# Package 'EMD'

May 7, 2026

**Version** 1.5.9

**Date** 2021-12-30

**Title** Empirical Mode Decomposition and Hilbert Spectral Analysis

**Author** Donghoh Kim [aut, cre],
Hee-Seok Oh [aut]

**Maintainer** Donghoh Kim <donghoh.kim@gmail.com>

**Depends** R (>= 3.0), fields (>= 6.9.1), locfit (>= 1.5-8)

**Description** For multiscale analysis, this package carries out empirical mode decomposition and Hilbert spectral
analysis. For usage of EMD, see Kim and Oh, 2009 (Kim, D and Oh, H.-S. (2009) EMD: A Package for Empirical
Mode Decomposition and Hilbert Spectrum, The R Journal, 1, 40-46).

**License** GPL (>= 3)

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2022-01-04 00:10:18 UTC

## Contents

---

`cvimpute.by.mean` *Imputation by the mean of the two adjacent values*

---

## Description

This function performs imputation by the mean of the two adjacent values for test dataset of cross-validation.

## Usage

```
cvimpute.by.mean(y, impute.index)
```

## Arguments

| | |
|---|---|
| `y` | observation |
| `impute.index` | test dataset index for cross-validation |

## Details

This function performs imputation by the mean of the two adjacent values for test dataset of cross-validation. See Kim et al. (2012) for detalis.

## Value

| | |
|---|---|
| `yimpute` | imputed values by the mean of the two adjacent values |

## References

Kim, D., Kim, K.-O. and Oh, H.-S. (2012) Extending the Scope of Empirical Mode Decomposition using Smoothing. *EURASIP Journal on Advances in Signal Processing*, **2012:168**, doi: 10.1186/1687-6180-2012-168.

## See Also

`cvtype`, `semd`.

cvtype *Generating test dataset index for cross-validation*

## Description

This function generates test dataset index for cross-validation.

## Usage

```
cvtype(n, cv.bsize=1, cv.kfold, cv.random=FALSE)
```

## Arguments

| | |
|---|---|
| n | the number of observation |
| cv.bsize | block size of cross-validation |
| cv.kfold | the number of fold of cross-validation |
| cv.random | whether or not random cross-validation scheme should be used. Set cv.random=TRUE for random cross-validation scheme |

## Details

This function provides index of test dataset according to various cross-validation scheme. One may construct K test datasets in a way that each testset consists of blocks of b consecutive data. Set cv.bsize = b for this. To select each fold at random, set cv.random = TRUE. See Kim et al. (2012) for detalis.

## Value

matrix of which row is test dataset index for cross-validation

## References

Kim, D., Kim, K.-O. and Oh, H.-S. (2012) Extending the Scope of Empirical Mode Decomposition using Smoothing. *EURASIP Journal on Advances in Signal Processing*, **2012:168**, doi: 10.1186/1687-6180-2012-168.

## Examples

```
# Traditional 4-fold cross-validation for 100 observations
cvtype(n=100, cv.bsize=1, cv.kfold=4, cv.random=FALSE)
# Random 4-fold cross-validation with block size 2 for 100 observations
cvtype(n=100, cv.bsize=2, cv.kfold=4, cv.random=TRUE)
```

emd *Empirical Mode Decomposition*

## Description

This function performs empirical mode decomposition.

## Usage

```
emd(xt, tt=NULL, tol=sd(xt)*0.1^2, max.sift=20, stoprule="type1",
    boundary="periodic", sm="none", smlevels=c(1), spar=NULL, alpha=NULL,
    check=FALSE, max.imf=10, plot.imf=FALSE, interm=NULL, weight=NULL)
```

## Arguments

| | |
|---|---|
| xt | observation or signal observed at time tt |
| tt | observation index or time index |
| tol | tolerance for stopping rule of sifting. If stoprule=type5, the number of iteration for S stoppage criterion. |
| max.sift | the maximum number of sifting |
| stoprule | stopping rule of sifting. The type1 stopping rule indicates that absolute values of envelope mean must be less than the user-specified tolerance level in the sense that the local average of upper and lower envelope is zero. The stopping rules type2, type3, type4 and type5 are the stopping rules given by equation (5.5) of Huang et al. (1998), equation (11a), equation (11b) and S stoppage of Huang and Wu (2008), respectively. |
| boundary | specifies boundary condition from “none", “wave", “symmetric", “periodic" or “evenodd". See Zeng and He (2004) for evenodd boundary condition. |
| sm | specifies whether envelop is constructed by interpolation, spline smoothing, kernel smoothing, or local polynomial smoothing. Use “none" for interpolation, “spline" for spline smoothing, “kernel" for kernel smoothing, or “locfit" for local polynomial smoothing. See Kim et al. (2012) for detalis. |
| smlevels | specifies which level of the IMF is obtained by smoothing other than interpolation. |
| spar | specifies user-supplied smoothing parameter of spline smoothing, kernel smoothing, or local polynomial smoothing. |
| alpha | deprecated. |
| check | specifies whether the sifting process is displayed. If check=TRUE, click the plotting area to start the next step. |
| max.imf | the maximum number of IMF's |
| plot.imf | specifies whether each IMF is displayed. If plot.imf=TRUE, click the plotting area to start the next step. |
| interm | specifies vector of periods to be excluded from the IMF's to cope with mode mixing. |
| weight | deprecated. |

## Details

This function performs empirical mode decomposition.

