

# REClass Function Documentation

March 15, 2019

---

atwt2d	<i>Computes A trous wavelet transform (ATWT)</i>
--------	--

---

## Description

Computes ATWT of the 2d array up to max\_scale. If max\_scale is outside the boundaries, number of scales will be reduced. Data is mirrored at the boundaries. Negative WT are removed. Not tested for non-square data.

## Usage

```
atwt2d(data2d, max_scale = -1)
```

## Arguments

data2d	2d image as array or matrix.
max_scale	computes wavelets up to max_scale. Leave blank for maximum possible scales.

## Value

array containing ATWT of input image with added 3rd dimension for scales.

## Note

Need to break this into smaller functions.

## Author(s)

Bhupendra Raut and Dileep M. Puranik

## See Also

Press et al. (1992) Numerical Recipes in C.

---

class3dTo2d	<i>2D projection of 3D convective-stratiform classes.</i>
-------------	---

---

### Description

Checks verticle profile of the classification and finds continuous regions of similar classification and assigns one dominant class. If both classes has comparable presence, mixed class is assigned.

### Usage

```
class3dTo2d(wt_class_3d, vert_clust, vert_range = 1:30)
```

### Arguments

wt_class_3d	Volume classification obtained from <a href="#">get_class</a>
vert_clust	Clusters saved in RDs files from functions <a href="#">clusterProfiles</a>
vert_range	same vert_range in as <a href="#">saveVertProf</a> and <a href="#">sampleVertProf</a>

### Value

2d array of pixels labeled with three classes. 1. stratiform, 2. Convection, 3. Mixed (Need correction)

### See Also

[get\\_class](#)

---

cleanWT	<i>Remove tiny fluctuations that may not be of interest.</i>
---------	--

---

### Description

This may not be needed for clean dataset. Use when WT has too much noise with trial and error approach.

### Usage

```
cleanWT(wt_sum, dbz_vol)
```

### Arguments

wt_scan	2d wt image at a scale or sum of several scales.
times_sd	Default value=1. pixels with WT value < mean + times_sd * SD are removed.

### Value

wt\_scan with small values removed.

---

dbz2rr	<i>computes rain rate using Z-R relationship.</i>
--------	---

---

**Description**

Uses standard values for ZRA=200 and ZRB=1.6.

**Usage**

```
dbz2rr(dbz, ZRA = 200, ZRB = 1.6)
```

**Arguments**

dbz	array, vector or matrix of reflectivity in dBZ
-----	--

**Value**

rr rain rate in mm/hr

---

getMaxScale	<i>Calculate the mximum possible scale of ATWT for given dimensions.</i>
-------------	--

---

**Description**

Calculate the mximum possible scale of ATWT for given dimensions.

**Usage**

```
getMaxScale(data_dims)
```

**Arguments**

data_dim	output of the dim(data2d) for given matrix or array.
----------	--

**Value**

integer value of the maximum scale.

---

getScaleBreak	<i>compute scale break for convection and stratiform regions.</i>
---------------	---

---

### Description

WT will be computed upto this scale and features will be designated as convection.

### Usage

```
getScaleBreak(res_km, conv_scale_km)
```

### Arguments

res_km	resolution of the image.
conv_scale_km	expected size of spatial variations due to convection.

### Value

dyadic integer scale break in pixels.

---

getWTClass	<i>Compute scan-by-scan ATWT of radar volume.</i>
------------	---

---

### Description

Converts dBZ to rain rates using standard Z-R relationship. This is to transform the normally distributed dBZ to gamma-like distribution.

### Usage

```
getWTClass(dbz_data, res_km, conv_scale_km = 20)
```

### Arguments

vol_data	3D array containing radar data. Last dimension should be levels.
res_km	resolution of the radar data in km.
conv_scale	approximate scale break (in km) between convective and stratiform scales.

### Value

Sum of wavelets upto conv\_scale for each scan.

### See Also

[getWTSum](#)

---

getWTSum	<i>returns sum of WT upto given scale.</i>
----------	--

---

**Description**

Works with both 2d scans and Volume data.

**Usage**

```
getWTSum(vol_data, conv_scale)
```

---

labelClasses	<i>Lables 1. stratiform, 2. intense/heavy convective and 3. moderate+transitional convective regions using given thersholds.</i>
--------------	--

---

**Description**

Lables 1. stratiform, 2. intense/heavy convective and 3. moderate+transitional convective regions using given thersholds.

**Usage**

```
labelClasses(wt_sum, vol_data)
```

# Index

atwt2d, [1](#)

class3dTo2d, [2](#)

cleanWT, [2](#)

clusterProfiles, [2](#)

dbz2rr, [3](#)

get\_class, [2](#)

getMaxScale, [3](#)

getScaleBreak, [4](#)

getWTClass, [4](#)

getWTSum, [4](#), [5](#)

labelClasses, [5](#)

sampleVertProf, [2](#)

saveVertProf, [2](#)