

My Project

Generated by Doxygen 1.8.13

Contents

1	Modules Index	1
1.1	Modules List	1
2	File Index	3
2.1	File List	3
3	Module Documentation	5
3.1	olis_f90stdlib Module Reference	5
3.1.1	Function/Subroutine Documentation	5
3.1.1.1	alloc_complex_eigenvects()	6
3.1.1.2	alloc_complex_svd()	6
3.1.1.3	c_inv2()	6
3.1.1.4	complex_eigenvects()	7
3.1.1.5	complex_svd()	7
3.1.1.6	complextrace()	8
3.1.1.7	matrixnorm()	8
3.1.1.8	outerproduct()	8
3.1.1.9	printvectors()	9
3.1.1.10	randseed()	9
4	File Documentation	11
4.1	olis_f90stdlib.f90 File Reference	11
	Index	13

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

olis_f90stdlib	5
--	---

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

olis_f90stdlib.f90	11
------------------------------------	-------	----

Chapter 3

Module Documentation

3.1 olis_f90stdlib Module Reference

Functions/Subroutines

- subroutine [alloc_complex_eigenvects](#) (matrix, eigenvals, u, v)
allocates eigenvals, u & v arrays for eigenvals & eigenvects
- subroutine [alloc_complex_svd](#) (matrix, sigma, u, vt)
allocates sigma (singular vals), u and vt for complexSVD allocates temp work arrays too
- subroutine [randseed](#) (seed)
generates random seed
- subroutine [printvectors](#) (vect, desc, f)
print formatted matrices can take optional args for labels or write directly to a file
- complex(kind=dp) function, dimension(2, 2) [outerproduct](#) (a, b)
outerproduct of two complex vectors, returns a complex matrix
- complex(kind=dp) function [complextrace](#) (a)
computes the trace of a complex matrix
- subroutine [complex_eigenvects](#) (a, w, vl, vr)
computes the complex eigenvalues and eigenvectors overwrites matrix in, input eigenvalue array and eigenvector arrays uses the zgeev subroutine from lapack
- subroutine [complex_svd](#) (a, sigma, u, vt)
computes the complex eigenvalues and eigenvectors overwrites matrix in, input eigenvalue array and eigenvector arrays uses the zgeev subroutine from lapack
- complex(kind=dp) function, dimension(2, 2) [c_inv2](#) (m_in)
inverse for a complex 2x2 matrix
- real(kind=dp) function [matrixnorm](#) (c)
computed Frobenius matrix norm of complex matrix using lapack zlange

3.1.1 Function/Subroutine Documentation

3.1.1.1 alloc_complex_eigenvects()

```
subroutine olis_f90stdlib::alloc_complex_eigenvects (
    complex(kind=dp), dimension(:,:), intent(in) matrix,
    complex(kind=dp), dimension(:), intent(inout), allocatable eigenvals,
    complex(kind=dp), dimension(:,:), intent(inout), allocatable u,
    complex(kind=dp), dimension(:,:), intent(inout), allocatable v )
```

allocates eigenvals, u & v arrays for eigenvals & eigenvects

allocated temp work arrays also

Author

Oliver Thomas August 2018

Parameters

<i>matrix</i>	input complex matrix
<i>eigenvals</i>	1d array for eigenvalues, is overwritten on exit
<i>u</i>	2d array of left eigenvectors
<i>v</i>	3d array of right eigenvectors

3.1.1.2 alloc_complex_svd()

```
subroutine olis_f90stdlib::alloc_complex_svd (
    complex(kind=dp), dimension(:,:), intent(in) matrix,
    real(kind=dp), dimension(:), intent(inout), allocatable sigma,
    complex(kind=dp), dimension(:,:), intent(inout), allocatable u,
    complex(kind=dp), dimension(:,:), intent(inout), allocatable vt )
```

allocates sigma (singular vals), u and vt for complexSVD allocates temp work arrays too

Parameters

<i>matrix</i>	input complex matrix
<i>sigma</i>	real vector of singular values sorted in descending order
<i>u</i>	unitary matrix
<i>vt</i>	unitary matrix returns $V^{**}H$ NOT v

3.1.1.3 c_inv2()

```
complex(kind=dp) function, dimension(2,2) olis_f90stdlib::c_inv2 (
    complex(kind=dp), dimension(2,2), intent(in) m_in )
```

inverse for a complex 2x2 matrix

Parameters

$m \leftrightarrow$ _in	is input complex 2x2 matrix
----------------------------	-----------------------------

3.1.1.4 complex_eigenvects()

```
subroutine olis_f90stdlib::complex_eigenvects (
    complex(kind=dp), dimension(:,:), allocatable a,
    complex(kind=dp), dimension(:), allocatable w,
    complex(kind=dp), dimension(:,:), allocatable vl,
    complex(kind=dp), dimension(:,:), allocatable vr )
```

computes the complex eigenvalues and eigenvectors overwrites matrix in, input eigenvalue array and eigenvector arrays uses the zgeev subroutine from lapack

Parameters

<i>a</i>	input allocatable complex matrix to be diagonalised
<i>w</i>	output allocatable complex 1d array containing eigenvals
<i>vl</i>	output allocatable complex 2d array containing left eigenvectors
<i>vr</i>	output allocatable complex 2d array containing right eigenvectors