## Value

| | |
|---|---|
| `imf` | IMF's |
| `residue` | residue signal after extracting IMF's from observations `xt` |
| `nimf` | the number of IMF's |

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

Huang, N. E. and Wu, Z. (2008) A review on Hilbert-Huang Transform: Method and its applications to geophysical studies. *Reviews of Geophysics*, **46**, RG2006.

Kim, D., Kim, K.-O. and Oh, H.-S. (2012) Extending the Scope of Empirical Mode Decomposition using Smoothing. *EURASIP Journal on Advances in Signal Processing*, **2012:168**, doi: 10.1186/1687-6180-2012-168.

Zeng, K and He, M.-X. (2004) A simple boundary process technique for empirical mode decomposition. *Proceedings of 2004 IEEE International Geoscience and Remote Sensing Symposium*, **6**, 4258–4261.

## See Also

`extrema`, `extractimf`.

## Examples

```
### Empirical Mode Decomposition
ndata <- 3000
tt2 <- seq(0, 9, length=ndata)
xt2 <- sin(pi * tt2) + sin(2* pi * tt2) + sin(6 * pi * tt2)  + 0.5 * tt2

try <- emd(xt2, tt2, boundary="wave")

### Ploting the IMF's
par(mfrow=c(try$nimf+1, 1), mar=c(2,1,2,1))
rangeimf <- range(try$imf)
for(i in 1:try$nimf) {
    plot(tt2, try$imf[,i], type="l", xlab="", ylab="", ylim=rangeimf,
    main=paste(i, "-th IMF", sep="")); abline(h=0)
}
plot(tt2, try$residue, xlab="", ylab="", main="residue", type="l", axes=FALSE); box()
```

`emd.pred` *Prediction by EMD and VAR model*

## Description

This function calculates prediction values and confidence limits using EMD and VAR (vector autoregressive) model.

## Usage

```
emd.pred(varpred, trendpred, ci = 0.95, figure = TRUE)
```

## Arguments

| | |
|---|---|
| `varpred` | prediction result of IMF's by VAR model. |
| `trendpred` | prediction result of residue by polynomial regression model. |
| `ci` | confidence interval level. |
| `figure` | specifies whether prediction result is displayed. |

## Details

This function calculates prediction values and confidence limits using EMD and VAR (vector autoregressive) model. See Kim et al. (2008) for detalis.

## Value

| | |
|---|---|
| `fcst` | prediction values |
| `lower` | lower limits of prediction |
| `upper` | upper limits of prediction |

## References

Kim, D, Paek, S.-H. and Oh, H.-S. (2008) A Hilbert-Huang Transform Approach for Predicting Cyber-Attacks. *Journal of the Korean Statistical Society*, **37**, 277–283, doi:10.1016/j.jkss.2008.02.006.

emd2d *Bidimenasional Empirical Mode Decomposition*

## Description

This function performs the bidimenasional empirical mode decomposition utilizing extrema detection based on the equivalence relation between neighboring pixels.

## Usage

```
emd2d(z, x = NULL, y = NULL, tol = sd(c(z)) * 0.1^2, max.sift = 20,
    boundary = "reflexive", boundperc = 0.3, max.imf = 5, sm = "none",
    smlevels = 1, spar = NULL, weight = NULL, plot.imf = FALSE)
```

## Arguments

| | |
|---|---|
| z | matrix of an image observed at (x, y) |
| x, y | locations of regular grid at which the values in z are measured |
| tol | tolerance for stopping rule of sifting |
| max.sift | the maximum number of sifting |
| boundary | specifies boundary condition from “none", “symmetric" or “reflexive". |
| boundperc | expand an image by adding specified percentage of image at the boundary when boundary condition is 'symmetric' or 'reflexive'. |
| max.imf | the maximum number of IMF's |
| sm | specifies whether envelop is constructed by interpolation, thin-plate smoothing, Kriging, local polynomial smoothing, or loess. Use “none" for interpolation, “Tps" for thin-plate smoothing, “mKrig" for Kriging, “locfit" for local polynomial smoothing, or “loess" for loess. See Kim et al. (2012) for detalis. |
| smlevels | specifies which level of the IMF is obtained by smoothing other than interpolation. |
| spar | specifies user-supplied smoothing parameter of thin-plate smoothing, Kriging, local polynomial smoothing, or loess. |
| weight | deprecated. |
| plot.imf | specifies whether each IMF is displayed. If plot.imf=TRUE, click the plotting area to start the next step. |

## Details

This function performs the bidimenasional empirical mode decomposition utilizing extrema detection based on the equivalence relation between neighboring pixels. See Kim et al. (2012) for detalis.

## Value

| | |
|---|---|
| imf | two dimensional IMF's |
| residue | residue image after extracting the IMF's |
| maxindex | index of maxima |
| minindex | index of minima |
| nimf | number of IMF's |

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

Kim, D., Park, M. and Oh, H.-S. (2012) Bidimensional Statistical Empirical Mode Decomposition. *IEEE Signal Processing Letters*, **19**, 191–194, doi: 10.1109/LSP.2012.2186566.

