Package 'cda'

February 19, 2015

Version 1.5.1

License GPL-3

Copyright file inst/COPYRIGHTS for src/zgels.f obtained from LAPACK

Title Coupled dipole approximation of light scattering by clusters of nanoparticles

Description Solves the electromagnetic problem of coupled-dipoles (scattering and absorption by a cluster of subwavelength particles in arbitrary 3D configuration) by setting up the interaction matrix and solving the linear system for multiple incident fields. Functions are provided for linear polarisation as well as circular polarisation with full angular averaging (optical activity). Retardation is included in the dipole-dipole interaction.

URL https://github.com/baptiste/cda

Type Package

LazyLoad yes

VignetteBuilder knitr

SystemRequirements GNU make

LinkingTo Rcpp, RcppArmadillo

Depends methods, dielectric

Suggests rgl, grid, ggplot2, gridExtra, testthat, knitr

Imports Rcpp, statmod, randtoolbox, reshape2, plyr

RcppModules cda, cd, dispersion, array

Author Baptiste Auguie [aut, cre],
LAPACK authors [cph] (copyrights apply to src/zgels.f)

Maintainer Baptiste Auguie <baptiste.auguie@gmail.com>

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-01-19 20:44:59

R topics documented:

`cda-package` *cda*

Description

Coupled dipole approximation in electromagnetic scattering

Details

Solves the electromagnetic problem of coupled-dipoles (scattering and absorption by a cluster of subwavelength particles in arbitrary 3D configuration) by direct inversion of the interaction matrix. Functions are provided for linear polarisation with varying angle of incidence, as well as circular polarisation with angular averaging (optical activity). Retardation is included in the interaction.

Author(s)

baptiste Auguie

References

Draine BT. The discrete-dipole approximation and its application to interstellar graphite grains. Astrophysical Journal. 1988.

Schatz GC, Duyne RP. Discrete dipole approximation for calculating extinction and Raman intensities for small particles with arbitrary shapes. Journal of Chemical Physics. 1995.

Gunnarsson L, Zou S, Schatz GC, et al. Confined plasmons in nanofabricated single silver particle pairs: Experimental observations of strong interparticle interactions. Journal of Physical Chemistry B. 2005.

Any one of the following references should be used to cite and acknowledge the use of this package.

Circular dichroism:

B. Auguie, J.L. Alonso-Gomez, A. Guerrero-Martinez, L.M. Liz-Marzan. Fingers crossed: circular dichroism with a dimer of plasmonic nanorods. J. Phys. Chem. Lett. 2, (2011)

Linear extinction:

B. Auguie, W.L. Barnes. Diffractive coupling in gold nanoparticle arrays and the effect of disorder. Optics Letters (2009)

Array factor (infinite case):

B. Auguie, W.L. Barnes. Collective resonances in gold nanoparticle arrays. Physical Review Letters (2008)

`alpha_kuwata` *alpha_kuwata*

Description

polarizability

Usage

```
alpha_kuwata(wavelength, epsilon, V, axis, L, medium = 1.33)
```

Arguments

`wavelength`	wavelength
`epsilon`	permittivity
`V`	volume
`axis`	semi-axis along incident field
`L`	shape factor
`medium`	refractive index

Details

prescription from Kuwata

Value

polarizability

Author(s)

baptiste Auguie

References

Kuwata et al. Resonant light scattering from metal nanoparticles: Practical analysis beyond Rayleigh approximation Appl. Phys. Lett. 83, 22 (2003)

See Also

Other user_level polarizability: `inverse_polarizability`, `La`, `polarizability_ellipsoid`

`array` *Rcpp module: array*

Description

Exposes a C++ calculation of the array factor.

Details

- array_factor Truncated lattice sum for a finite 2D square array

Examples

```
show( array )
```

`array_factor` *array factor*

Description

C++ calculation of the array factor

Usage

```
array_factor(wavelength, N, pitch)
```

Arguments

`wavelength`	wavelength in nm
`N`	half number of dipoles along one side
`pitch`	pitch in nm

Details

Brute-force numerical evaluation of the truncated 2D sum of dipole fields

Value

S

Author(s)

baptiste Auguie

Examples

```
S <- array_factor(seq(400, 600),  10,  500)
str(S)
```

---

cd *Rcpp module: cd*

Description

Exposes a calculation of orientation-averaged circular dichroism within the coupled-dipole approximation.

Details

- average_spectrum Loop over wavelengths and calculate the orientation averaging of the difference in extinction, absorption, scattering for left/right circularly polarised light

Examples

```
show( cd )
```

---

cda *Rcpp module: cda*

Description

Exposes basic C++ functions used in the coupled-dipole approximation.

