiagDef(3, 1.0, 2.0, 3.0);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

MatVander — 実バンデルモンデ行列の定義

Synopsis

```
Matrix  MatVander(size, elm1, elm2, ...)
int     size;
double  elm1, elm2, ...;
```

Arguments

size	成分の数
elm1	第 1 成分
elm2	第 2 成分

Description

実バンデルモンデ行列を定義する。

Return

定義された行列

See Also

MatDiagDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatVander(3, 1.0, 2.0, 3.0);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

MatIDef — 実単位行列の定義

Synopsis

```
Matrix  MatIDef(size)
int     size;
```

Arguments

size	行列の大きさ

Description

実単位行列を定義する。

Return

定義された行列

See Also

MatDef, MatOneDef, MatZDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatIDef(2);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

MatIDef2 — 対角要素が 1 である実行列の定義

Synopsis

```
Matrix  MatIDef2(column, row)
int     column;
int     row;
```

Arguments

column	列の大きさ
row	行の大きさ

Description

対角要素が 1 である実行列を定義する。

Return

定義された行列

See Also

MatIDef, MatOneDef, MatZDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatIDef2(2, 3);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

MatOneDef — 成分が 1 で満たされた行列の定義

Synopsis

```
Matrix  MatOneDef(column, row)
int     column;
int     row;
```

Arguments

column	列の大きさ
row	行の大きさ

Description

成分が 1 で満たされた行列の定義

Return

定義された行列

See Also

MatIDef, MatOneDef, MatZDef2

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatOneDef(2, 3);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

MatFillDef — 成分が定数で満たされた行列の定義

Synopsis

```
Matrix  MatFillDef(column, row, data)
int     column;
int     row;
double  data;
```

Arguments

column	列の大きさ
row	行の大きさ
data	定数

Description

成分が定数で満たされた行列の定義

Return

定義された行列

See Also

MatIDef2, MatZDef2

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatFillDef2(2, 3, 2.0);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

MatZDef — 正方の実零行列の定義

Synopsis

```
Matrix  MatZDef(size)
int     size;
```

Arguments

size	零行列の大きさ

Description

size × size の実零行列を定義する。

Return

定義された零行列

See Also

MatDef, MatZDef2, MatIDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatZDef(3);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

MatZDef2 — 非正方の実零行列の定義

Synopsis

```
Matrix  MatZDef2(rows, cols)
int     rows, cols;
```

Arguments

rows	rows 行
cols	cols 列

Description

rows × cols の実零行列を定義する。

Return

定義された零行列

See Also

MatDef, MatZDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatZDef2(2, 3);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

MatSameDef — 同じサイズの実行列を格納するためのメモリ割り付け

Synopsis

```
Matrix  MatSameDef(mat)
Matrix  mat;
```

Arguments

mat	サイズの対象となる行列

Description

mat 行列と同じサイズの実行列を格納するためのメモリ割り付けを行なう。

Return

格納領域へ (すべての成分が零)

See Also

MatDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B, C;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatSameDef(A);
    MatCopy(B, A);
    MatSetName(B, "B");
    MatPrint(B);
}
```

Name

MatSameClassDef — 同じクラス行列を格納するためのメモリ割り付け

Synopsis

```
Matrix  MatSameClassDef(mat, rows, cols)
Matrix  mat;
int     rows, cols;
```

Arguments

mat	クラスの対象となる行列
rows	行の大きさ
cols	列の大きさ

Description

mat 行列と同じクラスの行列を格納するためのメモリ割り付けを行なう。

Return

メモリを確保した行列

See Also

MatDef, MatSameDef

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   B = MatSameClassDef(A, 1, 2);
   MatCopy(B, A);
   MatSetName(B, "B"); MatPrint(B);
}
```


Name

MatTransDef — 実転置行列を格納するためのメモリ割り付け

Synopsis

```
Matrix  MatTransDef(mat)
Matrix  mat;
```

Arguments

mat	サイズの対象となる行列

Description

mat 行列の実転置行列と同じサイズのメモリ割り付けを行なう。

Return

格納領域へ

See Also

MatDef

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   B = MatTransDef(A);
   Mat_Trans(B, A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatMulDef — 実行列の乗算結果を格納するためのメモリ割り付け

Synopsis

```
Matrix    MatMulDef(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	乗算を行なう実行列 1
mat2	乗算を行なう実行列 2

Description

実行列の乗算結果を格納するためのメモリ割り付けを行なう。

Return

格納領域へ

See Also

MatDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B, C;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
    MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
    C = MatMulDef(A, B); Mat_Mul(C, A, B);
    MatSetName(C, "C"); MatPrint(C);
}
```


Name

MatEigValDef — 固有値を入れる行列の定義

Synopsis

```
Matrix    MatEigValDef(mat)
Matrix    mat;
```

Arguments

mat	固有値を求める行列

Description

固有値を入れる行列を定義する。

Return

固有値を入れる行列

See Also

MatDef, MatEigVecDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A, val, vec;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    val = MatEigValDef(A);
    vec = MatEigVecDef(A);
    Mat_EigVal(A, &val, &vec);
    MatSetName(val, "eigval(A)"); MatPrint(val);
    MatSetName(vec, "eigvec(A)"); MatPrint(vec);
}
```

Name

MatEigVecDef — 固有ベクトルを入れる行列の定義

Synopsis

```
Matrix  MatEigVecDef(mat)
Matrix  mat;
```

Arguments

mat	固有ベクトルを求める行列

Description

固有ベクトルを格納する行列を定義する。

Return

固有ベクトルを格納する行列

See Also

MatDef, MatEigValDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A, val, vec;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    val = MatEigValDef(A);
    vec = MatEigVecDef(A);
    Mat_EigVal(A, &val, &vec);
    MatSetName(A, "eigval(A)"); MatPrint(val);
    MatSetName(A, "eigvec(A)"); MatPrint(vec);
}
```


Name

MatColumnVec — 実縦ベクトルの定義

Synopsis

```
Matrix  MatColumnVec(size, elm1, elm2, ...)
int     size;
double  elm1, elm2, ...;
```

Arguments

size	ベクトルの大きさ
elm1	第 1 成分
elm2	第 2 成分

Description

実縦ベクトルを定義する。

Return

定義されたベクトル

See Also

MatDef, MatRowVec

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatColumnVec(3, 1.0, 2.0, 3.0);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

MatRowVec — 実横ベクトルの定義

Synopsis

```
Matrix  MatRowVec(size, elm1, elm2, ...)
int     size;
double  elm1, elm2, ...;
```

Arguments

size	ベクトルの大きさ
elm1	第 1 成分
elm2	第 2 成分

Description

実横ベクトルを定義する。

Return

定義されたベクトル

See Also

MatDef, MatColumnVec

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatRowVec(3, 1.0, 2.0, 3.0);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Chapter 3

高レベル行列演算

Name MatAdd — 行列の和

Synopsis

```
Matrix    MatAdd(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description mat1 と mat2 の和を計算する。

Return 和の結果の行列

See Also Mat_Add, C_Mat_Add

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatAdd(A, B);
   MatSetName(C, "C"); MatPrint(C);
}
```

Name

MatSub — 行列の差

Synopsis

```
Matrix    MatSub(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

mat1 と mat2 の差を計算する。(mat1 – mat2)

Return

差の結果の行列

See Also

Mat_Sub, C_Mat_Sub

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatSub(A, B);
   MatSetName(C, "C"); MatPrint(C);
}
```


Name

MatMul — 行列の積

Synopsis

```
Matrix    MatMul(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

mat1 と mat2 の積を計算する。(mat1 × mat2)

Return

積の結果の行列

See Also

Mat_Mul, C_Mat_Mul

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatMul(A, B);
   MatSetName(C, "C"); MatPrint(C);
}
```

Name

MatMulElem — 行列の成分毎の積

Synopsis

```
Matrix    MatMulElem(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

mat1 と mat2 の成分毎の積を計算する。

Return

積の結果の行列

See Also

MatMul, Mat_MulElem, C_Mat_MulElem

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatMulElem(A, B);
   MatSetName(C, "C"); MatPrint(C);
}
```


Name

MatInv — 逆行列

Synopsis

```
Matrix    MatInv(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の逆行列を計算する。

Return

逆行列の結果の行列

See Also

Mat_Inv, C_Mat_Inv

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatInv(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatInvElem — 成分が逆数の行列

Synopsis

```
Matrix  MatInvElem(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

成分が逆数の行列を求める。

Return

結果の行列

See Also

MatInv, Mat_InvElem, C_Mat_InvElem

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatInvElem(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```


Name

MatPseudoInv — 擬似逆行列

Synopsis

```
Matrix  MatPseudoMul(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の擬似逆行列を計算する。

Return

擬似逆行列の結果

See Also

Mat_PseudoInv, C_Mat_PseudoInv

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 1, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    B = MatPseudoInv(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```

Name

MatPow — 行列の累乗

Synopsis

```
Matrix   MatPow(mat, index)
Matrix   mat;
int      index;
```

Arguments

mat	任意のクラスの行列
index	指数

Description

mat の index 乗を計算する。 index が負の時 mat の逆行列の index 乗を計算する。

Return

mat の index 乗の結果

See Also

Mat_Pow, C_Mat_Pow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatPow(A, 3);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatScale — 行列の実数倍

Synopsis

```
Matrix   MatScale(mat, scale)
Matrix   mat;
double   scale;
```

Arguments

mat	任意のクラスの行列
scale	実数スケール

Description

mat の scale 倍を計算する。

Return

mat の scale 倍の結果

See Also

MatScaleC, Mat_Scale, C_Mat_Scale, C_Mat_ScaleC

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatScale(A, 2.0);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatScaleC — 行列の複素数倍

Synopsis

```
Matrix    MatScaleC(mat, scale)
Matrix    mat;
Complex   scale;
```

Arguments

mat	任意のクラスの行列
scale	複素数スケール

Description

mat の scale 倍を計算する。

Return

mat の scale 倍の結果

See Also

MatScale, Mat_Scale, C_Mat_Scale, C_Mat_ScaleC

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;
   Complex c;

   MatInit();
   c = CompDef(1.0, 2.0);
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatScaleC(A, c);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatScaleP — 行列の多項式倍

Synopsis

```
Matrix      MatScaleP(mat, scale)
Matrix      mat;
Polynomial  scale;
```

Arguments

mat	任意のクラスの行列
scale	多項式スケール

Description

mat の scale 倍を計算する。

Return

mat の scale 倍の結果

See Also

MatScale, MatScaleC

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;
   Polynomial p;

   MatInit();
   p = PolyConst(2.0);
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatScaleP(A, p);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatScaleR — 行列の有理関数倍

Synopsis

```
Matrix    MatScaleR(mat, scale)
Matrix    mat;
Rational  scale;
```

Arguments

mat	任意のクラスの行列
scale	有理関数スケール

Description

mat の scale 倍を計算する。

Return

mat の scale 倍の結果

See Also

MatScale, MatScaleC, MatScaleP

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;
    Polynomial r;

    MatInit();
    r = RatConst(2.0);
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatScaleR(A, r);
    MatSetName(B, "B");
    MatPrint(B);
}
```


Name

MatTrans — 転置行列

Synopsis

```
Matrix    MatTrans(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の転置行列を求める。

Return

mat の転置行列の結果

See Also

Mat_Trans, C_Mat_Trans

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatTrans(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```

Name

MatFlipLR — 左右反転行列

Synopsis

```
Matrix    MatFlipLR(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の左右反転行列を求める。

Return

mat の左右反転行列の結果

See Also

Mat_FlipLR, C_Mat_FlipLR, MatFlipUD

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatFlipLR(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```


Name

MatFlipUD — 上下反転行列

Synopsis

```
Matrix    MatFlipUD(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の上下反転行列を求める。

Return

mat の上下反転行列の結果

See Also

Mat_FlipUD, C_Mat_FlipUD, MatFlipLR

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatFlipUD(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```

Name

MatNegate — 行列の符合反転

Synopsis

```
Matrix  MatNegate(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の成分の符合を反転する。

Return

mat の成分の符合を反転した結果

See Also

Mat_Negate, C_Mat_Negate

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatNegate(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```