## See Also

extrema2dC, extractimf2d.

## Examples

```
data(lena)
z <- lena[seq(1, 512, by=4), seq(1, 512, by=4)]
image(z, main="Lena", xlab="", ylab="", col=gray(0:100/100), axes=FALSE)

## Not run:
lenadecom <- emd2d(z, max.imf = 4)
imageEMD(z=z, emdz=lenadecom, extrema=TRUE, col=gray(0:100/100))
## End(Not run)

### Test Image
ndata <- 128

x <- y <- seq(0, 9, length=ndata)
meanf1 <- outer(sin(2 * pi * x), sin(2 * pi * y))
meanf2 <- outer(sin(0.5 * pi * x), sin(0.5 * pi * y))
meanf <- meanf1 + meanf2

snr <- 2
set.seed(77)
zn <- meanf + matrix(rnorm(ndata^2, 0, sd(c(meanf))/snr), ncol=ndata)

rangezn <- range(c(meanf1, meanf2, meanf, zn))
par(mfrow=c(2,2), mar=0.1 + c(0, 0.25, 3, 0.25))
image(meanf1, main="high frequency component", xlab="", ylab="", zlim=rangezn,
    col=gray(100:0/100), axes=FALSE)
image(meanf2, main="low frequency component", xlab="", ylab="", zlim=rangezn,
    col=gray(100:0/100), axes=FALSE)
image(meanf, main="test image", xlab="", ylab="", zlim=rangezn, col=gray(100:0/100), axes=FALSE)
```

```
image(zn, main="noisy image", xlab="", ylab="", zlim=rangezn, col=gray(100:0/100), axes=FALSE)

## Not run:
out <- emd2d(zn, max.imf=3, sm="locfit", smlevels=1, spar=0.004125)
par(mfcol=c(3,1), mar=0.1 + c(0, 0.25, 0.25, 0.25))
image(out$imf[[1]], main="", xlab="", ylab="", col=gray(100:0/100), zlim=rangezn, axes=FALSE)
image(out$imf[[2]], main="", xlab="", ylab="", col=gray(100:0/100), zlim=rangezn, axes=FALSE)
image(out$imf[[3]], main="", xlab="", ylab="", col=gray(100:0/100), zlim=rangezn, axes=FALSE)
## End(Not run)
```

---

`emddenoise` *Denoising by EMD and Thresholding*

---

## Description

This function performs denoising by empirical mode decomposition and thresholding.

## Usage

```
emddenoise(xt, tt = NULL, cv.index, cv.level, cv.tol = 0.1^3,
    cv.maxiter = 20, by.imf = FALSE, emd.tol = sd(xt) * 0.1^2,
    max.sift = 20, stoprule = "type1", boundary = "periodic",
    max.imf = 10)
```

## Arguments

| | |
|---|---|
| `xt` | observation or signal observed at time `tt` |
| `tt` | observation index or time index |
| `cv.index` | test dataset index according to cross-validation scheme |
| `cv.level` | levels to be thresholded |
| `cv.tol` | tolerance for the optimization step of cross-validation |
| `cv.maxiter` | maximum iteration for the optimization step of cross-validation |
| `by.imf` | specifies whether shrinkage is performed by each IMS or not. |
| `emd.tol` | tolerance for stopping rule of sifting. If `stoprule=type5`, the number of iteration for S stoppage criterion. |
| `max.sift` | the maximum number of sifting |
| `stoprule` | stopping rule of sifting. The `type1` stopping rule indicates that absolute values of envelope mean must be less than the user-specified tolerance level in the sense that the local average of upper and lower envelope is zero. The stopping rules `type2`, `type3`, `type4` and `type5` are the stopping rules given by equation (5.5) of Huang et al. (1998), equation (11a), equation (11b) and S stoppage of Huang and Wu (2008), respectively. |
| `boundary` | specifies boundary condition from “none", “wave", “symmetric", “periodic" or “evenodd". See Zeng and He (2004) for `evenodd` boundary condition. |
| `max.imf` | the maximum number of IMF’s |

## Details

This function performs denoising by empirical mode decomposition and cross-validation. See Kim and Oh (2006) for details.

## Value

| | |
|---|---|
| `dxt` | denoised signal |
| `optlambda` | threshold values by cross-validation |
| `lambdaconv` | sequence of lambda's by cross-validation |
| `perr` | sequence of prediction error by cross-validation |
| `demd` | denoised IMF's and residue |
| `niter` | the number of iteration for optimal threshold value |

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

Huang, N. E. and Wu, Z. (2008) A review on Hilbert-Huang Transform: Method and its applications to geophysical studies. *Reviews of Geophysics*, **46**, RG2006.

Kim, D. and Oh, H.-S. (2006) Hierarchical Smoothing Technique by Empirical Mode Decomposition (Korean). *The Korean Journal of Applied Statistics*, **19**, 319–330.