Details

- absorption Absorption cross-section
- extinction Extinction cross-section
- axis_rotation 3D rotation matrix parametrized by axis + angle
- euler 3D rotation matrix parametrized by Euler angles
- interaction_matrix Build the coupled-dipole interaction matrix
- block_diagonal Diagonal blocks of the coupled-dipole interaction matrix
- incident_field Construct the incident fields for specific Euler angles
- multiple_incident_field Construct the incident fields for specific axes+angles

Examples

```
show( cda )
```

`circular_dichroism_spectrum` *circular_dichroism_spectrum*

Description

Simulate a CD spectrum

Usage

```
circular_dichroism_spectrum(cluster, material, medium = 1.33, Nquad = 100,
  averaging = c("GL", "QMC", "grid", "cheap"), iterative = FALSE,
  precision = 0.001, Nmax = 10000, dN = Nquad, full = TRUE,
  progress = FALSE, verbose = TRUE, result.matrix = FALSE)
```

Arguments

`cluster`	cluster (list)
`material`	material
`medium`	refractive index medium
`Nquad`	number of integration points
`averaging`	averaging method, using either Gauss Legendre quadrature (default), Quasi Monte Carlo, regular grid, or "cheap" (3 axes)
`iterative`	logical, increase N until convergence (QMC only)
`precision`	relative diff between two runs (QMC only)
`Nmax`	maximum N if convergence not attained (QMC only)
`dN`	iterative increase in N (QMC only)
`full`	logical use full (retarded) dipolar field
`progress`	print progress lines
`verbose`	display messages
`result.matrix`	logical return the results as a matrix

Details

CD spectrum

Author(s)

baptiste Auguie

References

Y. Okada, Efficient numerical orientation averaging of light scattering properties with a quasi-Monte-Carlo method, Journal of Quantitative Spectroscopy and Radiative Transfer, Volume 109, Issue 9, June 2008, Pages 1719-1742.

`cluster_chain` *cluster_chain*

Description

cluster_chain

Usage

```
cluster_chain(N, pitch = 500, a = 50, b = 30, c = b)
```

Arguments

`N`	number of rods
`pitch`	pitch
`a`	semi axis
`b`	semi axis
`c`	semi axis

Details

linear chain of parallel nanorods

Value

list with r, sizes, angles

Author(s)

baptiste Auguie

See Also

Other user_level cluster: `cluster_dimer`, `cluster_dimer_end`, `cluster_helix`, `equal_angles`, `equal_sizes`, `helix`

cluster_dimer *cluster_dimer*

Description

cluster_dimer

Usage

```
cluster_dimer(d = a, dihedral = 0, alpha1 = 0, alpha2 = 0, a = 0.035,
  b = 0.012)
```

Arguments

d	center-to-center distance
dihedral	dihedral angle
alpha1	angle first rod
alpha2	angle second rod
a	semi axis
b	semi axis

Details

cluster with two nanorods first rod along x at (0, 0, -d/2) second rod at (0, 0, d/2)

Value

list with r, sizes, angles

Author(s)

baptiste Auguie

See Also

Other user_level cluster: cluster_chain, cluster_dimer_end, cluster_helix, equal_angles, equal_sizes, helix

`cluster_dimer_end` *cluster_dimer_end*

Description

cluster_dimer_end

Usage

```
cluster_dimer_end(d = a, dihedral = 0, a = 0.035, b = 0.012,
  rescale = TRUE)
```

Arguments

`d`	end-to-end distance
`dihedral`	dihedral angle
`a`	semi axis
`b`	semi axis
`rescale`	logical, rescale the z coordinates so that d is the center-to-center distance

Details

cluster with two nanorods first rod along x at (0, 0, -d/2) second rod at (0, 0, d/2)

Value

list with r, sizes, angles

Author(s)

baptiste Auguie

See Also

Other user_level cluster: `cluster_chain`, `cluster_dimer`, `cluster_helix`, `equal_angles`, `equal_sizes`, `helix`

cluster_helix *cluster_helix*

Description

cluster_helix

Usage

```
cluster_helix(N = 5, R0 = 12, pitch = 15, delta = pi/2, delta0 = 0,
  right = TRUE, a = 5, b = a, c = a, angles = c("helix", "random",
  "fixed"), seed = 123, ...)
```