Name

MatExp — 指数関数行列

Synopsis

```
Matrix  MatExp(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

exp(mat) を計算する。

Return

exp(mat) の結果の行列

See Also

Mat_Exp(), C_Mat_Exp()

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatExp(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```

Name

MatConj — 複素共役行列

Synopsis

```
Matrix    MatConj(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の複素共役行列を求める。

Return

mat の複素共役行列の結果

See Also

C_Mat_Conj

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatConj(B);
   MatSetName(C, "C");
   MatPrint(C);
}
```


Name

MatConjTrans — 複素共役転置

Synopsis

```
Matrix    MatConjTrans(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の複素共役転置行列を求める。

Return

mat の複素共役転置行列の結果

See Also

Mat_Trans, C_Mat_ConjTrans

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatConjTrans(B);
   MatSetName(C, "C");
   MatPrint(C);
}
```

Name

MatVectorProduct — ベクトル積

Synopsis

```
Matrix    MatVectorProduct(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	3×1 の任意のクラスの行列
mat2	3×1 の任意のクラスの行列

Description

mat1 と mat2 のベクトル積を計算する。

Return

mat1 と mat2 のベクトル積

See Also

Mat_VectorProduct, C_Mat_VectorProduct

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 3, 1);
   B = MatDef("B", 3, 1);
   MatSetValue(1, 1, 1.0);
   MatSetValue(1, 2, 2.0);
   MatSetValue(1, 3, 3.0);
   MatCopy(B, A);
   C = MatVectorProduct(A, B);
   MatSetName(C, "C"); MatPrint(C);
}
```


Name

MatScalarProduct — 実ベクトルのスカラー積

Synopsis

```
double    MatScalarProduct(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	3×1 の実ベクトル
mat2	3×1 の実ベクトル

Description

mat1 と mat2 のスカラー積を計算する。

Return

mat1 と mat2 のスカラー積

See Also

C_MatScalarProduct

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;
   double dd;

   MatInit();
   A = MatDef("A", 3, 1);
   B = MatDef("B", 3, 1);
   MatSetValue(1, 1, 1.0);
   MatSetValue(1, 2, 2.0);
   MatSetValue(1, 3, 3.0);
   MatCopy(B, A);
   dd = MatScalarProduct(A, B);
   printf("dd = %lf‘n", dd);
}
```

Name

C_MatScalarProduct — 複素ベクトルのスカラー積

Synopsis

```
Complex   C_MatScalarProduct(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	3×1 の複素ベクトル
mat2	3×1 の複素ベクトル

Description

mat1 と mat2 のスカラー積を計算する。

Return

mat1 と mat2 のスカラー積

See Also

MatScalarProduct

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;
    Complex c;

    MatInit();
    A = MatDef("A", 3, 1);
    B = MatDef("B", 3, 1);
    MatSetValue(1, 1, 1.0);
    MatSetValue(1, 2, 2.0);
    MatSetValue(1, 3, 3.0);
    A = MatRealAndImag(A, B);
    MatCopy(B, A);
    c = C_MatScalarProduct(A, B);
    CompPrint(c);
}
```


Name

MatVecToDiag — ベクトルからの対角行列の生成

Synopsis

```
Matrix    MatVecToDiag(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの (列 / 行) ベクトル

Description

ベクトルの成分を対角成分にもつ対角行列を定義する。

Return

定義された行列

See Also

MatDiagDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 1);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 2, 1, 2.0);
    B = MatVecToDiag(A);
    MatPrint(B);
}
```

Name

MatDiagToVec — 対角行列からベクトルの生成

Synopsis

```
Matrix    MatDiagToVec(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの対角行列

Description

対角行列の対角成分をもつ列ベクトルを定義する。

Return

定義された行列

See Also

MatVecToDiag

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 1);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 2, 1, 2.0);
   B = MatVecToDiag(A);
   B = MatDiagToVec(B);
   MatPrint(B);
}
```


Name

MatEig — 一般行列の固有値 / 固有ベクトルの計算

Synopsis

```
void      MatEig(mat, val, vec)
Matrix    mat;
Matrix    *val;
Matrix    *vec;
```

Arguments

mat	(実 / 複素) 行列

Description

一般行列の固有値 / 固有ベクトルを計算する。

Return

なし

See Also

C_MatEig, Mat_Eig, MatHermitEig

Example

```
#include <matrix.h>

int main()
{
   Matrix A, val, vec;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   MatEig(A, &val, &vec);
   MatSetName(val, "eigval(A)"); MatPrint(val);
   MatSetName(vec, "eigvec(A)"); MatPrint(vec);
}
```

Name

MatHermitEig — エルミート行列の固有値 / 固有ベクトルの計算

Synopsis

```
void    MatHermitEig(mat, val, vec)
Matrix  mat;
Matrix  *val;
Matrix  *vec;
```

Arguments

mat	エルミート行列

Description

エルミート行列の固有値 / 固有ベクトルを計算する。

Return

なし

See Also

MatEig, C_Mat_HermitEig

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, val, vec;

   MatInit();
   A = MatDef("A", 2, 2); B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 2.0), 2, 2, 3.0);
   MatSetValue(B, 1, 2, -1.0);
   MatSetValue(B, 2, 1,  1.0);
   C = MatRealAndImag(A, B);
   MatHermitEig(C, &val, &vec);
   MatSetName(val, "eigval(C)"); MatPrint(val);
   MatSetName(vec, "eigvec(C)"); MatPrint(vec);
}
```


Name

MatEigVal — 一般行列の固有値

Synopsis

```
Matrix  MatEigVal(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の固有値を計算する。

Return

mat の固有値を列方向に並べた行列

See Also

MatEigVec, Mat_EigVal, C_Mat_EigVal

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatEigVal(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatEigVec — 一般行列の固有ベクトル

Synopsis

```
Matrix    MatEigVec(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の固有ベクトルを計算する。

Return

mat の固有ベクトルを列方向に並べた行列

See Also

MatEigVal, Mat_EigVec, C_Mat_EigVec

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatEigVec(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatEigValVec — 一般行列の固有値 / 固有ベクトル

Synopsis

```
Matrix    MatEigValVec(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の固有値 / 固有ベクトルを計算する。

Return

mat の固有値を列方向に並べた行列と
mat の固有ベクトルを列方向に並べた行列を
行方向に並べた行列 [val vec]

See Also

MatEig, MatEigVal, MatEigVec, Mat_EigValVec

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatEigValVec(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatSingVal — 特異値

Synopsis

```
Matrix    MatSingVal(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の特異値を計算する。

Return

mat の特異値を列方向に並べた行列

See Also

Mat_SingVal, C_Mat_SingVal

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatSingVal(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatSingVec — 特異値分解の左右の行列を対角に並べた行列

Synopsis

```
Matrix    MatSingVec(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の特異値分解の左右の行列を求める。

Return

mat の特異値分解の左右の行列を対角に並べた行列

See Also

MatSVD, MatSingVal

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatSingVec(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatSingValVec — 特異値分解の左右の行列と特異値を対角に並べた行列

Synopsis

```
Matrix    MatSingValVec(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の特異値分解の左右の行列を求める。

Return

mat の特異値分解の左右の行列と特異値を対角に並べた行列

See Also

MatSVD, MatSingVal, MatSingVec

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatSingValVec(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```


Name

MatSingLeftVec — 特異値分解の左の行列

Synopsis

```
Matrix    MatSingLeftVec(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の特異値分解の左の行列を求める。

Return

mat の特異値分解の左の行列

See Also

MatSVD, MatSingVec, MatSingRightVec

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatSingLeftVec(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```

Name

MatSingVec — 特異値分解の右の行列

Synopsis

```
Matrix    MatSingRightVec(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の特異値分解の右の行列を求める。

Return

mat の特異値分解の右の行列

See Also

MatSVD, MatSingVal, MatSingLeftVec

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatSingRightVec(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatSqrt — 平方根行列

Synopsis

```
Matrix    MatSqrt(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の平方根行列を計算する。

Return

mat の平方根行列

See Also

MatIsPositive,

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 0.0);
   MatSetValue(MatSetValue(A, 2, 1, 0.0), 2, 2, 4.0);
   B = MatSqrt(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatDeterm — 実行列の行列式

Synopsis

```
double    MatDeterm(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の行列式を計算する。

Return

mat の行列式

See Also

C_MatDeterm

Example

```
#include <matrix.h>

int main()
{
   Matrix A;
   double det;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   det = MatDeterm(A);
   printf("Determinat of A = %16.8E“n", det);
}
```


Name

MatFrobNorm — フロベニウスノルム

Synopsis

```
double    MatFrobNorm(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat のフロベニウスノルムを計算する。

Return

mat のフロベニウスノルム

See Also

Example

```
#include <matrix.h>

int main()
{
   Matrix A;
   double frob;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   frob = MatFrobNorm(A);
   printf("Frobenius norm of A = %16.8E“n", frob);
}
```

Name

MatInfNorm — 最大ノルム

Synopsis

```
double    MatInfNorm(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の最大ノルムを計算する。最大の絶対値を持つ成分の絶対値。

Return

mat の最大ノルム

See Also

Example

```
#include <matrix.h>

int main()
{
   Matrix A;
   double inf;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   inf = MatInfNorm(A);
   printf("Infinity norm of A = %16.8E“n", inf);
}
```


Name

MatTrace — 実行列のトレース

Synopsis

```
double    MatTrace(mat)
Matrix    mat;
```

Arguments

mat	実行列

Description

mat のトレースを計算する。

Return

mat のトレース

See Also

C_MatTrace

Example

```
#include <matrix.h>

int main()
{
   Matrix A;
   double trace;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   trace = MatTrace(A);
   printf("Trace of A = %16.8E“n", trace);
}
```

Name

MatMaxElem — 実行列の最大要素

Synopsis

```
double  MatMaxElem(mat)
Matrix  mat;
```

Arguments

mat	実行列

Description

行列の最大要素を取り出す。

Return

行列の最大要素

See Also

C_MatMaxElem

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    double max;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    max = MatMaxElem(A);
    printf("Maximum element of A = %16.8E“n", max);
}
```


Name

MatMinElem — 実行列の最小要素

Synopsis

```
double  MatMinElem(mat)
Matrix  mat;
```

Arguments

mat	実行列

Description

行列の最小要素を取り出す。

Return

行列の最小要素

See Also

C_MatMinElem

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    double min;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    min = MatMinElem(A);
    printf("Minimum element of A = %16.8E“n", min);
}
```

Name

MatSumElem — 実行列の要素の合計

Synopsis

```
double  MatSumElem(mat)
Matrix  mat;
```

Arguments

mat	実行列

Description

行列の成分の合計を計算する。

Return

行列の成分の合計

See Also

C_MatSumElem

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    double sum;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    sum = MatSumElem(A);
    printf("Sum of Afls elements = %16.8E“n", sum);
}
```


Name

MatMeanElem — 実行列の要素の平均値

Synopsis

```
double  MatMeanElem(mat)
Matrix  mat;
```

Arguments

mat	実行列

Description

行列の成分の平均値を計算する。

Return

行列の成分の平均値

See Also

C_MatMeanElem

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    double mean;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    mean = MatMeanElem(A);
    printf("Mean of Afls elements = %16.8E“n", mean);
}
```

Name

MatMaxSingVal — 最大特異値

Synopsis

```
double  MatMaxSingVal(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の最大特異値を求める。

Return

mat の最大特異値

See Also

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    double max;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    max = MatMaxSingVal(A);
    printf("Maxsingular value of A = %16.8E“n",max);
}
```


Name

MatMinSingVal — 最小特異値

Synopsis

```
double  MatMinSingVal(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の最小特異値を求める。

Return

mat の最小特異値

See Also

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    double min;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    min = MatMinSingVal(A);
    printf("Minsingular value of A = %16.8E“n",min);
}
```

Name

MatGetValue MatValue — 実行列の要素

Synopsis

```
double   MatGetMat(mat, column, row)
double   MatValue(mat, column, row)
Matrix   mat;
int      column, row;
```