Zeng, K and He, M.-X. (2004) A simple boundary process technique for empirical mode decomposition. *Proceedings of 2004 IEEE International Geoscience and Remote Sensing Symposium*, **6**, 4258–4261.

## See Also

`cvtype`, `emd`.

## Examples

```
ndata <- 1024
tt <- seq(0, 9, length=ndata)
meanf <- (sin(pi*tt) + sin(2*pi*tt) + sin(6*pi*tt)) * (0.0<tt & tt<=3.0) +
 (sin(pi*tt) + sin(6*pi*tt)) * (3.0<tt & tt<=6.0) +
 (sin(pi*tt) + sin(6*pi*tt) + sin(12*pi*tt)) * (6.0<tt & tt<=9.0)
snr <- 3.0
sigma <- c(sd(meanf[tt<=3]) / snr, sd(meanf[tt<=6 & tt>3]) / snr,
sd(meanf[tt>6]) / snr)
set.seed(1)
error <- c(rnorm(sum(tt<=3), 0, sigma[1]),
rnorm(sum(tt<=6 & tt>3), 0, sigma[2]), rnorm(sum(tt>6), 0, sigma[3]))
xt <- meanf + error

cv.index <- cvtype(n=ndata, cv.kfold=2, cv.random=FALSE)$cv.index

## Not run:
```

```
try10 <- emddenoise(xt, cv.index=cv.index, cv.level=2, by.imf=TRUE)

par(mfrow=c(2, 1), mar=c(2, 1, 2, 1))
plot(xt, type="l", main="noisy signal")
lines(meanf, lty=2)
plot(try10$dxt, type="l", main="denoised signal")
lines(meanf, lty=2)
## End(Not run)
```

---

`extractimf` *Intrinsic Mode Function*

---

## Description

This function extracts intrinsic mode function from given a signal.

## Usage

```
extractimf(residue, tt=NULL, tol=sd(residue)*0.1^2, max.sift=20,
    stoprule="type1", boundary="periodic", sm="none", spar=NULL,
    alpha=NULL, check=FALSE, weight=NULL)
```

## Arguments

| | |
|---|---|
| `residue` | observation or signal observed at time `tt` |
| `tt` | observation index or time index |
| `tol` | tolerance for stopping rule of sifting. If `stoprule=type5`, the number of iteration for S stoppage criterion. |
| `max.sift` | the maximum number of sifting |
| `stoprule` | stopping rule of sifting. The `type1` stopping rule indicates that absolute values of envelope mean must be less than the user-specified tolerance level in the sense that the local average of upper and lower envelope is zero. The stopping rules `type2`, `type3`, `type4` and `type5` are the stopping rules given by equation (5.5) of Huang et al. (1998), equation (11a), equation (11b) and S stoppage of Huang and Wu (2008), respectively. |
| `boundary` | specifies boundary condition from “none", “wave", “symmetric", “periodic" or “evenodd". See Zeng and He (2004) for `evenodd` boundary condition. |
| `sm` | specifies whether envelop is constructed by interpolation, spline smoothing, kernel smoothing, or local polynomial smoothing. Use “none" for interpolation, “spline" for spline smoothing, “kernel" for kernel smoothing, or “locfit" for local polynomial smoothing. See Kim et al. (2012) for detalis. |
| `spar` | specifies user-supplied smoothing parameter of spline smoothing, kernel smoothing, or local polynomial smoothing. |
| `alpha` | deprecated. |
| `check` | specifies whether the sifting process is displayed. If `check=TRUE`, click the plotting area to start the next step. |
| `weight` | deprecated. |

## Details

This function extracts intrinsic mode function from given a signal.

## Value

| | |
|---|---|
| `imf` | imf |
| `residue` | residue signal after extracting the finest imf from `residue` |
| `niter` | the number of iteration to obtain the `imf` |

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

Huang, N. E. and Wu, Z. (2008) A review on Hilbert-Huang Transform: Method and its applications to geophysical studies. *Reviews of Geophysics*, **46**, RG2006.

Kim, D., Kim, K.-O. and Oh, H.-S. (2012) Extending the Scope of Empirical Mode Decomposition using Smoothing. *EURASIP Journal on Advances in Signal Processing*, **2012:168**, doi: 10.1186/1687-6180-2012-168.

Zeng, K and He, M.-X. (2004) A simple boundary process technique for empirical mode decomposition. *Proceedings of 2004 IEEE International Geoscience and Remote Sensing Symposium*, **6**, 4258–4261.

## See Also

`extrema`, `emd`.

## Examples

```
### Generating a signal
ndata <- 3000
par(mfrow=c(1,1), mar=c(1,1,1,1))
tt2 <- seq(0, 9, length=ndata)
xt2 <- sin(pi * tt2) + sin(2* pi * tt2) + sin(6 * pi * tt2)  + 0.5 * tt2
plot(tt2, xt2, xlab="", ylab="", type="l", axes=FALSE); box()

### Extracting the first IMF by sifting process
tryimf <- extractimf(xt2, tt2, check=FALSE)
```

---

# extractimf2d *Bidimensional Intrinsic Mode Function*

---

## Description

This function extracts the bidimensional intrinsic mode function from given an image utilizing extrema detection based on the equivalence relation between neighboring pixels.

