Tutorial

How to analyze your data with the Empirical Mode Decompsition (EMD) in R!

You can do it!

Renée Bichler

DLR Deutsches Zentrum für Luft- und Raumfahrt

Content

1. Empirical Mode Decomposition (EMD)

1.1. Introduction

Every method is based on different assumptions. Most of them assume linearity and in a statistical meaning stationarity but most of the time we have to deal with non-linear systems and thus with datasets that are non-stationary. To analyze these datastes, the EMD is a method which can perform on non-linear and non-stationary data.

A good introduction of to the Empirical Mode Decomposition (EMD) can be found in Kim and Oh (2009), where they describe the principle of the EMDs in a step by step procedure.

In general, this method is used to decompose a time series or signal into so-called intrinsic mode functions (IMFs) to get information about the frequency and time. An IMF is used to find oscillations which are embedded in the given time series or signal. Therefore, it is firstly necessary to identify the local extrema, minima and maxima, and interpolate them. The result should be, as Kim and Oh (2009) describe in their research paper, on the one hand an upper and on the other hand a lower envelope. In the next step, the average between the upper and lower envelope is calculated, and further subtracted from the original signal. The result after this process provides the opportunity to separate the oscillating pattern from the input signal. However, in order to be able to declare the result as an IMF, according to Huang et al. (1998), two requirements must be met:

1. The amount of the extrema and the amount of the so-called null-crossings only differ by one.
2. The calculated local average has to be zero.

In case the separated oscillating pattern does not meet these requirements, the procedure described above must be repeated on the oscillating pattern.

Huang et al. (1998) further mention that the EMD method can be used either on non-stationary or stationary datasets.

1.2. Literature

Huang, N.E., Shen, Z., Long, S.R., Wu, M.C., Shih, H.H., Zheng, Q., Yen, N.-C., Tung, C.C., 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., Oh, H.-S., 2009. EMD. A Package for Empirical Mode Decomposition and Hilbert Spectrum. The R Journal Vol.1/1, ISSN 2073-4859; online: https://journal.r-project.org/archive/2009/RJ-2009-002/RJ-2009-002.pdf (last access: 26-Jul-2021)

2. EMD in R

In case the EMD package isn't installed yet type *install.packages("EMD")* in your command line.

```
## Install Packages
install.packages("EMD")
install.packages("dplyr")
install.packages("ggplot2")
install.packages("lubridate")
```

At the beginning of the R script we make sure to load all the necessary libraries that we need for our project.

```
## Load Packages
library(EMD)
library(dplyr)
library(ggplot2)
library(lubridate)
```

In the next step we set the input file and the output path.

```
## Define input (_in) and output (_out) file
file_in <- "C:/Users/bich_re/Desktop/Tutorials/Tutorial_1/00_DATA/no2_mean_data-2021-07-26.csv"
file_out <- "C:/Users/bich_re/Desktop/Tutorials/Tutorial_1/00_DATA"
```

It is also always recommended to define variables at the beginning of your code. It will make it easier to adapt the code in case your input file changes and the column names turn out to be different.

```
## Variables
column <- "Mean_Mean"
file_tail <- strsplit(file_in,"/")[[1]][8]
filename <- strtrim(file_tail,24)
plot_title <- "Empirical Mode Decomposition for tropospheric NO2 over northern Italy"
```

Afterwards we are ready to read the data from the CSV file.

```
## Load NO2 time series as data frame
df <- read.csv(file_in)
```

Since we would like to use the date information from our dataframe (df) later in our EMD plots we are going to convert the character (you can check your columns by writing *class(df$Period1)*) into a date.

```
## Convert date column into date format
df$Period1 <- as.Date(df$Period1)
```

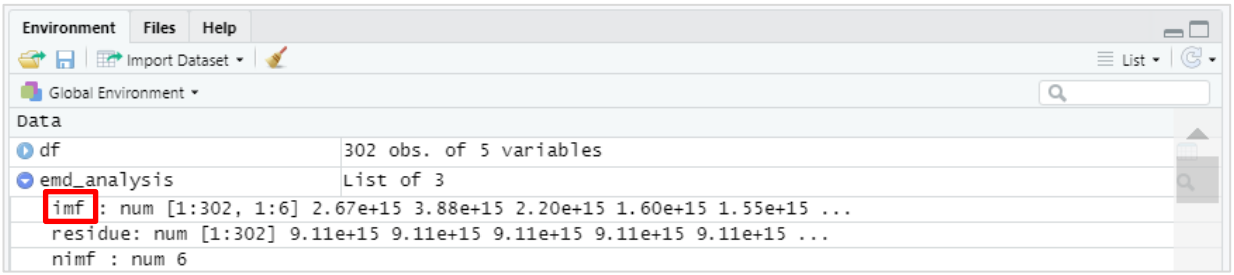
To start the EMD make sure that you loaded the EMD library at the beginning. If yes, type *emd(xt=your dataframe, etc.)*.

```
## Run EMD
emd_analysis <- emd(xt=df[[column]], boundary="wave")
```

In this tutorial we used the predefined settings but please feel free to play a little around with the settings and how it might impact the results.