Arguments

mat	実行列
column	column 行
row	row 列

Description

mat の column 行 row 列の要素を取り出す。

Return

mat の column 行 row 列の要素

See Also

MatSetValue, C_MatGetValue, C_MatValue

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    double elm;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    elm = MatGetValue(A, 1, 1);
    printf("A(1,1) = %16.8E“n", elm);
}
```


Name

MatSetValue — 実行列の成分の代入

Synopsis

```
Matrix   MatSetValue(mat, column, row, data)
Matrix   mat;
int      column, row;
double   data;
```

Arguments

mat	成分の代入を行なう行列
column	第 column 行
row	第 row 列
data	設定するデータ

Description

行列の成分（実数）を代入する。

Return

成分を代入した結果の行列

See Also

MatGetValue, MatValue

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatPrint(A);
}
```

Name

MatGetPtr MatPtr — 実行列の成分へ

Synopsis

```
double    MatGetPtr(mat, column, row)
double    MatPtr(mat, column, row)
Matrix    mat;
int       column, row;
```

Arguments

mat	実行列
column	column 行
row	row 列

Description

行列の要素を返す。

Return

行列の要素

See Also

C_MatGetPtr

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   printf("A(1,1) = %16.8E“n", *MatGetPtr(A));
}
```


Name

MatRank — 行列のランク

Synopsis

```
int       MatRank(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

行列のランクを計算する。

Return

行列のランク

See Also

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   printf("Rank of A is %d“n", MatRank(A));
}
```

Name

MatKernel — 一般行列の零空間 (カーネル)

Synopsis

```
Matrix    MatKernel(mat)
Matrix    mat;
```

Arguments

mat	(実 / 複素) 行列

Description

mat の零空間 (カーネル) を求める。

Return

mat の零空間 (カーネル)

See Also

MatSVD, MatImage

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatKernel(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Chapter 4

行列の合成・分解・変換

Name MatCatColumn — 行列の列接続

Synopsis

```
Matrix    MatCatColumn(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description 列方向に行列を接続する。
[mat1 mat2]′

Return 列方向に接続した行列

See Also MatCatRow, Mat_CatColumn

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatCatColumn(A, B);
   MatSetName(C, "C"); MatPrint(C);
}
```

Name

MatCatColumns — 複数の行列の列接続

Synopsis

```
Matrix    MatCatColumns(num, mat1, mat2, ...)
int       num;
Matrix    mat1, mat2;
```

Arguments

num	行列の数
mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

列方向に複数の行列を接続する。
[mat1 mat2]′

Return

列方向に接続した行列

See Also

MatCatRows, MatCatColumn,

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   B = MatDef("B", 1, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   C = MatCatColumns(4, A, A, B, B);
   MatSetName(C, "C"); MatPrint(C);
}
```


Name

MatCatRow — 行列の行接続

Synopsis

```
Matrix    MatCatRow(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

行方向に行列を接続する。
[mat1 mat2]

Return

行方向に接続した行列

See Also

MatCatColumn, Mat_CatRow, C_Mat_CatRow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatCatRow(A, B);
   MatSetName(C, "C"); MatPrint(C);
}
```

Name

MatCatRows — 複数の行列の行接続

Synopsis

```
Matrix    MatCatRows(number, mat1, mat2, ...)
int       number;
Matrix    mat1, mat2;
```

Arguments

number	行列の数
mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

行方向に複数の行列を接続する。
[mat1 mat2]

Return

行方向に接続した行列

See Also

MatCatColumns, MatCatRow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   B = MatDef("B", 1, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   C = MatCatRows(4, A, A, B, B);
   MatSetName(C, "C"); MatPrint(C);
}
```


Name

MatCat4 — ４つの行列の接続

Synopsis

```
Matrix    MatCat4(mat1, mat2, mat3, mat4)
Matrix    mat1, mat2, mat3, mat4;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列
mat3	任意のクラスの行列
mat4	任意のクラスの行列

Description

４つの行列を接続する。
[[mat11 mat12] [mat21 mat22]]

Return

４つの行列を接続した行列

See Also

Mat_Cat4, C_Mat_Cat4

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D, E;

   MatInit();
   A = MatDef("A", 2, 2); B = MatDef("B", 2, 2);
   C = MatDef("C", 2, 2); D = MatDef("D", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   MatCopy(B, A); MatCopy(C, A); MatCopy(D, A);
   E = MatCat4(A, B, C, D);
   MatSetName(E, "E"); MatPrint(E);
}
```

Name

MatCut — ブロック行列の取り出し

Synopsis

```
Matrix    MatCut(mat, sc, sr, ec, er)
Matrix    mat;
int       sc, sr, ec, er;
```

Arguments

mat	任意のクラスの行列
sc	左上の行番号
sr	左上の列番号
ec	右下の行番号
er	右下の列番号

Description

(sc, sr) を左上、 (ec, sr) を右下とするブロック行列を取り出す。

Return

取り出した行列

See Also

Mat_Cut, C_Mat_Cut, MatPut

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(4, 1.0, 2.0, 3.0, 4.0);
   B = MatCut(A, 2, 2, 3, 3);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatColumnVector — 列ベクトルの取り出し

Synopsis

```
Matrix    MatColumnVector(mat, number)
Matrix    mat;
int       number;
```

Arguments

mat	任意のクラスの行列
number	列の番号

Description

number 番目の列ベクトルを取り出す。

Return

取り出した列ベクトル

See Also

MatColumnVec, MatCut, MatRowVector

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(4, 1.0, 2.0, 3.0, 4.0);
   B = MatColumnVector(A, 2);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatRowVector — 行ベクトルの取り出し

Synopsis

```
Matrix    MatRowVector(mat, number)
Matrix    mat;
int       number;
```

Arguments

mat	任意のクラスの行列
number	行の番号

Description

number 番目の行ベクトルを取り出す。

Return

取り出した行ベクトル

See Also

MatRowVec, MatCut, MatColumnVector

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(4, 1.0, 2.0, 3.0, 4.0);
   B = MatRowVector(A, 2);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatPut — ブロック行列の代入

Synopsis

```
Matrix    MatPut(mat1, sc, sr, mat2)
Matrix    mat1, mat2;
int       sc, sr;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列
sc	左上の行番号
sr	左上の列番号

Description

(sc, sr) が左上になるようにブロック行列を代入する。

Return

(sc, sr) が左上になるようにブロック行列を代入した行列

See Also

Mat_Put, C_Mat_Put

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 4, 4);
   B = MatDiagDef(2, 1.0, 2.0);
   MatPut(A, 2, 2, B);
   MatPrint(A);
}
```

Name

MatRealPart — 行列の実部行列

Synopsis

```
Matrix    MatRealPart(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の実部行列を取り出す。

Return

mat の実部行列の結果

See Also

MatImagPart, C_MatRealPart, C_MatImagPart

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatRealPart(B);
   MatSetName(C, "C"); MatPrint(C);
}
```


Name

MatImagPart — 複素行列の虚部行列

Synopsis

```
Matrix    MatImagPart(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

mat の虚部行列を取り出す。

Return

mat の虚部行列の結果

See Also

MatRealPart, Mat_RealPart, Mat_ImagPart

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatImagPart(B);
   MatSetName(C, "C");
   MatPrint(C);
}
```

Name

MatRealAndImag — 実部と虚部による複素行列の合成

Synopsis

```
Matrix  MatRealAndImag(real, imag)
Matrix  real, imag;
```

Arguments

real	実行列
imag	実行列

Description

real を実部、 imag を虚部とする複素行列を作る。

Return

合成した複素行列

See Also

Mat_RealAndImag

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDiagDef(2, 1.0, 2.0);
   B = MatDiagDef(2, 3.0, 4.0);
   C = MatRealAndImag(A, B);
   MatSetName(C, "C");
   MatPrint(C);
}
```


Name

MatRealToComp — 実行列から複素行列への変換

Synopsis

```
Matrix  MatRealToComp(mat)
Matrix  mat;
```

Arguments

mat	実行列

Description

mat を複素行列に変換する。

Return

変換結果の複素行列

See Also

Mat_RealToComp

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(2, 1.0, 2.0);
   B = MatRealToComp(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatRealToPoly — 実行列から多項式行列への変換

Synopsis

```
Matrix  MatRealToPoly(mat)
Matrix  mat;
```

Arguments

mat	実行列

Description

mat を多項式行列に変換する。

Return

変換結果の多項式行列

See Also

MatRealToComp, MatRealToRat

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDiagDef(2, 1.0, 2.0);
    B = MatRealToPoly(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```


Name

MatRealToRat — 実行列から有理関数行列への変換

Synopsis

```
Matrix  MatRealToRat(mat)
Matrix  mat;
```

Arguments

mat	実行列

Description

mat を有理関数行列に変換する。

Return

変換結果の有理関数行列

See Also

MatRealToComp, MatRealToPoly

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDiagDef(2, 1.0, 2.0);
    B = MatRealToRat(A);
    MatSetName(B, "B");
    MatPrint(B);
}
```

Name

MatCompToPoly — 複素行列から多項式行列への変換

Synopsis

```
Matrix    MatCompToPoly(mat)
Matrix    mat;
```

Arguments

mat	複素行列

Description

mat を多項式行列に変換する。

Return

変換結果の多項式行列

See Also

MatRealToComp, MatRealToPoly

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(2, 1.0, 2.0);
   A = MatRealTopComp(A);
   B = MatCompToPoly(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatCompToRat — 複素行列から有理関数行列への変換

Synopsis

```
Matrix    MatCompToRat(mat)
Matrix    mat;
```

Arguments

mat	複素行列

Description

mat を有理関数行列に変換する。

Return

変換結果の有理関数行列

See Also

MatRealToRat, MatCompToPoly

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(2, 1.0, 2.0);
   A = MatRealTopComp(A);
   B = MatCompToRat(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatPolyToRat — 多項式行列から有理関数行列への変換

Synopsis

```
Matrix    MatPolyToRat(mat)
Matrix    mat;
```

Arguments

mat	多項式行列

Description

mat を有理関数行列に変換する。

Return

変換結果の有理関数行列

See Also

MatRealToRat, MatCompToPoly

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(2, 1.0, 2.0);
   A = MatRealTopPoly(A);
   B = MatPolyToRat(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatBlockDiag — ブロック対角行列の定義

Synopsis

```
Matrix    MatBlockDiag(size, mat1, mat2, ...)
int       size;
Matrix    mat1, mat2, ...;
```

Arguments

size	対角ブロックの数
mat1	第（1、1）成分
mat2	第（2、2）成分

Description

ブロック対角行列を定義する。

Return

定義された行列

See Also

MatDiagDef,

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(3, 1.0, 2.0, 3.0);
   B = MatBlockDiag(3, A, A, A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

MatCompanion — 行同伴形行列の定義

Synopsis

```
Matrix    MatCompanion(mat)
Matrix    mat
```

Arguments

mat	任意のクラスの行ベクトル

Description

mat 行列が最下行の行同伴形行列を定義する。

Return

定義された行列

See Also

MatCompanion2

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatRowVec(3, 1.0, 2.0, 3.0);
   B = MatCompanion(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

MatCompanion2 — 列同伴形行列の定義

Synopsis

```
Matrix    MatCompanion2(mat)
Matrix    mat
```

Arguments

mat	任意のクラスの行ベクトル

Description

mat の共役複素転置行列が最右行の列同伴形行列を定義する。

Return

定義された行列

See Also

MatCompanion

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatColumnVec(3, 1.0, 2.0, 3.0);
   B = MatCompanion2(A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Chapter 5

行列の特性

Name MatGetClass MatCalss — 行列のクラス

Synopsis

```
int     MatGetClass(mat)
int     MatClass(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description クラス (実 / 複素 / 多項式 / 有理関数) を返す。

Return 行列のクラス (実 / 複素 / 多項式 / 有理関数)

See Also MatSetClass

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MAT_REAL = MatGetClass(A))
        printf("A is a real matrix.“n");
    else
        printf("A is a complex matrix.“n");
}
```

Name

MatSetClass — 行列のクラスの設定

Synopsis

```
Matrix    MatSetClass(mat, class)
Matrix    mat;
int       class;
```

Arguments

mat	クラスの設定を行なう行列
class	クラス (実 / 複素 / 多項式 / 有理関数)

Description

クラス (実 / 複素 / 多項式 / 有理関数) を設定する。
MAT_REAL, MAT_COMPLEX,
MAT_POLY, MAT_RAT

Return

クラスを設定した行列

See Also

MatGetClass, MatClass

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatSetClass(A, MAT_REAL);
    MatPrint(A);
}
```