## Usage

```
extractimf2d(residue, x=NULL, y=NULL, nnrow=nrow(residue),
    nncol=ncol(residue), tol=sd(c(residue))*0.1^2,
    max.sift=20, boundary="reflexive", boundperc=0.3,
    sm="none", spar=NULL, weight=NULL, check=FALSE)
```

## Arguments

| | |
|---|---|
| residue | matrix of an image observed at (x, y) |
| x, y | locations of regular grid at which the values in residue are measured |
| nnrow | the number of row of an input image |
| nncol | the number of column of an input image |
| tol | tolerance for stopping rule of sifting |
| max.sift | the maximum number of sifting |
| boundary | specifies boundary condition from “none", “symmetric" or “reflexive". |
| boundperc | expand an image by adding specified percentage of image at the boundary when boundary condition is 'symmetric' or 'reflexive'. |
| sm | specifies whether envelop is constructed by interpolation, thin-plate smoothing, Kriging, local polynomial smoothing, or loess. Use “none" for interpolation, “Tps" for thin-plate smoothing, “mKrig" for Kriging, “locfit" for local polynomial smoothing, or “loess" for loess. |
| spar | specifies user-supplied smoothing parameter of thin-plate smoothing, Kriging, local polynomial smoothing, or loess. |
| weight | deprecated. |
| check | specifies whether the sifting process is displayed. If check=TRUE, click the plotting area to start the next step. |

## Details

This function extracts the bidimensional intrinsic mode function from given image utilizing extrema detection based on the equivalence relation between neighboring pixels. See Kim et al. (2012) for detalis. See Kim et al. (2012) for detalis.

## Value

| | |
|---|---|
| imf | two dimensional IMF |
| residue | residue signal after extracting the finest IMF from residue |
| maxindex | index of maxima |
| minindex | index of minima |
| niter | number of iteration obtaining the IMF |

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

Kim, D., Park, M. and Oh, H.-S. (2012) Bidimensional Statistical Empirical Mode Decomposition. *IEEE Signal Processing Letters*, **19**, 191–194, doi: 10.1109/LSP.2012.2186566.

## See Also

`extrema2dC`, `emd2d`.

## Examples

```
data(lena)
z <- lena[seq(1, 512, by=4), seq(1, 512, by=4)]

## Not run:
lenaimf1 <- extractimf2d(z, check=FALSE)
## End(Not run)
```

---

`extrema` *Finding Local Extrema and Zero-crossings*

---

## Description

This function indentifies extrema and zero-crossings.

## Usage

```
extrema(y, ndata = length(y), ndatam1 = ndata - 1)
```

## Arguments

| | |
|---|---|
| `y` | input signal |
| `ndata` | the number of observation |
| `ndatam1` | the number of observation - 1 |

## Details

This function indentifies extrema and zero-crossings.

## Value

| | |
|---|---|
| minindex | matrix of time index at which local minima are attained. Each row specifies a starting and ending time index of a local minimum |
| maxindex | matrix of time index at which local maxima are attained. Each row specifies a starting and ending time index of a local maximum. |
| nextreme | the number of extrema |
| cross | matrix of time index of zero-crossings. Each row specifies a starting and ending time index of zero-crossings. |
| ncross | the number of zero-crossings |

## See Also

extrema2dC, extractimf, emd.

## Examples

```
y <- c(0, 1, 2, 1, -1, 1:4, 5, 6, 0, -4, -6, -5:5, -2:2)
#y <- c(0, 0, 0, 1, -1, 1:4, 4, 4, 0, 0, 0, -5:5, -2:2, 2, 2)
#y <- c(0, 0, 0, 1, -1, 1:4, 4, 4, 0, 0, 0, -5:5, -2:2, 0, 0)

plot(y, type = "b"); abline(h = 0)
extrema(y)
```

---

# extrema2dC *Finding Local Extrema*

---

## Description

This function finds the bidimensional local extrema based on the equivalence relation between neighboring pixels.

## Usage

```
extrema2dC(z, nnrow=nrow(z), nncol=ncol(z))
```

## Arguments

| | |
|---|---|
| z | matrix of an input image |
| nnrow | the number of row of an input image |
| nncol | the number of column of an input image |

## Details

This function finds the bidimensional local extrema based on the equivalence relation between neighboring pixels. See Kim et al. (2012) for detalis.

## Value

| | |
|---|---|
| minindex | index of minima. Each row specifies index of local minimum. |
| maxindex | index of maxima. Each row specifies index of local maximum. |

## References

Kim, D., Park, M. and Oh, H.-S. (2012) Bidimensional Statistical Empirical Mode Decomposition. *IEEE Signal Processing Letters*, **19**, 191–194, doi: 10.1109/LSP.2012.2186566.

## See Also

extrema, , extractimf2d, emd2d.

