# Package ‘cosmoFns’

May 9, 2022

**Type** Package

**Title** Functions for Cosmological Distances, Times, Luminosities, Etc

**Version** 1.1-1

**Date** 2022-05-08

**Author** Andrew Harris

**Maintainer** Andrew Harris <harris@astro.umd.edu>

**Description** Package encapsulates standard expressions for distances, times, luminosities, and other quantities useful in observational cosmology, including molecular line observations. Currently coded for a flat universe only.

**License** GPL (>= 2)

**LazyLoad** yes

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-05-09 07:20:05 UTC

## R topics documented:

`cosmoFns-package` *Cosmology functions*

## Description

Package contains functions for computation of distances and luminosities in a flat cosmology.

## Details

| | |
|---|---|
| Package: | cosmoFns |
| Type: | Package |
| Version: | 1.1-1 |
| Date: | 2022-05-08 |
| License: | GPL |
| LazyLoad: | yes |

## Author(s)

A. Harris

Maintainer: <harris@astro.umd.edu>

## References

"Distance Measures in Cosmology," D.W. Hogg (2000), arXiv:astro-ph/9905116; "Warm Molecular Gas in the Pirmeval Galaxy 10214+4724", P.M. Solomon, D. Downes, and S.J.E. Radford (1992), Ap.J. 398, L29; "First-year WMAP observations...", Spergel et al., ApJS 148:175 (2003). "Submillimetre and far-infrared spectral energy distributions of galaxies...", A.W. Blain, V.E. Barnard \& S.C. Chapman 2003, MNRAS 338, 733.

## Examples

```
D.L(z=2.3)
```

`D.A` *Angular diameter distance*

## Description

Function computes angular diameter distance

## Usage

```
D.A(z, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

## Arguments

| | |
|---|---|
| z | Redshift |
| omega.m | Omega matter parameter |
| omega.lambda | Omega lambda parameter |
| H.0 | Hubble constant in km/s/Mpc |

## Value

Angular distance in Mpc

## Note

For flat universe, omega.k = 0.

## Author(s)

A. Harris

## References

Hogg (2000), arXiv:astro-ph/9905116, equation (18)

## Examples

```
D.A(2.3)

z <- seq(0.1, 5, 0.1)
d <- D.A(z)
plot(z, d/max(d), t='l', xlab='z', ylab='Normalized D.A')
```

---

# D.L *Luminosity distance*

---

## Description

Function computes luminosity distance in a flat cosmology.

## Usage

```
D.L(z, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

### Arguments

| | |
|---|---|
| `z` | Redshift |
| `omega.m` | Omega matter parameter |
| `omega.lambda` | Omega lambda parameter |
| `H.0` | Hubble constant in km/s/Mpc |

### Value

Luminosity distance in Mpc

### Author(s)

A. Harris

### References

Hogg (2000), arXiv:astro-ph/9905116, equation (21)

### Examples

```
D.L(2.3)
```

---

## D.M *Comoving distance*

---

### Description

Function computes comoving distance in a flat cosmology.

### Usage

```
D.M(z, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

### Arguments

| | |
|---|---|
| `z` | Redshift |
| `omega.m` | Omega matter parameter |
| `omega.lambda` | Omega lambda parameter |
| `H.0` | Hubble constant in km/s/Mpc |

### Value

Comoving distance in Mpc

### Note

For flat universe, `omega.k = 0`, so transverse and line-of-sight comoving distances are equal.

### Author(s)

A. Harris

### References

Hogg (2000), arXiv:astro-ph/9905116, equations (16) and (15)

### Examples

```
D.M(2.3)
```

---

`dComovVol` *Differential comoving volume*

---

### Description

Function computes differential comoving volume in a flat cosmology.

### Usage

```
dComovVol(z, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

### Arguments

| | |
|---|---|
| `z` | Redshift |
| `omega.m` | Omega matter parameter |
| `omega.lambda` | Omega lambda parameter |
| `H.0` | Hubble constant in km/s/Mpc |

### Value

Differential comoving volume in Mpc^3

### Author(s)

A. Harris

### References

Hogg (2000), arXiv:astro-ph/9905116, equation (28)

### Examples

```
dComovVol(2.3)
```

`dimmingFactor` *Flux dimming factor*

## Description

Function computes flux dimming factor in a flat cosmology.