Name

MatGetType MatType — 行列のタイプ

Synopsis

```
int       MatGetType(mat)
int       MatType(mat)
Matrix    mat;
```

Arguments

mat	任意のクラスの行列

Description

行列のタイプ (登録行列 / テンポラリ行列) を返す。

Return

行列のタイプ (登録行列 / テンポラリ行列)

See Also

MatSetType

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MAT_VAR = MatGetType(A))
        printf("A is registered matrix.“n");
    else
        printf("A is not a registered matrix.“n");
}
```

Name

MatSetType — 行列のタイプの設定

Synopsis

```
Matrix  MatSetType(mat, type)
Matrix  mat;
int     type;
```

Arguments

mat	タイプの設定を行なう行列
type	行列のタイプ

Description

タイプ (登録行列 / テンポラリ行列) を設定する。
MAT_VAR, MAT_TMP

Return

タイプを設定した行列

See Also

MatGetType, MatType

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   MatSetTYPE(A, MAT_VAR);
   MatPrint(A);
}
```


Name

MatGetName MatName — 行列の名前

Synopsis

```
char    * MatGetName(mat)
char    * MatName(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

行列の名前を返す。

Return

行列の名前を返す。

See Also

MatSetName, MatSearch

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   printf("Name is %s“n", MatGetName(A));
}
```

Name

MatSetName — 行列の名前の設定

Synopsis

```
Matrix  MatSetName(mat, name)
Matrix  mat;
char    * name;
```

Arguments

mat	名前の設定を行なう行列
name	行列の名前

Description

行列の名前を設定する。

Return

名前を設定した行列

See Also

MatGetName, MatName

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatNonNameDef(2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

MatRows Rows — 行列の行の数

Synopsis

```
int     MatRows(mat)
int     Rows(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

行列の行の数を返す。

Return

行列の行の数

See Also

MatCols, Cols

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    printf("Row size of A is %d“n", Rows(A));
}
```

Name

MatCols Cols — 行列の列の数

Synopsis

```
int     MatCols(mat)
int     Cols(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

行列の列の数を返す。

Return

行列の列の数

See Also

MatRows, Rows

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    printf("Column size of A is %d“n", Cols(A));
}
```


Name

MatIsReal — 実行列の有無

Synopsis

```
int     MatIsReal(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

実行列の有無を調べる。

Return

実行列なら真（１）

See Also

MatIsComplex

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsReal(A))
        printf("A is a real matrix“n");
    else
        printf("A is not a real matrix“n");
}
```

Name

MatIsComplex — 複素行列の有無

Synopsis

```
int     MatIsComplex(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

複素行列の有無を調べる。

Return

複素行列なら真 (1)

See Also

MatIsReal

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsComplex(A))
        printf("A is a complex matrix“n");
    else
        printf("A is not a complex matrix“n");
}
```


Name

MatIsVar — 登録行列の有無

Synopsis

```
int     MatIsVar(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

登録行列の有無を調べる。

Return

登録行列なら真 (1)

See Also

MatIsTmp

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsVar(A))
        printf("A is a registered matrix“n");
    else
        printf("A is not a registered matrix“n");
}
```

Name

MatIsTmp — テンポラリ行列の有無

Synopsis

```
int     MatIsTmp(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

テンポラリ行列の有無を調べる。

Return

テンポラリ行列なら真（１）

See Also

MatIsVar

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsTmp(A))
        printf("A is a temporary matrix“n");
    else
        printf("A is not a temporay matrix“n");
}
```


Name

MatIsFullRank — フルランクの有無

Synopsis

```
int     MatIsFullRank(mat)
Matrix  mat;
```

Arguments

mat	（実 / 複素）行列

Description

フルランクの有無をを調べる。

Return

フルランクなら真（１）

See Also

MatRank

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsFullRank(A))
        printf("A is a Matrix of full rank.“n");
    else
        printf("A is not a Matrix of full rank.“n");
}
```

Name

MatIsSingular — 非正則の有無

Synopsis

```
int     MatIsSingular(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

非正則の有無をを調べる。

Return

非正則なら真 (1)

See Also

MatIsNonSingular

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsSingular(A))
        printf("A is singular.“n");
    else
        printf("A is nonsingular.“n");
}
```


Name

MatIsNonSingular — 正則の有無

Synopsis

```
int     MatIsNonSingular(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

正則の有無をを調べる。

Return

正則なら真 (1)

See Also

MatIsSingular

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsNonSingular(A))
        printf("A is nonsingular.“n");
    else
        printf("A is singular.“n");
}
```

Name

MatIsSquare — 正方の有無

Synopsis

```
int     MatIsSquare(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

正方の有無をを調べる。

Return

正方なら真（１）

See Also

MatIsNonSquare

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsSquare(A))
        printf("A is a square matrix.¥n");
    else
        printf("A is not a square matrix.¥n");
}
```


Name

MatIsNonSquare — 非正方の有無

Synopsis

```
int     MatIsNonSquare(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

非正方の有無をを調べる。

Return

非正方なら真（１）

See Also

MatIsSquare

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsNonSquare(A))
        printf("A is not a square matrix.¥n");
    else
        printf("A is a square matrix.¥n");
}
```

Name

MatIsPositive — 正定の有無

Synopsis

```
int     MatIsPositive(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

正定の有無をを調べる。

Return

正定なら真 (1)

See Also

MatIsPositiveSemi, MatIsNegative

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsPositive(A))
        printf("A is positive definite.“n");
}
```


Name

MatIsPositiveSemi — 準正定の有無

Synopsis

```
int     MatIsPositiveSemi(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

準正定の有無をを調べる。

Return

準正定なら真 (1)

See Also

MatIsPositive, MatIsNegative

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsPositiveSemi(A))
        printf("A is positive semi definite.“n");
}
```

Name

MatIsNonPositivei — 非正定の有無

Synopsis

```
int     MatIsNonPositive(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

非正定の有無をを調べる。

Return

非正定なら真 (1)

See Also

MatIsPositive, MatIsNegative

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsNonPositivei(A))
        printf("A is nonpositive definite.“n");
}
```


Name

MatIsNegative — 負定の有無

Synopsis

```
int     MatIsNegative(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

負定の有無をを調べる。

Return

負定なら真 (1)

See Also

MatIsPositive, MatIsNegativeSemi

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsNegative(A))
        printf("A is negative definite.“n");
}
```

Name

MatIsNegativeSemi — 準負定の有無

Synopsis

```
int     MatIsNegativeSemi(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

準負定の有無をを調べる。

Return

準負定なら真 (1)

See Also

MatIsPositive, MatIsNegative

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsNegativeSemi(A))
        printf("A is negative semi definite.“n");
}
```


Name

MatIsNonNegative — 準負定の有無

Synopsis

```
int     MatIsNonNegative(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

非負定の有無をを調べる。

Return

非負定なら真 (1)

See Also

MatIsPositive, MatIsNegative

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsNonNegative(A))
        printf("A is nonnegative definite.“n");
}
```

Name

MatIsZero — 零行列の有無

Synopsis

```
int     MatIsZero(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

零行列の有無をを調べる。

Return

零行列なら真 (1)

See Also

MatIsNonZero()

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsZero(A))
        printf("A is a zero matrix.“n");
}
```


Name

MatIsNonZero — 非零行列の有無

Synopsis

```
int     MatIsNonZero(mat)
Matrix  mat;
```

Arguments

mat	(実 / 複素) 行列

Description

非零行列の有無をを調べる。

Return

非零行列なら真 (1)

See Also

MatIsZero()

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsNonZero(A))
        printf("A is a zero matrix.“n");
}
```

Name

MatIsEqual — 等号の有無

Synopsis

```
int     MatIsEqual(mat1, mat2)
Matrix  mat1,mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

等号の有無をを調べる。2 つの行列のクラスは一致していなければならない。

Return

等号なら真 (1)

See Also

MatIsNonEqual()

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2), B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatCopy(B, A);
    if (MatIsEqual(A, B))
        printf("A is equal to B.“n");
}
```


Name

MatIsNonEqual — 非等号の有無

Synopsis

```
int     MatIsNonEqual(mat1, mat2)
Matrix  mat1,mat2;
```

Arguments

mat1	任意のクラスの行列
mat2	任意のクラスの行列

Description

非等号の有無をを調べる。2 つの行列のクラスは一致していなければならない。

Return

非等号なら真 (1)

See Also

MatIsEqual()

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2), B = MatDef("B", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    MatCopy(B, A);
    if (MatIsNonEqual(A, B))
        printf("A is not equal to B.“n");
}
```

Name

MatIsColumnVec — 列ベクトルの有無

Synopsis

```
int     MatIsColumnVec(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

列ベクトルの有無をを調べる。

Return

列ベクトルなら真（１）

See Also

MatIsRowVec()

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsColumnVec(A))
        printf("A is a column vector.“n");
}
```


Name

MatIsRowVec — 行ベクトルの有無

Synopsis

```
int     MatIsRowVec(mat)
Matrix  mat;
```

Arguments

mat	任意のクラスの行列

Description

行ベクトルの有無をを調べる。

Return

行ベクトルなら真（１）

See Also

MatIsColumnVec()

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    if (MatIsRowVec(A))
        printf("A is a row vector.“n");
}
```

Chapter 6

低レベル実行列演算

Name

Mat_Add — 実行列の和

Synopsis

```
Matrix    Mat_Add(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

ans	実行列
mat1	実行列
mat2	実行列

Description

mat1 と mat2 の和を計算して，ans に代入する。

Return

ans 行列

See Also

MatAdd, C_Mat_Add

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatDef("C", 2, 2);
   Mat_Add(C, A, B);
   MatPrint(C);
}
```

Name

Mat_Sub — 実行列の差

Synopsis

```
Matrix    Mat_Sub(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

asns	実行列
mat1	実行列
mat2	実行列

Description

mat1 と mat2 の差を計算して， ans に代入する。

Return

ans 行列

See Also

MatSub, C_Mat_Sub

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatDef("C", 2, 2);
   Mat_Sub(C, A, B);
   MatPrint(C);
}
```


Name

Mat_Mul — 実行列の積

Synopsis

```
Matrix    Mat_Mul(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

ans	実行列
mat1	実行列
mat2	実行列

Description

mat1 と mat2 の積を計算して， ans に代入する。

Return

ans 行列

See Also

MatMul, C_Mat_Mul

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatDef("C", 2, 2);
   Mat_Mul(C, A, B);
   MatPrint(C);
}
```

Name

Mat_MulElem — 実行列の成分毎の積

Synopsis

```
Matrix    Mat_MulElem(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

ans	実行列
mat1	実行列
mat2	実行列

Description

mat1 と mat2 の成分毎の積を計算して， ans に代入する。

Return

ans 行列

See Also

MatMulElem, C_Mat_MulElem

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatDef("C", 2, 2);
   Mat_MulElem(C, A, B);
   MatPrint(C);
}
```


Name

Mat_Inv — 実逆行列

Synopsis

```
Matrix    Mat_Inv(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列 (正方)

Description

mat の逆行列を計算して， ans に代入する。

Return

ans 行列

See Also

MatInv, C_Mat_Inv

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   Mat_Inv(B, A);
   MatPrint(B);
}
```

Name

Mat_InvElem — 逆数を成分とする行列

Synopsis

```
Matrix    Mat_InvElem(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列

Description

成分が逆数である行列をを計算して， ans に代入する。

Return

ans 行列

See Also

MatInvElem, C_Mat_InvElem

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   Mat_InvElem(B, A);
   MatPrint(B);
}
```


Name

Mat_PseudoInv — 実擬似逆行列

Synopsis

```
Matrix    Mat_PseudoMul(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列

Description

mat の擬似逆行列を計算して， ans に代入する。

Return

ans 行列

See Also

MatPseudoInv, C_Mat_PseudoInv

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   B = MatDef("B", 2, 1);
   Mat_PseudoInv(B, A);
   MatPrint(B);
}
```