## Examples

```
data(lena)
z <- lena[seq(1, 512, by=4), seq(1, 512, by=4)]

par(mfrow=c(1,3), mar=c(0, 0.5, 2, 0.5))
image(z, main="Lena", xlab="", ylab="", col=gray(0:100/100), axes=FALSE)

example <- extrema2dC(z=z)
localmin <- matrix(256, 128, 128)
localmin[example$minindex] <- z[example$minindex]
image(localmin, main="Local minimum", xlab="", ylab="", col=gray(0:100/100), axes=FALSE)

localmax <- matrix(0, 128, 128)
localmax[example$maxindex] <- z[example$maxindex]
image(localmax, main="Local maximum", xlab="", ylab="", col=gray(0:100/100), axes=FALSE)
```

---

# hilbertspec *Hilbert Transform and Instantaneous Frequency*

---

## Description

This function calculates the amplitude and instantaneous frequency using Hilbert transform.

## Usage

```
hilbertspec(xt, tt=NULL, central=FALSE)
```

## Arguments

| | |
|---|---|
| xt | matrix of multiple signals. Each column represents a signal. |
| tt | observation index or time index |
| central | If central=TRUE, use central difference method to calculate the instantaneous frequency |

## Details

This function calculates the amplitude and instantaneous frequency using Hilbert transform.

## Value

| | |
|---|---|
| `amplitude` | matrix of amplitudes for multiple signals `xt` |
| `instantfreq` | matrix of instantaneous frequencies for multiple signals `xt` |
| `energy` | cumulative energy of multiple signals |

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

Dasios, A., Astin, T. R. and McCann C. (2001) Compressional-wave Q estimation from full-waveform sonic data. *Geophysical Prospecting*, **49**, 353–373.

## See Also

`spectrogram`.

## Examples

```
tt <- seq(0, 0.1, length = 2001)[1:2000]
f1 <- 1776; f2 <- 1000
xt <- sin(2*pi*f1*tt) * (tt <= 0.033 | tt >= 0.067) + sin(2*pi*f2*tt)

### Before treating intermittence
interm1 <- emd(xt, tt, boundary="wave", max.imf=2, plot.imf=FALSE)
### After treating intermittence
interm2 <- emd(xt, tt, boundary="wave", max.imf=2, plot.imf=FALSE,
interm=0.0007)

par(mfrow=c(2,1), mar=c(2,2,2,1))
test1 <- hilbertspec(interm1$imf)
spectrogram(test1$amplitude[,1], test1$instantfreq[,1])

test2 <- hilbertspec(interm2$imf, tt=tt)
spectrogram(test2$amplitude[,1], test2$instantfreq[,1])
```

---

# imageEMD *Plot of Bidimenasional Empirical Mode Decomposition Result*

---

## Description

This function draws plots of input image, IMF's, residue and extrema.

### Usage

```
imageEMD(z = z, emdz, extrema = FALSE, ...)
```

### Arguments

| | |
|---|---|
| z | matrix of an image |
| emdz | decomposition result |
| extrema | specifies whether the extrma is displayed according to the level of IMF |
| ... | the usual arguments to the image function |

### Details

This function draws plots of input image, IMF's, residue and extrema.

### Examples

```
data(lena)
z <- lena[seq(1, 512, by=4), seq(1, 512, by=4)]
image(z, main="Lena", xlab="", ylab="", col=gray(0:100/100), axes=FALSE)

## Not run:
lenadecom <- emd2d(z, max.imf = 4)
imageEMD(z=z, emdz=lenadecom, extrema=TRUE, col=gray(0:100/100))
## End(Not run)
```

---

## kospi200 *Korea Stock Price Index 200*

---

### Description

the weekly KOSPI 200 index from January, 1990 to February, 2007.

### Usage

```
data(kospi200)
```

### Format

A list of date and KOSPI200 index

### Details

See Kim and Oh (2009) for the analysis for kospi200 data using EMD.

### References

Kim, D. and Oh, H.-S. (2009) A Multi-Resolution Approach to Non-Stationary Financial Time Series Using the Hilbert-Huang Transform. *The Korean Journal of Applied Statistics*, **22**, 499–513.

### Examples

```
data(kospi200)
names(kospi200)
plot(kospi200$date, kospi200$index, type="l")
```

---

## lena *Gray Lena image*

---

### Description

A 512x512 gray image of Lena.

### Usage

```
data(lena)
```

### Format

A 512x512 matrix.

### Examples

```
data(lena)
image(lena, col=gray(0:100/100), axes=FALSE)
```

---

## lennon *Gray John Lennon image*

---

### Description

A 256x256 gray image of John Lennon.

### Usage

```
data(lennon)
```

### Format

A 256x256 matrix.

### Examples

```
data(lennon)
image(lennon, col=gray(100:0/100), axes=FALSE)
```

semd *Statistical Empirical Mode Decomposition*

## Description

This function performs empirical mode decomposition using spline smoothing not interpolation for sifting process. The smoothing parameter is automatically detemined by cross-validation.