Arguments

N	number of particles
R0	radius of helix
pitch	pitch of helix
delta	angle between particles
delta0	initial angle
right	logical, helicity
a	ellipsoid semi-axis
b	ellipsoid semi-axis
c	ellipsoid semi-axis
angles	type of angular orientation
seed	random seed for reproducibility
...	extra arguments (ignored)

Details

helical cluster of ellipsoids

Value

list

Author(s)

baptiste Auguie

See Also

Other user_level cluster: cluster_chain, cluster_dimer, cluster_dimer_end, equal_angles, equal_sizes, helix

`curve_povray` *curve_povray*

Description

create long string of small particles for povray input

Usage

```
curve_povray(positions, size = 0.005, radius, out = "positions.pov")
```

Arguments

`positions`	matrix of positions
`size`	radius of particles
`radius`	radius of inner cylinder
`out`	output filename

Details

writes a list of particles, as well as cylinder

Value

side-effect only (note append=TRUE)

Author(s)

baptiste Auguie

See Also

Other user_level povray: `particles_povray`

`dispersion` *Rcpp module: dispersion*

Description

Exposes C++ calculation of scattering and absorption of dipolar particles by linearly polarised light in fixed orientation.

Details

- dispersion Returns absorption, scattering and extinction cross-sections for two orthogonal polarisations at multiple angles of incidence, fixed wavelength (subroutine)
- dispersion_spectrum Returns absorption, scattering and extinction spectra for two orthogonal polarisations at multiple angles of incidence

Examples

```
show( dispersion )
```

`dispersion_spectrum` *dispersion_spectrum*

Description

dispersion spectrum

Usage

```
dispersion_spectrum(cluster, material, medium = 1.33, angles = 0,
  axes = "z", polarisation = c("linear", "circular"), progress = FALSE)
```

Arguments

`cluster`	list describing a cluster
`material`	list
`medium`	medium refractive index
`angles`	of incident field in radians
`axes`	of incident field rotation character vector from ('x', 'y', 'z')
`polarisation`	linear or circular polarisation
`progress`	logical, display progress bar

Details

dispersion spectrum

Value

data.frame

Author(s)

baptiste Auguie

equal_angles *equal_angles*

Description

equal_angles

Usage

```
equal_angles(phi = 0, theta = 0, psi = 0, N)
```

Arguments

phi	phi
theta	theta
psi	psi
N	N

Details

generate a matrix of equal angles

Value

matrix Nx3

Author(s)

baptiste Auguie

See Also

Other user_level cluster: cluster_chain, cluster_dimer, cluster_dimer_end, cluster_helix, equal_sizes, helix

equal_sizes *equal_sizes*

Description

equal_sizes

Usage

```
equal_sizes(a, b, c, N)
```

Arguments

a	a
b	b
c	c
N	N

Details

generate a matrix of equal particle sizes

Value

matrix Nx3

Author(s)

baptiste Auguie

See Also

Other user_level cluster: cluster_chain, cluster_dimer, cluster_dimer_end, cluster_helix, equal_angles, helix

G0 *Lattice sum*

Description

Converged lattice sum G0 for an infinite 2D array of dipoles at normal incidence

Format

`wavelength` A numeric vector

`Qx` A numeric vector

`Gxx` A complex vector

Details

The calculation was made using code from Prof. J. G. de Abajo (CSIC, Spain)

Examples

```
data(G0)
## Not run: demo(lattice_sum)
```

helix *helix*

Description

helix curve

Usage

```
helix(R0 = 500, pitch = 600, N = 5, delta = pi/8, delta0 = pi/2,
  n.smooth = 100 * N, right = TRUE)
```

Arguments

`R0`	radius in nm
`pitch`	pitch in nm
`N`	number of particles
`delta`	twist angle between two particles
`delta0`	angle shift at origin
`n.smooth`	number of interpolation points (for plotting purposes)
`right`	logical, handedness

Details

add particles on an helix

Value

list with r, sizes, invalpha, angles, R0 and smooth interp. points

Author(s)

baptiste Auguie

See Also

Other user_level cluster: cluster_chain, cluster_dimer, cluster_dimer_end, cluster_helix, equal_angles, equal_sizes

Examples

```
cl <- helix(500, 1000, 36, delta=pi/6, n.smooth=1e3) ; str(cl)
## Not run:
require(rgl)
open3d()
spheres3d(cl$smooth, radius=1,col=2)
## ellipsoids are oriented following the helix
sizes <- equal_sizes(40, 20, 20,NROW(cl$positions))
rgl.ellipsoids(cl$positions, sizes, cl$angles, col="gold")

## End(Not run)
```