Name

Mat_Pow — 実行列の累乗

Synopsis

```
Matrix    Mat_Pow(ans, mat, index)
Matrix    ans, mat;
int       index;
```

Arguments

ans	実行列
mat	実行列
index	指数

Description

mat の index 乗を計算して， ans に代入する。

Return

ans 行列

See Also

MatPow, C_Mat_Pow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   Mat_Pow(B, A, 3);
   MatPrint(B);
}
```


Name

Mat_Scale — 実行列の実数倍

Synopsis

```
Matrix    Mat_Scale(ans, mat, scale)
Matrix    ans, mat;
double    scale;
```

Arguments

ans	実行列
mat	実行列
scale	実数スケール

Description

mat の scale 倍を計算して， ans に代入する。

Return

ans 行列

See Also

MatScale, C_Mat_Scale, C_Mat_ScaleC

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   Mat_Scale(B, A, 2.0);
   MatPrint(B);
}
```

Name

Mat_Trans — 実転置行列

Synopsis

```
Matrix    Mat_Trans(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列

Description

mat の転置行列を求める。

Return

ans 行列

See Also

MatTrans, C_Mat_Trans

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 1, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    B = MatDef("B", 2, 1);
    Mat_Trans(B, A);
    MatPrint(B);
}
```


Name

Mat_FlipLR — 実左右反転行列

Synopsis

```
Matrix    Mat_FlipLR(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列

Description

mat の実左右反転行列を求める。

Return

ans 行列

See Also

MatFlipLR, C_Mat_FlipLR

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatDef("B", 2, 2);
    Mat_FlipLR(B, A);
    MatPrint(B);
}
```

Name

Mat_FlipUD — 実上下反転行列

Synopsis

```
Matrix    Mat_FlipUD(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列

Description

mat の実上下反転行列を求める。

Return

ans 行列

See Also

MatFlipUD, C_Mat_FlipUD

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   Mat_FlipUD(B, A);
   MatPrint(B);
}
```


Name

Mat_Negate — 実行列の符合反転

Synopsis

```
Matrix    Mat_Negate(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列

Description

mat の成分の符合を反転して， ans に代入する。

Return

ans 行列

See Also

MatNegate, C_Mat_Negate

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   Mat_Negate(B, A);
   MatPrint(B);
}
```

Name

Mat_CatColumn — 実行列の列接続

Synopsis

```
Matrix    Mat_CatColumn(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

ans	実行列
mat1	実行列
mat2	実行列

Description

列方向に行列を接続して，ans に代入する。
$[\ \mathrm{mat1}\ \mathrm{mat2}\ ]'$

Return

ans 行列

See Also

MatCatColumn, C_Mat_CatColumn

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatDef("C", 4, 2);
   Mat_CatColumn(C, A, B);
   MatPrint(C);
}
```


Name

Mat_CatRow — 実行列の行接続

Synopsis

```
Matrix    Mat_CatRow(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

ans	実行列
mat1	実行列
mat2	実行列

Description

行方向に行列を接続して，ans に代入する。
$[\ \mathrm{mat1}\ \mathrm{mat2}\ ]$

Return

ans 行列

See Also

MatCatRow, C_Mat_CatRow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatDef("C", 2, 4);
   Mat_CatRow(C, A, B);
   MatPrint(C);
}
```

Name

Mat_Cat4 — ４つの実行列の接続

Synopsis

```
Matrix  Mat_Cat4(ans, mat1, mat2, mat3, mat4)
Matrix  ans, mat1, mat2, mat3, mat4;
```

Arguments

ans	実行列
mat1, mat2	実行列
mat3, mat4	実行列

Description

４つの行列を接続して，ans に代入する。
[[mat11 mat12] [mat21 mat22]]

Return

ans 行列

See Also

MatCat4, C_Mat_Cat4

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D, E;

   MatInit();
   A = MatDef("A", 1, 2); B = MatDef("B", 1, 2);
   C = MatDef("C", 1, 2); D = MatDef("D", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatCopy(B, A); MatCopy(C, A); MatCopy(D, A);
   E = MatDef("E", 4, 4);
   Mat_Cat4(E, A, B, C, D); MatPrint(E);
}
```


Name

Mat_Cut — 実ブロック行列の取り出し

Synopsis

```
Matrix  Mat_Cut(ans, mat, sc, sr, ec, er)
Matrix  ans, mat;
int     sc, sr, ec, er;
```

Arguments

ans	実行列
mat	実行列
sc	左上の行番号
sr	左上の列番号
ec	右下の行番号
er	右下の列番号

Description

(sc, sr) を左上，(ec, sr) を右下とするブロック行列を取り出して，ans に代入する。

Return

ans 行列

See Also

MatCut, C_Mat_Cut

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

    MatInit(); A = MatDiagDef(4, 1.0, 2.0, 3.0, 4.0);
B = MatDef("B", 2, 2); Mat_Cut(B, A, 2, 2, 3, 3);
MatPrint(B); }
```

Name

Mat_Put — 実ブロック行列の代入

Synopsis

```
Matrix    Mat_Put(mat1, sc, sr, mat2)
Matrix    mat1, mat2;
int       sc, sr;
```

Arguments

mat1	実行列
mat2	実行列
sc	左上の行番号
sr	左上の列番号

Description

(sc, sr) が左上になるようにブロック行列を代入する。

Return

mat1 行列

See Also

MatPut, C_Mat_Put

Example

```
#include <matrix.h>

int main() { Matrix A, B;

  MatInit(); A = MatDef("A", 4, 4); B = MatDiagDef(2, 1.0, 2.0); Mat_Put(A, 2, 2, B); MatPrint(A);
}
```


Name

Mat_VectorProduct — 実ベクトル積

Synopsis

```
Matrix    Mat_VectorProduct(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

ans	3×1 の実行列
mat1	3×1 の実行列
mat2	3×1 の実行列

Description

mat1 と mat2 のベクトル積を計算して， ans に代入する。

Return

ans 行列

See Also

MatVectorProduct

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 3, 1);
   B = MatDef("B", 3, 1);
   C = MatDef("C", 3, 1);
   MatSetValue(1, 1, 1.0);
   MatSetValue(1, 2, 2.0);
   MatSetValue(1, 3, 3.0);
   MatCopy(B, A);
   Mat_VectorProduct(C, A, B);
   MatPrint(C);
}
```

Name

Mat_Eig — 一般実行列の固有値 / 固有ベクトルの計算

Synopsis

```
void    Mat_Eig(mat, val, vec)
Matrix  mat;
Matrix  val, vec;
```

Arguments

mat	実行列
val	複素行列
vec	複素行列

Description

一般行列の固有値 / 固有ベクトルを計算して，
val と vec に代入する。

Return

なし

See Also

MatEig, MatHermitEig

Example

```
#include <matrix.h>

int main()
{
   Matrix A, val, vec;

   MatInit();
   A = MatDef("A", 2, 2);
   val = MatEigValDef(A);
   vec = MatEigVecDef(A);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   Mat_Eig(A, val, vec);
   MatSetName(val, "eigval(A)"); MatPrint(val);
   MatSetName(vec, "eigvec(A)"); MatPrint(vec);
}
```


Name

Mat_EigVal — 一般実行列の固有値

Synopsis

```
Matrix  Mat_EigVal(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	実行列

Description

mat の固有値を計算して， ans に代入する。

Return

ans 行列

See Also

MatEigVal, C_Mat_EigVal

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   B = MatEigValDef(A);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   Mat_EigVal(B, A);
   MatSetName(B, "B"); MatPrint(B);
}
```

Name

Mat_EigVec — 一般実行列の固有ベクトル

Synopsis

```
Matrix    Mat_EigVec(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	複素行列
mat	実行列

Description

mat の固有ベクトルを計算して， ans に代入する。

Return

ans 行列

See Also

MatEigVec, C_Mat_EigVec

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   B = MatEigVecDef(A);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   Mat_EigVec(B, A);
   MatSetName(B, "B");
   MatPrint(B);
}
```


Name

Mat_EigValVec — 一般実行列の固有値 / 固有ベクトル

Synopsis

```
Matrix    MatEigValVec(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	複素行列
mat	実行列

Description

mat の固有値を列方向に並べた行列と
mat の固有ベクトルを列方向に並べた行列を
行方向に並べて（[val vec]）ans に代入する。

Return

ans 行列

See Also

MatEigValVec, C_Mat_EigValVec

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, val, vec;

   MatInit();
   A = MatDef("A", 2, 2);
   val = MatEigValDef(A); vec = MatEigVecDef(A);
   B = MatCatRow(val, vec);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   Mat_EigValVec(B, A);
   MatSetName(B, "B");
   MatPrint(B);
}
```

Name

Mat_SingVal — 実行列の特異値

Synopsis

```
Matrix    Mat_SingVal(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	実行列

Description

mat の特異値を計算して， ans に代入する。

Return

ans 行列

See Also

MatSingVal, C_MatSingVal

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   B = MatDef("B", 2, 1);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   Mat_SingVal(B, A);
   MatPrint(B);
}
```

Chapter 7

複素行列ユーティリティー

Name C_MatInput — キーボードからの複素行列の入力

Synopsis

```
Matrix  C_MatInput(name)
char    * name;
```

Arguments

name	入力する複素行列の名前

Description キーボードから複素行列を入力し、名前を name とする。

Return 入力された複素行列

See Also MatEdit, MatInput

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   C_MatInput("A");
   MatPrint(A);
}
```

Name

C_MatCopy — 複素行列の成分のコピー

Synopsis

```
Matrix    C_MatCopy(to, from)
Matrix    to, from;
```

Arguments

to	代入先の複素行列
from	被代入複素行列

Description

from 複素行列の成分を to 複素行列にコピーする。

Return

コピーされた結果の複素行列

See Also

MatAssign, MatCopy

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_MatCopy(C, A);
   MatPrint(C);
}
```


Name

C_MatRead — キーボードからの複素行列の入力

Synopsis

```
Matrix    C_MatInput(mat)
char      * mat;
```

Arguments

mat	複素行列

Description

キーボードから複素行列を入力し、 mat に代入する。

Return

mat 行列

See Also

MatEdit, MatInput

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = C_MatDef("A", 2, 2);
   C_MatRead(A);
   MatPrint(A);
}
```

Name

C_MatPrint — 複素行列の表示

Synopsis

```
void      C_MatPrint(mat)
Matrix    mat;
```

Arguments

mat	画面に表示する複素行列

Description

mat 複素行列を画面に表示する。

Return

なし

See Also

MatPrint, MatAllPrint

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   MatSetName(B, "B");
   C_MatPrint(B);
}
```


Name

C_MatSwap — 2 つの複素行列行列の成分の交換

Synopsis

```
void      C_MatSwap(mat1, matg2)
Matrix    mat1, mat2;
```

Arguments

mat1	成分交換を行なう複素行列 1
mat2	成分交換を行なう複素行列 2

Description

mat1 と mat2 の成分を交換する。

Return

なし

See Also

MatSwap

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D;

   MatInit();
   A = MatDef("A", 2, 2); B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   MatSetValue(MatSetValue(B, 1, 1, 5.0), 1, 2, 6.0);
   MatSetValue(MatSetValue(B, 2, 1, 7.0), 2, 2, 8.0);
   C = MatRealAndImag(A, A); MatSetName(C, "C");
   D = MatRealAndImag(B, B); MatSetName(D, "D");
   C_MatSwap(C, D);
   MatPrint(C); MatPrint(D);
}
```

Chapter 8

複素行列メモリ割り付け

Name

C_MatDef — 名前付き複素行列のメモリ割り付け

Synopsis

```
Matrix  C_MatDef(name, rows, cols)
char    *name;
int     rows, cols;
```

Arguments

name	設定する行列の名前
rows	行の大きさ
cols	列の大きさ

Description

複素行列のメモリ割り付けを行ない、名前をつける。

Return

メモリ割り付けを行なった行列

See Also

MatDef, C_MatNonNameDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = C_MatDef("B", 2, 2);
    Mat_RealAndImag(B, A, A);
    MatPrint(B);
}
```