## Usage

```
semd(xt, tt=NULL, cv.kfold, cv.tol=0.1^1, cv.maxiter=20,
    emd.tol=sd(xt)*0.1^2, max.sift=20, stoprule="type1", boundary="periodic",
    smlevels=1, max.imf=10)
```

## Arguments

| | |
|---|---|
| xt | observation or signal observed at time tt |
| tt | observation index or time index |
| cv.kfold | the number of fold of cross-validation |
| cv.tol | tolerance for cross-validation |
| cv.maxiter | maximum iteration for cross-validation |
| emd.tol | tolerance for stopping rule of sifting. If stoprule=type5, the number of iteration for S stoppage criterion. |
| max.sift | the maximum number of sifting |
| stoprule | stopping rule of sifting. The type1 stopping rule indicates that absolute values of envelope mean must be less than the user-specified tolerance level in the sense that the local average of upper and lower envelope is zero. The stopping rules type2, type3, type4 and type5 are the stopping rules given by equation (5.5) of Huang et al. (1998), equation (11a), equation (11b) and S stoppage of Huang and Wu (2008), respectively. |
| boundary | specifies boundary condition from “none", “wave", “symmetric", “periodic" or “evenodd". See Zeng and He (2004) for evenodd boundary condition. |
| smlevels | specifies which level of the IMF is obtained by smoothing spline. |
| max.imf | the maximum number of IMF's |

## Details

This function performs empirical mode decomposition using spline smoothing not interpolation for sifting process. The smoothing parameter is automatically detemined by cross-validation. Optimization is done by golden section search. See Kim et al. (2012) for details.

## Value

| | |
|---|---|
| imf | IMF's |
| residue | residue signal after extracting IMF's from observations xt |
| nimf | the number of IMF's |
| optlambda | smoothing parameter minimizing prediction errors of cross-validation |
| lambdaconv | a sequence of smoothing parameters for searching optimal smoothing papameter |
| perr | prediction errors of cross-validation according to lambdaconv |

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

Huang, N. E. and Wu, Z. (2008) A review on Hilbert-Huang Transform: Method and its applications to geophysical studies. *Reviews of Geophysics*, **46**, RG2006.

Kim, D., Kim, K.-O. and Oh, H.-S. (2012) Extending the Scope of Empirical Mode Decomposition using Smoothing. *EURASIP Journal on Advances in Signal Processing*, **2012:168**, doi: 10.1186/1687-6180-2012-168.

Zeng, K and He, M.-X. (2004) A simple boundary process technique for empirical mode decomposition. *Proceedings of 2004 IEEE International Geoscience and Remote Sensing Symposium*, **6**, 4258–4261.

## See Also

extractimf, emd.

## Examples

```
ndata <- 2048
tt <- seq(0, 9, length=ndata)
xt <- sin(pi * tt) + sin(2* pi * tt) + sin(6 * pi * tt)  + 0.5 * tt
set.seed(1)
xt <- xt + rnorm(ndata, 0, sd(xt)/5)

## Not run:
### Empirical Mode Decomposition by Interpolation
emdbyint <- emd(xt, tt, max.imf = 5, boundary = "wave")
### Empirical Mode Decomposition by Smoothing
emdbysm <- semd(xt, tt, cv.kfold=4, boundary="wave", smlevels=1, max.imf=5)

par(mfcol=c(6,2), mar=c(2,2,2,1), oma=c(0,0,2,0))
rangext <- range(xt); rangeimf <- rangext - mean(rangext)
plot(tt, xt, xlab="", ylab="", main="signal", ylim=rangext, type="l")
mtext("Decomposition by EMD", side = 3, line = 2, cex=0.85, font=2)
plot(tt, emdbyint$imf[,1], xlab="", ylab="", main="imf 1", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
plot(tt, emdbyint$imf[,2], xlab="", ylab="", main="imf 2", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
```

```
plot(tt, emdbyint$imf[,3], xlab="", ylab="", main="imf 3", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
plot(tt, emdbyint$imf[,4], xlab="", ylab="", main="imf 4", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
plot(tt, emdbyint$imf[,5]+emdbyint$residue, xlab="", ylab="", main="remaining signal",
    ylim=rangext, type="l")

plot(tt, xt, xlab="", ylab="", main="signal", ylim=rangext, type="l")
mtext("Decomposition by SEMD", side = 3, line = 2, cex=0.85, font=2)
plot(tt, emdbysm$imf[,1], xlab="", ylab="", main="noise", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
plot(tt, emdbysm$imf[,2], xlab="", ylab="", main="imf 1", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
plot(tt, emdbysm$imf[,3], xlab="", ylab="", main="imf 2", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
plot(tt, emdbysm$imf[,4], xlab="", ylab="", main="imf 3", ylim=rangeimf,  type="l")
abline(h=0, lty=2)
plot(tt, emdbysm$residue, xlab="", ylab="", main="residue", ylim=rangext, type="l")
## End(Not run)
```

---

`solar` *Solar Irradiance Proxy Data*

---

## Description

solar irradiance proxy data.

Hoyt and Schatten (1993) reconstructed solar irradiance (from 1700 through 1997) using the amplitude of the 11-year solar cycle together with a long term trend estimated from solar-like stars. They put relatively more weight on the length of the 11-year cycle.

Lean et al. (1995) reconstructed solar irradiance (from 1610 through 2000) using the amplitude of the 11-year solar cycle and a long term trend estimated from solar-like stars.