## Usage

```
dimmingFactor(z, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

## Arguments

| | |
|---|---|
| `z` | Redshift |
| `omega.m` | Omega matter parameter |
| `omega.lambda` | Omega lambda parameter |
| `H.0` | Hubble constant in km/s/Mpc |

## Value

Flux dimming factor, unnormalied. Mathematically, it is `(1+z)/D.L^2`. This is the factor that scales luminosity densitiy in the observed frame to flux density in the observed frame.

## Author(s)

A. Harris

## References

Hogg (2000), arXiv:astro-ph/9905116: section 7, part of equation (22)

## See Also

`D.L`

## Examples

```
z <- seq(0.1, 5, 0.1)
df <- dimmingFactor(z)
plot(z, df/max(df), t='l', xlab='z', ylab='Normalized dimming factor')
```

`lineLum` *Line luminosity*

## Description

Compute rest-frame line luminosity.

## Usage

```
lineLum(intInt, z, f.rest = 115.27, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

## Arguments

| | |
|---|---|
| `intInt` | Integrated intensity in Jy km/s |
| `z` | Redshift |
| `f.rest` | Line rest frequency in GHz |
| `omega.m` | Omega matter parameter |
| `omega.lambda` | Omega lambda parameter |
| `H.0` | Hubble constant in km/s/Mpc |

## Value

Rest-frame line luminosity in solar luminosities.

## Note

For flat universe, `omega.k = 0`.

## Author(s)

A. Harris

## References

Solomon, Downes & Radford (1992), ApJ 398, L29, equation (1)

## See Also

`Lprime`

## Examples

```
snu <- 1.e-3 # 1 mJy peak
wid <- 400   # 400 km/s wide
intInt <- 1.06*snu*wid  # Gaussian line
z <- 2.3
lineLum(intInt, z)
```

`lookbackTime` *Cosmic lookback time*

## Description

Compute cosmic lookback time given z and cosmological parameters

## Usage

```
lookbackTime(z, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

## Arguments

| | |
|---|---|
| z | Redshift |
| omega.m | Omega matter parameter |
| omega.lambda | Omega lambda parameter |
| H.0 | Hubble constant in km/s/Mpc |

## Details

Defaults for `omega.m`, `omega.lambda`, and `omega.m`, are from WMAP cosmology; `omega.k` (curvature term) is computed from relationship between `omegas` in flat cosmology (`omega.k = 0`).

## Value

Lookback time in Gyr.

## Author(s)

A. Harris

## References

"Principles of Physical Cosmology," P.J. Peebles, Princeton c. 1993, (5.63); "Distance Measures in Cosmology," Hogg (2000), arXiv:astro-ph/9905116, equation (30); "First-year WMAP observations...", Spergel et al., ApJS 148:175 (2003)

## Examples

```
# lookback time for z = 2
lookbackTime(2)
# Inverse problem, age of Earth (4.6 Gyr) example:
uniroot(function(x) lookbackTime(x) - 4.6, c(0,2))$root
```

`Lprime` *Line luminosity, L'*

## Description

Compute L' line luminosity

## Usage

```
Lprime(intInt, z, f.rest = 115.27, omega.m = 0.27, omega.lambda = 0.73, H.0 = 71)
```

## Arguments

| | |
|---|---|
| `intInt` | Integrated intensity in Jy km/s |
| `z` | Redshift |
| `f.rest` | Line rest frequency in GHz |
| `omega.m` | Omega matter parameter |
| `omega.lambda` | Omega lambda parameter |
| `H.0` | Hubble constant in km/s/Mpc |

## Value

Rest-frame line luminosity in `K km/s pc^-2`.

## Note

For flat universe, `omega.k = 0`. Useful for empirical mass estimates. L' is proportional to the brightness temperature of the transition.

## Author(s)

A. Harris

## References

Solomon, Downes & Radford (1992), ApJ 398, L29, equation (3)

## See Also

`lineLum`, `mass.CO`

## Examples

```
snu <- 1.e-3 # 1 mJy peak
wid <- 400   # 400 km/s wide
intInt <- 1.06*snu*wid  # Gaussian line
z <- 2.3
Lprime(intInt, z)
```

mass.CO *Molecular mass*

## Description

Compute molecular mass (default CO J = 1-0) from L' and empirical conversion factor.

## Usage

```
mass.CO(intInt, z, alpha = 0.8, f.rest = 115.27, omega.m = 0.27,
omega.lambda = 0.73, H.0 = 71)
```

## Arguments

| | |
|---|---|
| `intInt` | Integrated intensity in Jy km/s |
| `z` | Redshift |
| `alpha` | Empirical mass conversion factor, see details |
| `f.rest` | Line rest frequency in GHz |
| `omega.m` | Omega matter parameter |
| `omega.lambda` | Omega lambda parameter |
| `H.0` | Hubble constant in km/s/Mpc |

## Details

`alpha` is an empirical mass conversion factor. The exact value is a topic of considerable debate. For CO, see Solomon and Vanden Bout (2005), also Tacconi et al. (2008) for reviews.

## Value

Gas mass in solar masses.

## Author(s)

A. Harris

## References

Solomon, Downes & Radford (1992), ApJ 398, L29, equations (3) and (4); Solomon & Vanden Bout (2005) ARA&A 43, 677; Tacconi et al. (2008) ApJ 680, 246.

## See Also

`Lprime`

## Examples