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

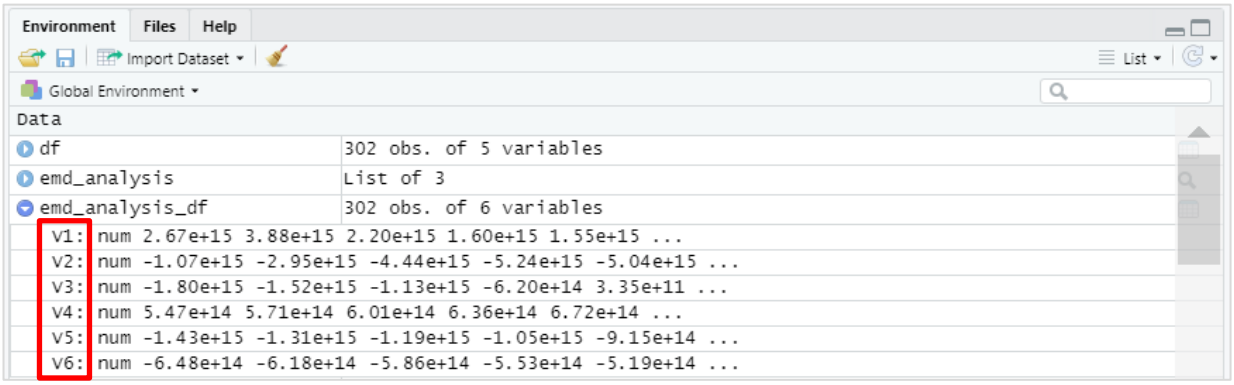
After the successful pass of the emd function you should be able to find a *imf variable* in your environment.



In the next step part of the script we want to plot each imf, in this case six single plots because of the six imf's. Therefore, we have to prepare our data with the imf information in it.

```
## Exctract the intrinsic mode functions
emd_analysis_df <- as.data.frame(emd_analysis$imf)
```

When you convert the IMF information from the emd_analysis into a dataframe you will see that the IMFs are going to be named as Vs. In a later step we want to keep the names of the IMFs but instead of calling them V1, V2, V3 etc. we want to rename them as IMF1, IMF2, IMF3 and so on.



Thus we have to change the column name.

```
## Change column name from V to IMF
col <- colnames(emd_analysis_df)
col <- gsub("V", "IMF", col)
colnames(emd_analysis_df) <- col
```

You're doing great! We are almost done!

Since the ggplot function that we are going to use to create our plots needs a specific format, we need to add some imformation to our dataframe. The %>% expression is called pipe and will be activated by loading the *dyplr library*.

```
## Create a data frame for the plot
emd_plot_df <- emd_analysis_df %>%
  mutate(date = df$Period1) %>%
  gather(key = "imf", value = "value", -date) %>%
  group_by(imf)
```

The following part of the code is only to set up the x-axis in our plot in a specific format. In the code below you will also find a function which is called *year(…)*. This function is available when loading the *lubridate library* in R.

```
## Axis labeling
start <- as.Date.numeric(summary(df$Period1)["Min."],origin = "1970-01-01")
end <- as.Date.numeric(summary(df$Period1)["Max."],origin = "1970-01-01")
```

```
xaxis_lim = scale_x_date(date_breaks = "4 years",
                                  date_minor_breaks = "2 years",
                                  limits=as.Date(c(as.character(year(start)),
                                  as.character(year(end))),"%Y"),
                         date_labels = "%Y\n(%b)")
```

Now we are able to plot our data by using ggplot from the *ggplot2 library*.

```
## Plot the IMFs
emd_plot <- ggplot(emd_plot_df, aes(x = date, y = value)) +
  geom_line(color="grey45",size=0.6) +
  facet_wrap( ~ imf) +
  xaxis_lim +
  labs(title = plot_title, x = NULL, y = NULL) +
  theme(plot.title = element_text(size=18))
```

As we would like to save the plots in a folder on our computer for a later use we have to add the ggsave function as follows:

```
## Save the plot
output1 <- paste0(file_out, "/01_EMD_", filename, ".png")

ggsave(filename=output1, plot=emd_plot,
       scale=1, width = 340, height = 178, units = "mm",
       dpi = 400, limitsize = TRUE)
```

Finally, we also set up a dataframe which makes it easy to comprehend what were the results of the EMD analysis in case we need the information later or in case we decide to adapt the plots for a research paper or a later presentation. Therefore, we put everything into a clean and easy to read structure and export the dataframe.

```
## Prepare dataframe for download

## Add columns to the data frame
emd_analysis_df["date"] <- df$date
emd_analysis_df["residue"] <- emd_analysis$residue
emd_analysis_df["nimf"] <- emd_analysis$nimf

emd_df <- as.data.frame(emd_analysis_df)
```

Last but not least, we export our EMD analysis into a new CSV file.

```
output2 <- paste0(file_out, "/01_EMD_", filename, ".csv")
write.csv(emd_df,output2)
```

I am impressed! You did it!