10-Beryllium (10Be) is measured in polar ice from 1424 through 1985. 10-Beryllium (10Be) is produced in the atmosphere by incoming cosmic ray flux, which in turn is influenced by the solar activity. The higher the solar activity, the lower the flux of cosmic radiation entering the earth atmosphere and therefore the lower the production rate of 10Be. The short atmospheric lifetime of 10Be of one to two years (Beer et al. 1994) allows the tracking of solar activity changes and offers an alternative way to the sunspot based techniques for the analysis of the amplitude and length of the solar cycle as well as for low frequency variations.

## Usage

```
data(solar.hs)
data(solar.lean)
data(beryllium)
```

## Format

A list of year and solar (solar irradiance proxy data) for solar.hs and solar.lean A list of year and be (10-Beryllium) for beryllium

## References

Beer, J., Baumgartner, S., Dittrich-Hannen, B., Hauenstein, J., Kubik, P., Lukasczyk, C., Mende, W., Stellmacher, R. and Suter, M. (1994) Solar variability traced by cosmogenic isotopes. *In: Pap, J.M., FrQohlich, C., Hudson, H.S., Solanki, S. (Eds.), The Sun as a Variable Star: Solar and Stellar Irradiance Variations*, Cambridge University Press, Cambridge, 291–300.

Beer, J., Mende, W. and Stellmacher, R. (2000) The role of the sun in climate forcing. *Quaternary Science Reviews*, **19**, 403–415.

Hoyt, D. V and, Schatten, K. H. (1993) A discussion of plausible solar irradiance variations, 1700–1992. *Journal of Geophysical Research*, **98 (A11)**, 18,895–18,906.

Lean, J. L., Beer, J. and Bradley, R. S. (1995) Reconstruction of solar irradiance since 1610: Implications for climate change. *Geophysical Research Letters*, **22 (23)**, 3195–3198.

Oh, H-S, Ammann, C. M., Naveau, P., Nychka, D. and Otto-Bliesner, B. L. (2003) Multi-resolution time series analysis applied to solar irradiance and climate reconstructions. *Journal of Atmospheric and Solar-Terrestrial Physics*, **65**, 191–201.

## Examples

```
data(solar.hs)
names(solar.hs)
plot(solar.hs$year, solar.hs$solar, type="l")

data(solar.lean)
names(solar.lean)
plot(solar.lean$year, solar.lean$solar, type="l")

data(beryllium)
names(beryllium)
plot(beryllium$year, beryllium$be, type="l")
```

---

`spectrogram` *Spectrogram*

---

## Description

This function produces image of amplitude by time index and instantaneous frequency. The horizontal axis represents time, the vertical axis is instantaneous frequency, and the color of each point in the image represents amplitude of a particular frequency at a particular time.

## Usage

```
spectrogram(amplitude, freq, tt = NULL, multi = FALSE,
nlevel = NULL, size = NULL)
```

## Arguments

| | |
|---|---|
| amplitude | vector or matrix of amplitudes for multiple signals |
| freq | vector or matrix of instantaneous frequencies for multiple signals |
| tt | observation index or time index |
| multi | specifies whether spectrograms of multiple signals are separated or not. |
| nlevel | the number of color levels used in legend strip |
| size | vector of image size. |

## Details

This function produces image of amplitude by time index and instantaneous frequency. The horizontal axis represents time, the vertical axis is instantaneous frequency, and the color of each point in the image represents amplitude of a particular frequency at a particular time.

## Value

image

## References

Huang, N. E., Shen, Z., Long, S. R., Wu, M. L. Shih, H. H., Zheng, Q., Yen, N. C., Tung, C. C. and Liu, H. H. (1998) The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis. *Proceedings of the Royal Society London A*, **454**, 903–995.

## See Also

hilbertspec.

## Examples

```
tt <- seq(0, 0.1, length = 2001)[1:2000]
f1 <- 1776; f2 <- 1000
xt <- sin(2*pi*f1*tt) * (tt <= 0.033 | tt >= 0.067) + sin(2*pi*f2*tt)

### Before treating intermittence
interm1 <- emd(xt, tt, boundary="wave", max.imf=2, plot.imf=FALSE)
### After treating intermittence
interm2 <- emd(xt, tt, boundary="wave", max.imf=2, plot.imf=FALSE,
interm=0.0007)

par(mfrow=c(2,1), mar=c(2,2,2,1))
test1 <- hilbertspec(interm1$imf)
spectrogram(test1$amplitude[,1], test1$instantfreq[,1])

test2 <- hilbertspec(interm2$imf, tt=tt)
spectrogram(test2$amplitude[,1], test2$instantfreq[,1])
```

sunspot *Sunspot Data*

## Description

sunspot from 1610 through 1995.

## Usage

```
data(sunspot)
```

## Format

A list of year and sunspot

## References

Oh, H-S, Ammann, C. M., Naveau, P., Nychka, D. and Otto-Bliesner, B. L. (2003) Multi-resolution time series analysis applied to solar irradiance and climate reconstructions. *Journal of Atmospheric and Solar-Terrestrial Physics*, **65**, 191–201.

## Examples

```
data(sunspot)
names(sunspot)
plot(sunspot$year, sunspot$sunspot, type="l")
```

# Index