My Project

Generated by Doxygen 1.8.13

Contents

Chapter 1

Modules Index

1.1 Modules List

Here is a list of all modules with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Module Documentation

3.1 makeopticalelements Module Reference

module for building symplectic matrices for optical elements

Functions/Subroutines

- subroutine make_bs (nspace, nspec, symp_mat, m1, m2, theta)

  makes beamsplitter symplectic matrix takes in an allocated matrix for the beamsplitter matrix to be written to uses the private ident_spec, spatial_work, n_work arrays
- subroutine alloc_temparrays (nspace, nspec)

Variables

- real(kind=dp), public ident

3.1.1 Detailed Description

module for building symplectic matrices for optical elements

3.1.2 Function/Subroutine Documentation

3.1.2.1 alloc_temparrays()

```
subroutine makeopticalelements::alloc_temparrays (
             integer, intent(in) nspace,
             integer, intent(in) nspec )
```

3.1.2.2 make_bs()

```
subroutine makeopticalelements::make_bs (
             integer nspace,
             integer nspec,
             complex(kind=dp), dimension(:,:), allocatable symp_mat,
             integer m1,
             integer m2,
             real(kind=dp) theta )
```

makes beamsplitter symplectic matrix takes in an allocated matrix for the beamsplitter matrix to be written to uses the private ident_spec, spatial_work, n_work arrays

Parameters

nspace	is number of total spatial modes
nspec	is number of total spectral modes
m_bs	allocated n∗n matrix for beamsplitter
m1	is spatial mode 1 for beam splitter
m2	is spatial mode 2 for beam splitter

3.1.3 Variable Documentation

3.1.3.1 ident

```
real(kind=dp), public makeopticalelements::ident
```

3.2 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, dimension(n, n) c_identity (n)

  makes complex identity matrix dim (nxn)
- complex(kind=dp) function, dimension(:,:), allocatable tprod (a, b)

  tensor product for complex matrices aXb

- 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 Frobenieus matrix norm of complex matrix using lapack zlange

Variables

- real(kind=dp), parameter pi =4.0_dp∗atan(1.0)

3.2.1 Function/Subroutine Documentation

3.2.1.1 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

matrix	input complex matrix
eigenvals	1d array for eigenvalues, is overwriten on exit
u	2d array of left eigenvectors
v	3d array of right eigenvectors

3.2.1.2 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

matrix	input complex matrix
sigma	real vector of singular values sorted in descending order
u	unitary matrix
vt	unitary matrix returns V∗∗H NOT v

3.2.1.3 c_identity()

```
complex(kind=dp) function, dimension(n,n) olis_f90stdlib::c_identity (
            integer, intent(in) n )
```

makes complex identity matrix dim (nxn)

Parameters

n	input dimension

3.2.1.4 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_in	is input complex 2x2 matrix

3.2.1.5 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

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

Note

need to check this is optimised

3.2.1.6 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

a	input allocatable complex matrix to be SVD'd
sigma	output allocatable complex 1d array containing ordered singular values
u	output allocatable complex 2d array containing u
vt	output allocatable complex 2d array containing v∗∗H

Note

need to check this is optimised

3.2.1.7 complextrace()

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

computes the trace of a complex matrix

Parameters

a	is the complex matrix in

3.2.1.8 matrixnorm()

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

computed Frobenieus matrix norm of complex matrix using lapack zlange

Parameters

c	input complex matrix

3.2.1.9 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

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

3.2.1.10 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

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

3.2.1.11 randseed()

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

generates random seed

Parameters

seed	is input allocatable 1d array

3.2.1.12 tprod()

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

tensor product for complex matrices aXb

Parameters

a	complex matrix in
b	complex matrix in

3.2.2 Variable Documentation

3.2.2.1 pi

```
real(kind=dp), parameter olis_f90stdlib::pi =4.0_dp*atan(1.0)
```

Chapter 4

File Documentation

4.1 makeopticalelements.f90 File Reference

Modules

- module makeopticalelements

  module for building symplectic matrices for optical elements

Functions/Subroutines

- subroutine makeopticalelements::make_bs (nspace, nspec, symp_mat, m1, m2, theta)

  makes beamsplitter symplectic matrix takes in an allocated matrix for the beamsplitter matrix to be written to uses the private ident_spec, spatial_work, n_work arrays
- subroutine makeopticalelements::alloc_temparrays (nspace, nspec)

Variables

- real(kind=dp), public makeopticalelements::ident

4.2 num_hom.f90 File Reference

Functions/Subroutines

- program num_hom

  program to compute matrix of a JSA
- complex(kind=dp) function f (w1, w2, sig)

  JSA function taking two freq.

4.2.1 Function/Subroutine Documentation

4.2.1.1 f()

```
complex(kind=dp) function num_hom::f (
            real(kind=dp), intent(in) w1,
            real(kind=dp), intent(in) w2,
            real(kind=dp), intent(in) sig )
```

JSA function taking two freq.

Parameters

w1	input signal freq
w2	input idler freq
sig	input variance

4.2.1.2 num_hom()

```
program num_hom ( )
```

program to compute matrix of a JSA

4.3 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, dimension(n, n) olis_f90stdlib::c_identity (n)

  makes complex identity matrix dim (nxn)
- complex(kind=dp) function, dimension(:,:), allocatable olis_f90stdlib::tprod (a, b)

  tensor product for complex matrices aXb
- 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 Frobenieus matrix norm of complex matrix using lapack zlange

Variables

- real(kind=dp), parameter olis_f90stdlib::pi =4.0_dp∗atan(1.0)

Index