2.1. R Package

Grolemund, G., Wickham, H., 2011. Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25; Online: https://www.jstatsoft.org/v40/i03/ (last access: 31-Mar-2022)

Kim, D., Oh, H.-S., 2009. EMD: A Package for Empirical Mode Decomposition and Hilbert Spectrum. The R Journal, 1, 40-46. Online: https://journal.r-project.org/archive/2009/RJ-2009-002/RJ-2009-002.pdf (last access: 31-Mar-2022)

Kim, D., Oh, H.-S., 2021. EMD: Empirical Mode Decomposition and Hilbert Spectral Analysis. R package version 1.5.9; Online: https://cran.r-project.org/web/packages/EMD/EMD.pdf (last access: 31-Mar-2022)

Wickham, H., 2016. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016; Online: https://cran.r-project.org/web/packages/ggplot2/ggplot2.pdf (last access: 31-Mar-2022)

Wickham, H., François, R., Henry, L., Müller, K., 2021. dplyr: A Grammar of Data Manipulation. R package version 1.0.7; Online: https://CRAN.R-project.org/package=dplyr (last access: 31-Mar-2022)

2.2. Results

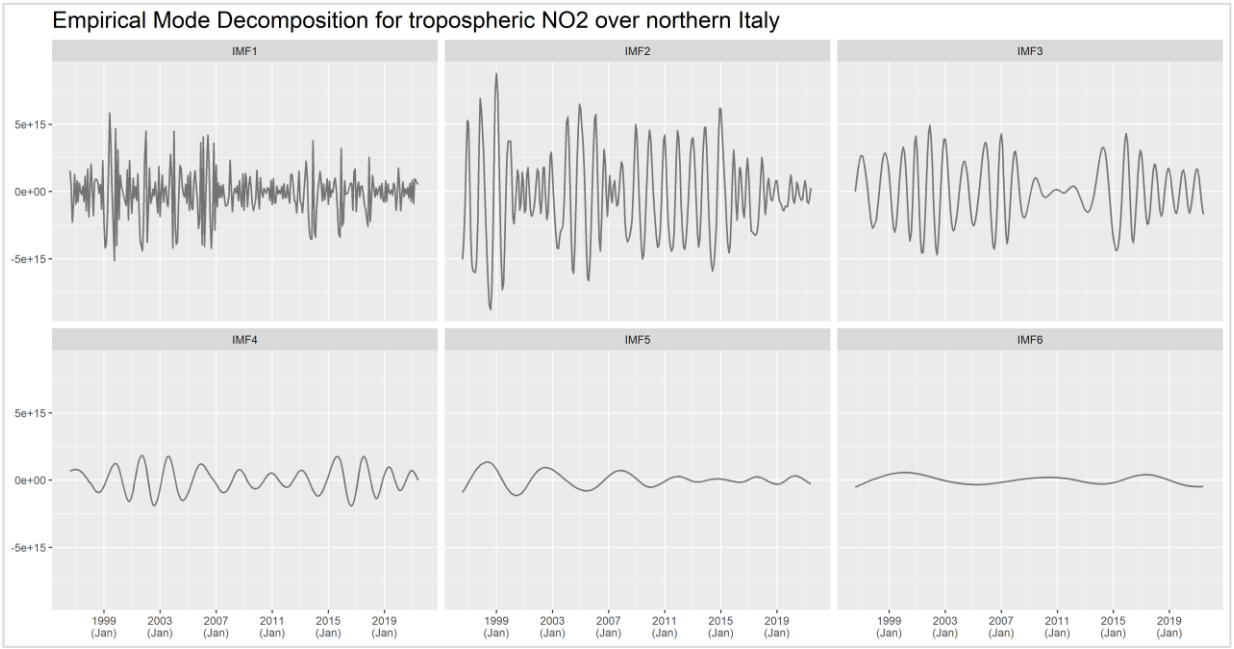


Figure 1: Intrinsic Mode Functions (IMFs) of the Empirical Mode Decomposition (EMD) based on tropospheric NO_2 column density data from 1996 to 2021 over northern Italy.

Great job!