---

inverse_polarizability

inverse_polarizability

Description

inverse polarizability tensors

Usage

```
inverse_polarizability(cluster, material,
  polarizability_fun = polarizability_ellipsoid, ...)
```

Arguments

cluster	cluster
material	material
polarizability_fun	polarizability function
...	additional arguments passed to polarizability_fun

Details

calculates and formats the principal polarizability of several particles

Value

matrix with each row being the 3 principal values of each polarizability tensor

Author(s)

Baptiste Auguie

See Also

Other user_level polarizability: `alpha_kuwata`, `La`, `polarizability_ellipsoid`

`La` *La*

Description

Shape factor for an ellipsoid

Usage

```
La(a = 50, b = a, c = a)
```

Arguments

`a`	semi-axis in nm
`b`	semi-axis in nm
`c`	semi-axis in nm

Details

calculates the shape factor for a general ellipsoid

Value

shape factor along a

Author(s)

baptiste Auguie

See Also

Other user_level polarizability: `alpha_kuwata`, `inverse_polarizability`, `polarizability_ellipsoid`

particles_povray *particles_povray*

Description

create particles for povray input

Usage

```
particles_povray(positions, angles, sizes, out = "positions.pov")
```

Arguments

positions	matrix of positions
angles	matrix of Euler angles in radians
sizes	matrix of particle sizes
out	output filename

Details

writes a list of particles

Value

side-effect only

Author(s)

baptiste Auguie

See Also

Other user_level povray: curve_povray

polarizability_ellipsoid *polarizability_ellipsoid*

Description

principal polarizability components for an ellipsoidal particle

Usage

```
polarizability_ellipsoid(wavelength, epsilon, a = 50, b = 30, c = b,
  medium = 1.33, kuwata = TRUE)
```

Arguments

wavelength	wavelength in nm
epsilon	complex permittivity
a	semi-axis in nm
b	semi-axis in nm
c	semi-axis in nm
medium	surrounding medium
kuwata	logical, use Kuwata or Clausius Mossotti prescription, see Details

Details

uses the Kuwata prescription (see references)

The Kuwata version includes semi-empirical terms of radiative correction and dynamic depolarisation to better match the fully retarded dipolar response in a reasonable range of (subwavelength) sizes and aspect ratios.

Value

matrix of polarizability

Author(s)

baptiste Auguie

References

Kuwata et al. Resonant light scattering from metal nanoparticles: Practical analysis beyond Rayleigh approximation Appl. Phys. Lett. 83, 22 (2003)

See Also

Other user_level polarizability: alpha_kuwata, inverse_polarizability, La

---

rgl.ellipsoid *rgl.ellipsoid*

Description

creates an rgl ellipsoid

Usage

```
rgl.ellipsoid(x = 0, y = 0, z = 0, a = 1, b = 1, c = 1, phi = 0,
  theta = 0, psi = 0, subdivide = 3, smooth = TRUE, ...)
```

Arguments

x	x
y	y
z	z
a	axis
b	axis
c	axis
phi	phi
theta	theta
psi	psi
subdivide	subdivision
smooth	smoothing
...	additional params

Details

deforms, rotate, and translate a sphere

Value

an rgl mesh

Author(s)

baptiste Auguie

See Also

Other user_level rgl: rgl_annotate, rgl.ellipsoids

Examples

```
## Not run:  require(rgl) ;  ee <- rgl.ellipsoid()
shapelist3d(ee)
## End(Not run)
```

rgl.ellipsoids *rgl.ellipsoids*

Description

Create a list of rgl ellipsoids oriented in space

Usage

```
rgl.ellipsoids(positions, sizes, angles, ...)
```

Arguments

positions	matrix of positions
sizes	matrix of axis lengths
angles	matrix of Euler angles
...	additional params

Details

each ellipsoid is specified by its position, dimensions, and Euler angles

Value

rgl mesh

Author(s)

baptiste Auguie

See Also

Other user_level rgl: rgl_annotate, rgl.ellipsoid

Examples

```
cl <- helix(0.5, 1, 36, delta=pi/6, n.smooth=1e3)
sizes <- equal_sizes(0.04,0.02,0.02,NROW(cl$positions))
## Not run:  require(rgl) ; rgl.ellipsoids(cl$positions, sizes, cl$angles, col="gold")
```

rgl_annotate *rgl_annotate*

Description

Add axes to a rgl scene

Usage

```
rgl_annotate()
```

Details

x, y, z axes

Value

draw axes

Author(s)

baptiste Auguie

See Also

Other user_level rgl: rgl.ellipsoid, rgl.ellipsoids

Index