```
snu <- 1.e-3 # 1 mJy peak
wid <- 400   # 400 km/s wide
intInt <- 1.06*snu*wid  # Gaussian line
z <- 2.3
mass.CO(intInt, z)
```

---

sedFitThin *Optically-thin SED fit*

---

## Description

Function takes Herschel-SPIRE photometry and fits optically-thin greybody function for a single-component temperature and galaxy luminosity. Function generates nsamp realizations of observed flux densities with standard deviations for error analysis.

## Usage

```
sedFitThin(s, e = s*0.2, z = 2.5, nsamp = 100, alpha = 2, beta = 1.5,
wl= c(250, 350, 500), sc.start = 1.e-6, T.start = 50)
```

## Arguments

| | |
|---|---|
| `s` | Vector of observed-frame flux densities [Jy] |
| `e` | Vector of standard deviation of observed-frame flux density [Jy] |
| `z` | Galaxy redshift |
| `nsamp` | Number of realizations for Monte-Carlo calculation |
| `alpha` | Index of power-law for short-wavelength extension |
| `beta` | Dust emissivity power law |
| `wl` | Vector of observed-frame wavelengths corresponding to `s` and `e` [microns] |
| `sc.start` | Initial guess for fit luminosity density scaling factor |
| `T.start` | Initial guess for dust temperature [K] |

## Details

Conversion from observed to rest frame is from equation (24) in Hogg 2000. Dust temperature and 8-1000 micron luminosity derivation is described in Blain, Barnard & Chapman 2003. Galaxy SEDs typically fall off more slowly than greybody on the Wien side; see plot generated by examples below to visualize power-law extension suggested by Blain et al. 2003.

## Value

List of class `sedfit` with elements:

| | |
|---|---|
| `td` | Mean of dust temperature distribution |
| `e.td` | Standard deviation of dust temperature distribution |
| `lum.gb` | Mean of greybody luminosity distribution |
| `e.lum.gb` | Standard deviation of greybody luminosity distribution |
| `lum.gbpl` | Mean of greybody-power law luminosity distribution |
| `e.lum.gbpl` | Standard deviation of greybody-power law luminosity distribution |
| `scaleFactor` | Conversion between observed frame flux density and rest frame luminosity density |
| `success` | Fraction of fit attempts that converged |
| `results` | Matrix with `nsamp` rows and 5 columns: dust temperature in K, greybody luminosity, luminosity for greybody with smoothly-joined power law to short wavelengths, luminosity density scaling, and transition frequency in GHz for power law extension. The first row contains results for the center-of-error input flux densities `s`. |

## Note

Fit will sometimes crash on numerical derivative and throw an error. In this case the routine will halt without producing results. The more usual lack of convergence is reported as a warning, and the corresponding results will be `NA` in the output matrix.

## Author(s)

A. Harris

## References

Hogg 2000, astro-ph 9905116v4; Blain, Barnard & Chapman 2003, MNRAS 338, 733.

## Examples

```
s <- c(0.242, 0.293, 0.231)
e <- c(0.037, 0.045, 0.036)
z <- 2.41
beta <- 1.5
alpha <- 2
X <- sedFitThin(s=s, e=e, z=z, alpha=alpha, beta=beta, nsamp=100)
str(X)

## Make a plot
# greybody in blue, power-law extension in red dashed line
# functions
# optically thin greybody
otGreybody <- function(nu, T, beta, sc=1) {
                       # nu in GHz, T in K, beta and sc unitless
```

```
                        sc*nu^(3+beta)/(exp(0.04801449*nu/T) - 1)
               }
# high frequency tail
hfTail <- function(nu, alpha) nu^-alpha
#
# setups for 8-1000 microns:
nu.low <- 3e5/1000
nu.high <- 3e5/8
l.nue <- s*X$scaleFactor
#
# greybody
nue.sweep <- seq(nu.low, nu.high, len=350)
pred <- otGreybody(nue.sweep, X$results[1,1], beta=beta,
                   X$results[1,4])
ylim <- range(pred, l.nue)
par(fig=c(0,1,0.2,1), mgp=c(1.8, 0.6, 0))
plot(3e5/nue.sweep, pred, t='l', ylim=ylim, log='xy', col=4,
     xlab='Rest frame wavelength [microns]',
     ylab=expression(paste('Luminosity density [ ', L[sun],
                           ' ', Hz^-1, ']')))
# power law
nue.sweep <- seq(X$results[1,5], nu.high, len=100)
val.t <- otGreybody(nu=X$results[1,5], T=X$results[1,1], beta=beta,
     sc=X$results[1,4])
lines(3e5/nue.sweep, val.t*hfTail(nue.sweep/X$results[1,5], alpha=alpha),
      col=2, lwd=1, lty=2)
# data
wl <- c(250, 350, 500)
nue <- 3e5/wl*(1+z)
points(3e5/nue, l.nue, pch=16, col=3)
```

# Index