Name

C_MatNonNameDef — 名前なし複素行列のメモリ割り付け

Synopsis

```
Matrix    C_MatNonNameDef(rows, cols)
int       rows, cols;
```

Arguments

rows	行の大きさ
cols	列の大きさ

Description

複素行列のメモリ割り付けを行なう。

Return

メモリ割り付けを行なった行列

See Also

MatDef, C_MatDef, MatNonNameDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = C_MatNonNameDef(2, 2);
    Mat_RealAndImag(B, A, A);
    MatSetName(B, "B");
    MatPrint(B);
}
```


Name

C_MatDiagDef — 複素対角行列の定義

Synopsis

```
Matrix    C_MatDiagDef(size, elm1, elm2, ...)
int       size;
Complex   elm1, elm2, ...;
```

Arguments

size	対角成分の数
elm1	第（１、１）成分の複素数
elm2	第（２、２）成分の複素数

Description

複素対角行列を定義する。

Return

定義された行列

See Also

C_MatDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    Complex a, b, c;

    MatInit();
    a = CompDef(1.0, 2.0);
    b = CompDef(3.0, 4.0);
    c = CompDef(5.0, 6.0);
    A = C_MatDiagDef(3, a, b, c);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

C_MatVander — 複素バンデルモンデ行列の定義

Synopsis

```
Matrix    C_MatVander(size, elm1, elm2, ...)
int       size;
Complex   elm1, elm2, ...;
```

Arguments

size	成分の数
elm1	第 1 成分の複素数
elm2	第 2 成分の複素数

Description

複素バンデルモンデ行列を定義する。

Return

定義された行列

See Also

C_MatDiagDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    Complex a, b, c;

    MatInit();
    a = CompDef(1.0, 2.0);
    b = CompDef(3.0, 4.0);
    c = CompDef(5.0, 6.0);
    A = C_MatVander(3, a, b, c);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

C_MatIDef — 複素単位行列の定義

Synopsis

```
Matrix    C_MatIDef(size)
size      size;
```

Arguments

int	行列の大きさ

Description

複素単位行列を定義する。

Return

定義された行列

See Also

C_MatDef, C_MatOneDef, C_MatZDef

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = C_MatIDef(2);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

C_MatIDef2 — 対角要素が 1 である複素行列の定義

Synopsis

```
Matrix  C_MatIDef2(column, row)
int     column;
int     row;
```

Arguments

column	列の大きさ
row	行の大きさ

Description

対角要素が 1 である複素行列の定義

Return

定義された行列

See Also

C_MatIDef, C_MatOneDef, C_MatZDef2

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = C_MatIDef2(2, 3);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

C_MatOneDef — 1 で満たされた複素行列の定義

Synopsis

```
Matrix  C_MatOneDef(column, row)
int     column;
int     row;
```

Arguments

column	列の大きさ
row	行の大きさ

Description

1 で満たされた複素行列の定義

Return

定義された行列

See Also

C_MatIDef, C_MatIDef2, C_MatZDef2

Example

```
#include <matrix.h>

int main()
{
    Matrix A;

    MatInit();
    A = C_MatOneDef(2, 3);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Name

C_MatIDef2 — 定数で満たされた複素行列の定義

Synopsis

```
Matrix    C_MatFillDef(column, row, cmp)
int       column;
int       row;
Complex   cmp;
```

Arguments

column	列の大きさ
row	行の大きさ
cmp	複素定数

Description

定数で満たされた複素行列の定義

Return

定義された行列

See Also

C_MatIDef2, C_MatZDef2

Example

```
#include <matrix.h>

int main()
{
   Matrix A;
   Complex c;

   MatInit();
   c = CompDef(1.0, 2.0);
   A = C_MatFillDef(2, 3, c);
   MatSetName(A, "A");
   MatPrint(A);
}
```


Name

C_MatZDef — 正方の複素零行列の定義

Synopsis

```
Matrix    C_MatZDef(size)
int       size;
```

Arguments

size	複素零行列の大きさ

Description

size × size の複素零行列を定義する。

Return

定義された複素零行列

See Also

C_MatDef, C_MatZDef2

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = C_MatZDef(3);
   MatSetName(A, "A");
   MatPrint(A);
}
```

Name

C_MatZDef2 — 非正方の複素零行列の定義

Synopsis

```
Matrix  C_MatZDef2(rows, cols)
int     rows, cols;
```

Arguments

rows	rows 行
cols	cols 列

Description

rows × cols の複素零行列を定義する。

Return

定義された複素零行列

See Also

C_MatDef, C_MatZDef

Example

```
#include <matrix.h>

int main()
{
   Matrix A;

   MatInit();
   A = C_MatZDef2(2, 3);
   MatSetName(A, "A");
   MatPrint(A);
}
```


Name

C_MatSameDef — 同じサイズの複素行列を格納するためのメモリ割り付け

Synopsis

```
Matrix  C_MatSameDef(mat)
Matrix  mat;
```

Arguments

mat	サイズの対象となる行列

Description

mat 行列と同じサイズのメモリ割り付けを行なう。

Return

格納領域へ (すべての成分が零)

See Also

C_MatDef

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatSameDef(B);
   MatCopy(C, B);
   MatSetName(C, "C");
   MatPrint(C);
}
```

Name

C_MatTransDef — 複素行列の転置行列を格納するためのメモリ割り付け

Synopsis

```
Matrix    C_MatTransDef(mat)
Matrix    mat;
```

Arguments

mat	サイズの対象となる行列

Description

mat 行列の転置行列と同じサイズのメモリ割り付けを行なう。

Return

格納領域へ

See Also

C_MatDef

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   B = MatRealAndImag(A, A);
   C = C_MatTransDef(B);
   C_Mat_Trans(C, B);
   MatSetName(C, "C");
   MatPrint(C);
}
```


Name

C_MatMulDef — 複素行列の行列の乗算結果を格納するためのメモリ割り付け

Synopsis

```
Matrix    C_MatMulDef(mat1, mat2)
Matrix    mat1, mat2;
```

Arguments

mat1	乗算を行なう行列 1
mat2	乗算を行なう行列 2

Description

複素行列の乗算結果を格納するためのメモリ割り付けを行なう。

Return

格納領域へ

See Also

C_MatDef

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatRealAndImag(A, A);
   D = C_MatMulDef(B, C);
   Mat_Mat(D, B, C);
   MatSetName(D, "D"); MatPrint(D);
}
```

Name

C_MatColumnVec — 複素縦ベクトルの定義

Synopsis

```
Matrix   C_MatColumnVec(size, elm1, elm2, ...)
int      size;
Complex  elm1, elm2, ...;
```

Arguments

size	ベクトルの大きさ
elm1	第 1 成分の複素数
elm2	第 2 成分の複素数

Description

複素縦ベクトルを定義する。

Return

定義されたベクトル

See Also

C_MatDef, C_MatRowVec

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    Complex a, b, c;

    MatInit();
    a = CompDef(1.0, 2.0);
    b = CompDef(3.0, 4.0);
    c = CompDef(5.0, 6.0);
    A = C_MatColumnVec(3, a, b, c);
    MatSetName(A, "A");
    MatPrint(A);
}
```


Name

C_MatRowVec — 複素横ベクトルの定義

Synopsis

```
Matrix   C_MatRowVec(size, elm1, elm2, ...)
int      size;
Complex  elm1, elm2, ...;
```

Arguments

size	ベクトルの大きさ
elm1	第 1 成分の複素数
elm2	第 2 成分の複素数

Description

複素横ベクトルを定義する。

Return

定義されたベクトル

See Also

C_MatDef, C_MatColumnVec

Example

```
#include <matrix.h>

int main()
{
    Matrix A;
    Complex a, b, c;

    MatInit();
    a = CompDef(1.0, 2.0);
    b = CompDef(3.0, 4.0);
    c = CompDef(5.0, 6.0);
    A = C_MatRowVec(3, a, b, c);
    MatSetName(A, "A");
    MatPrint(A);
}
```

Chapter 9

高レベル複素行列演算

Name C_MatDeterm — 複素行列の行列式

Synopsis

```
Complex   C_MatDeterm(mat)
Matrix    mat;
```

Arguments

mat	複素行列

Description mat の行列式を計算する。

Return mat の行列式

See Also MatDeterm

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;
    Complex det;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatRealAndImag(A, A);
    det = C_MatDeterm(B);
    printf("Dterminat of B = ");
    CompPrint(det);
}
```

Name

C_MatTrace — 複素行列のトレース

Synopsis

```
Complex   C_MatTrace(mat)
Matrix    mat;
```

Arguments

mat	複素行列

Description

mat のトレースを計算する。

Return

mat のトレース

See Also

MatTrace

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;
    Complex trace;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatRealAndImag(A, A);
    trace = C_MatTrace(B);
    printf("Trace of B = ");
    CompPrint(trace);
}
```


Name

C_MatMaxElem — 複素行列の最大要素

Synopsis

```
Complex   C_MatMaxElem(mat)
Matrix    mat;
```

Arguments

mat	複素行列

Description

複素行列の最大要素を取り出す。

Return

複素行列の最大要素

See Also

MatMaxElem, C_MatMinElem

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;
    Complex max;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatRealAndImag(A, A);
    max = C_MatMaxElem(B);
    printf("Maximum element of B = ");
    CompPrint(max);
}
```

Name

C_MatMinElem — 複素行列の最小要素

Synopsis

```
Complex  C_MatMinElem(mat)
Matrix   mat;
```

Arguments

mat	複素行列

Description

複素行列の最小要素を取り出す。

Return

複素行列の最小要素

See Also

MatMinElem, C_MatMaxElem

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;
   Complex min;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   min = C_MatMinElem(B);
   printf("Minimum element of B = ");
   CompPrint(min);
}
```


Name

C_MatSumElem — 複素行列の要素の合計

Synopsis

```
Complex  C_MatSumElem(mat)
Matrix   mat;
```

Arguments

mat	複素行列

Description

複素行列の成分の合計を計算する。

Return

複素行列の成分の合計

See Also

MatSumElem

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;
   Complex sum;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   sum = C_MatSumElem(B);
   printf("Sum of Bfls elements = ");
   CompPrint(sum);
}
```

Name

C_MatMeanElem — 複素行列の要素の平均値

Synopsis

```
Complex   C_MatMeanElem(mat)
Matrix    mat;
```

Arguments

mat	複素行列

Description

複素行列の成分の平均値を計算する。

Return

複素行列の成分の平均値

See Also

MatMeanElem

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;
    Complex mean;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatRealAndImag(A, A);
    mean = C_MatMeanElem(B);
    printf("Mean of Bfls elements = ");
    CompPrint(mean);
}
```


Name

C_MatGetValue C_MatValue — 複素行列の要素

Synopsis

```
Complex   C_MatGetMat(mat, column, row)
Complex   C_MatValue(mat, column, row)
Matrix    mat;
int       column, row;
```

Arguments

mat	複素行列
column	column 行
row	row 列

Description

mat の column 行 row 列の要素を取り出す。

Return

mat の column 行 row 列の要素

See Also

C_MatSetValue, MatGetValue, MatValue

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B;
    Complex elm;

    MatInit();
    A = MatDef("A", 1, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    B = MatRealAndImag(A, A);
    elm = C_MatGetValue(B, 1, 1);
    printf("B(1,1) = "); CompPrint(elm);
}
```

Name

C_MatSetValue — 複素行列の成分の代入

Synopsis

```
Complex   C_MatSetMat(mat, column, row, elm)
Matrix    mat;
int       column, row;
Complex   elm;
```

Arguments

mat	複素行列
column	column 行
row	row 列
elm	複素数

Description

mat の column 行 row 列に成分 (複素数) を代入する。

Return

成分を代入した結果の行列

See Also

C_MatGetValue, MatSetValue

Example

```
#include <matrix.h>

int main()
{
   Matrix A;
   Complex elm1;

   MatInit();
   A = C_MatDef("A", 1, 1);
   elm1 = CompDef(1.0, 2.0);
   C_MatSetValue(A, 1, 1, elm1);
   MatPrint(A);
}
```


Name

C_MatGetPtr C_MatPtr — 複素行列の成分へ

Synopsis

```
Complex   C_MatGetPtr(mat, column, row)
Complex   C_MatPtr(mat, column, row)
Matrix    mat;
int       column, row;
```