Note

need to check this is optimised

3.1.1.5 complex_svd()

```
subroutine olis_f90stdlib::complex_svd (
    complex(kind=dp), dimension(:,:), intent(inout), allocatable a,
    real(kind=dp), dimension(:), allocatable sigma,
    complex(kind=dp), dimension(:,:), allocatable u,
    complex(kind=dp), dimension(:,:), allocatable vt )
```

computes the complex eigenvalues and eigenvectors overwrites matrix in, input eigenvalue array and eigenvector arrays uses the zgeev subroutine from lapack

Parameters

<i>a</i>	input allocatable complex matrix to be SVD'd
<i>sigma</i>	output allocatable complex 1d array containing ordered singular values
<i>u</i>	output allocatable complex 2d array containing u
<i>vt</i>	output allocatable complex 2d array containing v**H

Note

need to check this is optimised

3.1.1.6 complextrace()

```
complex(kind=dp) function olis_f90stdlib::complextrace (
    complex(kind=dp), dimension(:, :) a )
```

computes the trace of a complex matrix

Parameters

<i>a</i>	is the complex matrix in
----------	--------------------------

3.1.1.7 matrixnorm()

```
real(kind=dp) function olis_f90stdlib::matrixnorm (
    complex(kind=dp), dimension(:, :) c )
```

computed Frobenius matrix norm of complex matrix using lapack zlange

Parameters

<i>c</i>	input complex matrix
----------	----------------------

3.1.1.8 outerproduct()

```
complex(kind=dp) function, dimension(2,2) olis_f90stdlib::outerproduct (
    complex(kind=dp), dimension(:), intent(in) a,
    complex(kind=dp), dimension(:), intent(in) b )
```

outerproduct of two complex vectors, returns a complex matrix

Parameters

<i>a</i>	is input vector 1, ket>
<i>b</i>	is input vector 2, <bra

3.1.1.9 printvectors()

```
subroutine olis_f90stdlib::printvectors (
    complex(kind=dp), dimension(:, :), intent(in) vect,
    character(len=*), intent(in), optional desc,
    integer, intent(in), optional f )
```

print formatted matrices can take optional args for labels or write directly to a file

Parameters

<i>vect</i>	is the input complex matrix
<i>desc</i>	is the optional string to be written above the matrix
<i>f</i>	is the optional file output unit to write to, default is console

3.1.1.10 randseed()

```
subroutine olis_f90stdlib::randseed (
    integer, dimension(:), allocatable seed )
```

generates random seed

Parameters

<i>seed</i>	is input allocatable 1d array
-------------	-------------------------------

Chapter 4

File Documentation

4.1 olis_f90stdlib.f90 File Reference

Modules

- module [olis_f90stdlib](#)

Functions/Subroutines

- subroutine [olis_f90stdlib::alloc_complex_eigenvects](#) (matrix, eigenvals, u, v)
allocates eigenvals, u & v arrays for eigenvals & eigenvects
- subroutine [olis_f90stdlib::alloc_complex_svd](#) (matrix, sigma, u, vt)
allocates sigma (singular vals), u and vt for complexSVD allocates temp work arrays too
- subroutine [olis_f90stdlib::randseed](#) (seed)
generates random seed
- subroutine [olis_f90stdlib::printvectors](#) (vect, desc, f)
print formatted matrices can take optional args for labels or write directly to a file
- complex(kind=dp) function, dimension(2, 2) [olis_f90stdlib::outerproduct](#) (a, b)
outerproduct of two complex vectors, returns a complex matrix
- complex(kind=dp) function [olis_f90stdlib::complextrace](#) (a)
computes the trace of a complex matrix
- subroutine [olis_f90stdlib::complex_eigenvects](#) (a, w, vl, vr)
computes the complex eigenvalues and eigenvectors overwrites matrix in, input eigenvalue array and eigenvector arrays uses the zgeev subroutine from lapack
- subroutine [olis_f90stdlib::complex_svd](#) (a, sigma, u, vt)
computes the complex eigenvalues and eigenvectors overwrites matrix in, input eigenvalue array and eigenvector arrays uses the zgeev subroutine from lapack
- complex(kind=dp) function, dimension(2, 2) [olis_f90stdlib::c_inv2](#) (m_in)
inverse for a complex 2x2 matrix
- real(kind=dp) function [olis_f90stdlib::matrixnorm](#) (c)
computed Frobenius matrix norm of complex matrix using lapack zlange

Index

- alloc_complex_eigenvecs
 - olis_f90stdlib, [5](#)
- alloc_complex_svd
 - olis_f90stdlib, [6](#)
- c_inv2
 - olis_f90stdlib, [6](#)
- complex_eigenvecs
 - olis_f90stdlib, [7](#)
- complex_svd
 - olis_f90stdlib, [7](#)
- complextrace
 - olis_f90stdlib, [8](#)
- matrixnorm
 - olis_f90stdlib, [8](#)
- olis_f90stdlib, [5](#)
 - alloc_complex_eigenvecs, [5](#)
 - alloc_complex_svd, [6](#)
 - c_inv2, [6](#)
 - complex_eigenvecs, [7](#)
 - complex_svd, [7](#)
 - complextrace, [8](#)
 - matrixnorm, [8](#)
 - outerproduct, [8](#)
 - printvectors, [8](#)
 - randseed, [9](#)
- olis_f90stdlib.f90, [11](#)
- outerproduct
 - olis_f90stdlib, [8](#)
- printvectors
 - olis_f90stdlib, [8](#)
- randseed
 - olis_f90stdlib, [9](#)