Arguments

mat	複素行列
column	column 行
row	row 列

Description

複素行列の要素ーを返す。

Return

複素行列の要素ー

See Also

MatGetPtr

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   printf("B(1,1) = ");
   CompPrint(C_MatGetPtr(B));
}
```

Name

C_MatRealPart — 複素行列の実部行列

Synopsis

```
Matrix  C_MatRealPart(mat)
Matrix  mat;
```

Arguments

mat	複素行列

Description

mat の実部行列を返す。

Return

実部行列

See Also

MatRealPart, C_MatImagPart

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatRealPart(B);
   MatPrint(C);
}
```


Name

C_MatImagPart — 複素行列の虚部行列

Synopsis

```
Matrix  C_MatImagPart(mat)
Matrix  mat;
```

Arguments

mat	複素行列

Description

mat の虚部行列を取り出して返す。

Return

虚部行列

See Also

MatImagPart, C_MatRealPart

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatImagPart(B);
   MatPrint(C);
}
```

Chapter 10

低レベル複素行列演算

Name

C_Mat_Add — 複素行列の和

Synopsis

```
Matrix  C_Mat_Add(ans, mat1, mat2)
Matrix  ans, mat1, mat2;
```

Arguments

ans	複素行列
mat1	複素行列
mat2	複素行列

Description

mat1 と mat2 の和を計算して，ans に代入する。

Return

ans 行列

See Also

MatAdd, Mat_Add

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B, C, D;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatRealAndImag(A, A);
    C = MatRealAndImag(A, A);
    D = C_MatDef("D", 2, 2);
    C_Mat_Add(D, B, C);
    MatPrint(D);
}
```

Name

C_Mat_Sub — 複素行列の差

Synopsis

```
Matrix    C_Mat_Sub(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

asns	複素行列
mat1	複素行列
mat2	複素行列

Description

mat1 と mat2 の差を計算して， ans に代入する。

Return

ans 行列

See Also

MatSub, Mat_Sub

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatRealAndImag(A, A);
   D = C_MatDef("D", 2, 2);
   C_Mat_Sub(D, B, C);
   MatPrint(D);
}
```


Name

C_Mat_Mul — 複素行列の積

Synopsis

```
Matrix    C_Mat_Mul(ans, mat1, mat2)
Matrix    ans, mat1, mat2;
```

Arguments

ans	複素行列
mat1	複素行列
mat2	複素行列

Description

mat1 と mat2 の積を計算して， ans に代入する。

Return

ans 行列

See Also

MatMul, Mat_Mul

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatRealAndImag(A, A);
   D = C_MatDef("D", 2, 2);
   C_Mat_Mul(D, B, C);
   MatPrint(D);
}
```

Name

C_Mat_MulElem — 複素行列の成分毎の積

Synopsis

```
Matrix  C_Mat_MulElem(ans, mat1, mat2)
Matrix  ans, mat1, mat2;
```

Arguments

ans	複素行列
mat1	複素行列
mat2	複素行列

Description

mat1 と mat2 の成分毎の積を計算して， ans に代入する。

Return

ans 行列

See Also

MatMulElem, Mat_MulElem

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatRealAndImag(A, A);
   D = C_MatDef("D", 2, 2);
   C_Mat_MulElem(D, B, C);
   MatPrint(D);
}
```


Name

C_Mat_Inv — 複素逆行列

Synopsis

```
Matrix  C_Mat_Inv(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の逆行列を計算して， ans に代入する。

Return

ans 行列

See Also

MatInv, Mat_Inv

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Inv(C, B);
   MatPrint(C);
}
```

Name

C_Mat_InvElem — 複素逆行列

Synopsis

```
Matrix  C_Mat_InvElem(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

逆数を成分とする行列を計算して， ans に代入する。

Return

ans 行列

See Also

MatInvElem, Mat_InvElem

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_InvElem(C, B);
   MatPrint(C);
}
```


Name

C_Mat_PseudoInv — 複素擬似逆行列

Synopsis

```
Matrix  C_Mat_PseudoMul(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の擬似逆行列を計算して， ans に代入する。

Return

ans 行列

See Also

MatPseudoInv, Mat_PseudoInv

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 1, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 1);
   C_Mat_PseudoInv(C, B);
   MatPrint(C);
}
```

Name

C_Mat_Pow — 複素行列の累乗

Synopsis

```
Matrix    C_Mat_Pow(ans, mat, index)
Matrix    ans, mat;
int       index;
```

Arguments

ans	複素行列
mat	複素行列
index	指数

Description

mat の index 乗を計算して， ans に代入する。

Return

ans 行列

See Also

MatPow, Mat_Pow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Pow(C, B, 3);
   MatPrint(C);
}
```


Name

C_Mat_Scale — 複素行列の複素数倍

Synopsis

```
Matrix    C_Mat_Scale(ans, mat, scale)
Matrix    ans, mat;
Complex   scale;
```

Arguments

ans	複素行列
mat	複素行列
scale	複素数スケール

Description

mat の scale 倍を計算して， ans に代入する。

Return

ans 行列

See Also

MatScale, MatScaleC, Mat_Scale, C_Mat_ScaleC

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;
   Complex scale;

   MatInit();
   scale = CompDef(1.0, 2.0);
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Scale(C, B, scale);
   MatPrint(C);
}
```

Name

C_Mat_Scale2 — 複素行列の実数倍

Synopsis

```
Matrix    C_Mat_Scale2(ans, mat, scale)
Matrix    ans, mat;
double    scale;
```

Arguments

ans	複素行列
mat	複素行列
scale	実数スケール

Description

mat の scale 倍を計算して， ans に代入する。

Return

ans 行列

See Also

MatScale, MatScaleC, Mat_Scale, C_Mat_Scale

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;
   double scale;

   MatInit();
   scale = 2.0;
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Scale2(C, B, scale);
   MatPrint(C);
}
```


Name

C_Mat_Trans — 複素転置行列

Synopsis

```
Matrix    C_Mat_Trans(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の転置行列を求める。

Return

ans 行列

See Also

MatTrans, Mat_Trans

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Trans(C, B);
   MatPrint(C);
}
```

Name

C_Mat_FlipLR — 複素左右反転行列

Synopsis

```
Matrix  C_Mat_FlipLR(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の左右反転行列を求める。

Return

ans 行列

See Also

MatFlipLR, Mat_FlipLR

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_FlipLR(C, B);
   MatPrint(C);
}
```


Name

C_Mat_FlipUD — 複素上下反転行列

Synopsis

```
Matrix  C_Mat_FlipUD(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の上下反転行列を求める。

Return

ans 行列

See Also

MatFlipUD, Mat_FlipUD

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_FlipUD(C, B);
   MatPrint(C);
}
```

Name

C_Mat_Conj — 複素共役行列

Synopsis

Matrix C_Mat_Conj(ans, mat)
Matrix ans, mat;

Arguments

ans	複素行列
mat	複素行列

Description

mat の共役行列を求める。

Return

ans 行列

See Also

MatConj

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Conj(C, B);
   MatPrint(C);
}
```


Name

C_Mat_ConjTrans — 複素共役転置行列

Synopsis

Matrix C_Mat_ConjTrans(ans, mat)
Matrix ans, mat;

Arguments

ans	複素行列
mat	複素行列

Description

mat の共役転置行列を求める。

Return

ans 行列

See Also

MatConjTrans

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_ConjTrans(C, B);
   MatPrint(C);
}
```

Name

C_Mat_Negate — 複素行列の符合反転

Synopsis

```
Matrix    C_Mat_Negate(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の成分の符合を反転して， ans に代入する。

Return

ans 行列

See Also

MatNegate, Mat_Negate

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Negate(C, B);
   MatPrint(C);
}
```


Name

C_Mat_Cut — 複素ブロック行列の取り出し

Synopsis

```
Matrix    C_Mat_Cut(ans, mat, sc, sr, ec, er)
Matrix    ans, mat;
int       sc, sr, ec, er;
```

Arguments

ans	複素行列
mat	複素行列
sc	左上の行番号
sr	左上の列番号
ec	右下の行番号
er	右下の列番号

Description

(sc, sr) を左上， (ec, sr) を右下とするブロック行列を取り出して， ans に代入する。

Return

ans 行列

See Also

MatCut, Mat_Cut

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDiagDef(4, 1.0, 2.0, 3.0, 4.0);
   B = MatRealAndImag(A, A);
   C = C_MatDef("C", 2, 2);
   C_Mat_Cut(C, B, 2, 2, 3, 3); MatPrint(C);
}
```

Name

C_Mat_Put — 複素ブロック行列の代入

Synopsis

```
Matrix  C_Mat_Put(mat1, sc, sr, mat2)
Matrix  mat1, mat2;
int     sc, sr;
```

Arguments

mat1	複素行列
mat2	（実 / 複素）行列
sc	左上の行番号
sr	左上の列番号

Description

(sc, sr) が左上になるようにブロック行列を代入する。

Return

mat1 行列

See Also

MatPut, Mat_Put

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;
   Complex a, b;

   MatInit();
   a = CompDef(1.0, 2.0);
   b = CompDef(3.0, 4.0);
   A = C_MatDef("A", 4, 4);
   B = C_MatDiagDef(2, a, b);
   C_Mat_Put(A, 2, 2, B);
   MatPrint(A);
}
```


Name

C_Mat_VectorProduct — 複素ベクトル積

Synopsis

```
Matrix  C_Mat_VectorProduct(ans, mat1, mat2)
Matrix  ans, mat1, mat2;
```

Arguments

ans	3×1 の複素行列
mat1	3×1 の複素行列
mat2	3×1 の複素行列

Description

mat1 と mat2 のベクトル積を計算して，ans に代入する。

Return

ans 行列

See Also

MatVectorProduct, Mat_VectorProduct

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;
   Complex a, b;;

   MatInit();
   A = C_MatDef("A", 3, 1);
   B = MatSameDef(A); C = MatSameDef(A);
   a = CompDef(1.0, 2.0);
   b = CompDef(3.0, 4.0);
   C_MatSetValue(A, 1, 1, a);
   C_MatSetValue(A, 2, 1, b);
   MatCopy(B, A);
   C_Mat_VectorProd(C, A, B); MatPrint(C);
}
```

Name

C_Mat_Eig — 一般複素行列の固有値 / 固有ベクトルの計算

Synopsis

```
void    C_Mat_Eig(mat, val, vec)
Matrix  mat, val, vec;
```

Arguments

mat	複素行列
val, vec	複素行列

Description

一般行列の固有値 / 固有ベクトルを計算して，
val と vec に代入する。

Return

なし

See Also

MatEig, Mat_Eig

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, val, vec;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   val = MatEigValDef(B); vec = MatEigVecDef(B);
   C_Mat_Eig(B, val, vec);
   MatPrint(val); MatPrint(vec);
}
```


Name

C_Mat_HermitEig — エルミート行列の固有値 / 固有ベクトルの計算

Synopsis

```
void    C_Mat_HermitEig(mat, val, vec)
Matrix  mat, val, vec;
```

Arguments

mat	エルミート行列
val, vec	複素行列

Description

エルミート行列の固有値 / 固有ベクトルを計算して，
val と vec に代入する。

Return

なし

See Also

MatHermitEig

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, val, vec;

   MatInit();
   A = MatIDef(2); B = MatDef("B", 2, 2);
   MatSetValue(MatSetValue(B, 1, 1, 0.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(B, 2, 1,−2.0), 2, 2, 0.0);
   C = MatRealAndImag(A, B);
   val = MatEigValDef(C); vec = MatEigVecDef(C);
   C_Mat_HermitEig(C, val, vec);
   MatPrint(val); MatPrint(vec);
}
```

Name

C_Mat_EigVal — 一般複素行列の固有値

Synopsis

```
Matrix  C_Mat_EigVal(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の固有値を計算して， ans に代入する。

Return

ans 行列

See Also

MatEigVal, Mat_EigVal

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatEigValDef(B);
   C_Mat_EigVal(C, B);
   MatSetName(C, "C"); MatPrint(C);
}
```


Name

C_Mat_EigVec — 一般複素行列の固有ベクトル

Synopsis

```
Matrix  C_Mat_EigVec(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の固有ベクトルを計算して， ans に代入する。

Return

ans 行列

See Also

MatEigVec, Mat_EigVec

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatEigVecDef(B);
   C_Mat_EigVec(C, B);
   MatSetName(C, "C"); MatPrint(C);
}
```

Name

C_Mat_EigValVec — 一般複素行列の固有値 / 固有ベクトル

Synopsis

```
Matrix  C_MatEigValVec(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	複素行列
mat	複素行列

Description

mat の固有値を列方向に並べた行列と
mat の固有ベクトルを列方向に並べた行列を
行方向に並べて（[val vec]）ans に代入する。

Return

ans 行列

See Also

MatEigValVec, Mat_EigValVec

Example

```
#include <matrix.h>

int main()
{
    Matrix A, B, val, vec;

    MatInit();
    A = MatDef("A", 2, 2);
    MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
    MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
    B = MatRealAndImag(A, A);
    val = MatEigValDef(B); vec = MatEigVecDef(B);
    C = MatCatRow(val, vec);
    C_Mat_EigValVec(C, B);
```



```
    MatSetName(C, "C"); MatPrint(C);
}
```

Name

C_Mat_SingVal — 複素行列の特異値

Synopsis

```
Matrix  C_Mat_SingVal(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	実行列
mat	複素行列

Description

mat の特異値を計算して， ans に代入する。

Return

ans 行列

See Also

MatSingVal, Mat_SingVal

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatDef("C", 2, 1);
   C_Mat_SingVal(C, B);
   MatPrint(C);
}
```


Name

C_Mat_CatColumn — 複素行列の列接続

Synopsis

```
Matrix  C_Mat_CatColumn(ans, mat1, mat2)
Matrix  ans, mat1, mat2;
```

Arguments

ans	複素行列
mat1	複素行列
mat2	複素行列

Description

列方向に行列を接続して， ans に代入する。
[mat1 mat2]′

Return

ans 行列

See Also

MatCatColumn, Mat_CatColumn

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A)
   C = MatRealAndImag(A, A)
   D = C_MatDef("D", 4, 2);
   C_Mat_CatColumn(D, B, C);
   MatPrint(D);
}
```

Name

C_Mat_CatRow — 複素行列の行接続

Synopsis

```
Matrix  C_Mat_CatRow(ans, mat1, mat2)
Matrix  ans, mat1, mat2;
```

Arguments

ans	複素行列
mat1	複素行列
mat2	複素行列

Description

行方向に行列を接続して， ans に代入する。
[mat1 mat2]

Return

ans 行列

See Also

MatCatRow, Mat_CatRow

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C, D;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatRealAndImag(A, A);
   D = C_MatDef("D", 2, 4);
   C_Mat_CatRow(D, B, C);
   MatPrint(D);
}
```


Name

C_Mat_RealPart — 複素行列の実部行列

Synopsis

```
Matrix  C_Mat_RealPart(ans, mat)
Matrix  ans, mat;
```

Arguments

ans	実行列
mat	複素行列

Description

mat の実部行列を取り出して， ans に代入する。

Return

ans 行列

See Also

MatRealPart, C_Mat_ImagPart

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatDef("C", 2, 2);
   C_Mat_RealPart(C, B);
   MatPrint(C);
}
```

Name

C_Mat_ImagPart — 複素行列の虚部行列

Synopsis

```
Matrix    C_Mat_ImagPart(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	実行列
mat	複素行列

Description

mat の虚部行列を取り出して， ans に代入する。

Return

ans 行列

See Also

MatImagPart, C_Mat_RealPart

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDef("A", 2, 2);
   MatSetValue(MatSetValue(A, 1, 1, 1.0), 1, 2, 2.0);
   MatSetValue(MatSetValue(A, 2, 1, 3.0), 2, 2, 4.0);
   B = MatRealAndImag(A, A);
   C = MatDef("C", 2, 2);
   C_Mat_ImagPart(C, B);
   MatPrint(C);
}
```


Name

C_Mat_RealAndImag — 実部と虚部による複素行列の合成

Synopsis

```
Matrix    C_Mat_RealAndImag(ans, real, imag)
Matrix    ans, real, imag;
```

Arguments

ans	複素行列
real	実行列
imag	実行列

Description

real を実部，imag を虚部とする複素行列を作り，ans に代入する。

Return

ans 行列

See Also

MatRealAndImag

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B, C;

   MatInit();
   A = MatDiagDef(2, 1.0, 2.0);
   B = MatDiagDef(2, 3.0, 4.0);
   C = C_MatDDef("C", 2, 2);
   C_Mat_RealAndImag(C, A, B);
   MatPrint(C);
}
```

Name

C_Mat_RealToComp — 実行列から複素行列への変換

Synopsis

```
Matrix    C_Mat_RealToComp(ans, mat)
Matrix    ans, mat;
```

Arguments

ans	複素行列
mat	実行列

Description

mat を複素行列に変換して，ans に代入する。

Return

ans 行列

See Also

MatRealToComp

Example

```
#include <matrix.h>

int main()
{
   Matrix A, B;

   MatInit();
   A = MatDiagDef(2, 1.0, 2.0);
   B = C_MatDef("B", 2, 2);
   C_Mat_RealToComp(B, A);
   MatPrint(B);
}
```

索 引

C_Mat_Add, 220
C_Mat_CatColumn, 246
C_Mat_CatRow, 247
C_Mat_Conj, 233
C_Mat_ConjTrans, 234
C_Mat_Cut, 236
C_Mat_Eig, 239
C_Mat_EigVal, 241
C_Mat_EigValVec, 243
C_Mat_EigVec, 242
C_Mat_FlipLR, 231
C_Mat_FlipUD, 232
C_Mat_HermitEig, 240
C_Mat_ImagPart, 249
C_Mat_Inv, 224
C_Mat_InvElem, 225
C_Mat_Mul, 222
C_Mat_MulElem, 223
C_Mat_Negate, 235
C_Mat_Pow, 227
C_Mat_PseudoInv, 226
C_Mat_Put, 237
C_Mat_RealAndImag, 250
C_Mat_RealPart, 248
C_Mat_RealToComp, 251
C_Mat_Scale, 228
C_Mat_Scale2, 229
C_Mat_SingVal, 245
C_Mat_Sub, 221
C_Mat_Trans, 230
C_Mat_VectorProduct, 238
C_MatColumnVec, 205
C_MatCopy, 187
C_MatDef, 192
C_MatDeterm, 208
C_MatDiagDef, 194
C_MatFillDef, 199
C_MatGetPtr, 216
C_MatGetValue, 214
C_MatIDef, 196
C_MatIDef2, 197
C_MatImagPart, 218
C_MatInput, 186
C_MatMaxElem, 210
C_MatMeanElem, 213
C_MatMinElem, 211
C_MatMulDef, 204
C_MatNonNameDef, 193
C_MatOneDef, 198
C_MatPrint, 189

C_MatPtr, 216
C_MatRead, 188
C_MatRealPart, 217
C_MatRowVec, 206
C_MatSameDef, 202
C_MatScalarProduct, 77
C_MatSetValue, 215
C_MatSumElem, 212
C_MatSwap, 190
C_MatTrace, 209
C_MatTransDef, 203
C_MatValue, 214
C_MatVander, 195
C_MatZDef, 200
C_MatZDef2, 201
Cols, 137

Mat_Add, 160
Mat_Cat4, 175
Mat_CatColumn, 173
Mat_CatRow, 174
Mat_Cut, 176
Mat_Eig, 179
Mat_EigVal, 180
Mat_EigValVec, 182
Mat_EigVec, 181
Mat_FlipUD, 170, 171
Mat_Inv, 164
Mat_InvElem, 165
Mat_Mul, 162
Mat_MulElem, 163
Mat_Negate, 172
Mat_Pow, 167
Mat_PseudoInv, 166
Mat_Put, 177
Mat_Scale, 168
Mat_SingVal, 183
Mat_Sub, 161
Mat_Trans, 169
Mat_VectorProduct, 178
MatAdd, 56
MatAllDestroy, 29
MatAllFree, 26
MatAllPrint, 23
MatAssign, 14
MatBlockDiag, 126
MatCalss, 130
MatCat4, 112
MatCatColumn, 108
MatCatColumns, 109
MatCatRow, 110
MatCatRows, 111
MatChangeColumn, 15
MatChangeRow, 16
MatClassSameDef, 47
MatCols, 33, 137
MatColumnVec, 52
MatColumnVector, 114
MatCompanion, 127
MatCompanion2, 128
MatCompToPoly, 123
MatCompToRat, 124
MatConj, 73
MatConjTrans, 74

MatCopy, 11
MatCut, 113
MatDef, 36
MatDestroy, 27
MatDestroys, 28
MatDeterm, 91
MatDiagDef, 38
MatDiagToVec, 79
MatDup, 12
MatEdit, 20
MatEig, 80
MatEigVal, 82
MatEigValDef, 50
MatEigValVec, 84
MatEigVec, 83
MatEigVecDef, 51
MatError, 31
MatExp, 72
MatFileRead, 17
MatFileWrite, 18
MatFlipLR, 69
MatFlipUD, 70
MatFree, 24, 25
MatFrobNorm, 92
MatGetClass, 130
MatGetName, 134
MatGetPtr, 103
MatGetType, 132
MatGetValue, 101
MatHermitEig, 81
MatIDef, 40
MatIDef2, 41, 43
MatImagPart, 118
MatInfNorm, 93
MatInit, 10
MatInput, 19
MatInv, 60
MatInvElem, 61
MatIsColumnVec, 157
MatIsComplex, 139
MatIsEqual, 155
MatIsFullRank, 142
MatIsNegative, 150
MatIsNegativeSemi, 151
MatIsNonEqual, 156
MatIsNonNegative, 152
MatIsNonPositive, 149
MatIsNonSingular, 144
MatIsNonSquare, 146
MatIsNonZero, 154
MatIsPositive, 147
MatIsPositiveSemi, 148
MatIsReal, 138
MatIsRowVec, 158
MatIsSingular, 143
MatIsSquare, 145
MatIsTmp, 141
MatIsVar, 140
MatIsZero, 153
MatKernel, 105
MatMaxElem, 95
MatMaxSingVal, 99
MatMeanElem, 98
MatMinElem, 96

MatMinSingVal, 100
MatMul, 58
MatMulDef, 49
MatMulElem, 59
MatName, 134
MatNegate, 71
MatNonNameDef, 37
MatOneDef, 42
MatPolyToRat, 125
MatPow, 63
MatPrint, 22
MatPseudoInv, 62
MatPtr, 103
MatPut, 116
MatRank, 104
MatRealAndImag, 119
MatRealPart, 117
MatRealToComp, 120
MatRealToPoly, 121
MatRealToRat, 122
MatRows, 32, 136
MatRowVec, 53
MatRowVector, 115
MatSameDef, 46
MatScalarProduct, 76
MatScale, 64
MatScaleC, 65
MatScaleP, 66
MatScaleR, 67
MatSearch, 21
MatSetClass, 131
MatSetName, 135
MatSetType, 133
MatSetValue, 102
MatSingLeftVec, 88
MatSingRightVec, 89
MatSingVal, 85
MatSingValVec, 87
MatSingVec, 86
MatSqrt, 90
MatSub, 57
MatSumElem, 97
MatSwap, 13
MatTmpFree, 30
MatTrace, 94
MatTrans, 68
MatTransDef, 48
MatType, 132
MatValue, 101
MatVander, 39
MatVecToDiag, 78
MatVectorProduct, 75
MatZDef, 44
MatZDef2, 45

Rows, 136

＜著者略歴＞

古賀 雅伸

1988 年　東京工業大学工学部制御工学科卒業
1993 年　博士 (工学)
1993 年　東京工業大学工学部助手
現　在　東京工業大学情報理工学研究科助手

Matlib Reference Manual

1995 年 1 月 20 日　　初　版　発　行

執筆者　古賀 雅伸

発行者　古賀 雅伸

印刷者　古賀 雅伸

製本者　古賀 雅伸

発行所　東京工業大学
〒 152 東京都目黒区大岡山 2-12-1
電話 (03)5734-2328

本書の内容を無断で複写生成することは，著作者の権利を侵害することがありますので，その場合にはあらかじめ著作者あて許諾をお求め下さい。