# Speech Signal Analysis

The various signals available for study are:

- `test44100.wav` (Sampling frequency Fe = 44100 Hz like for CDs)
- `test22050.wav` (Sampling frequency Fe = 22050 Hz)
- `test11025.wav` (Sampling frequency Fe = 11025 Hz)
- `test8000.wav` (Sampling frequency Fe = 8000 Hz like for telephone)
- `test8000slow.wav` (Sampling frequency Fe = 8000 Hz and slow elocution)
- `test8000fast.wav` (Sampling frequency Fe = 8000 Hz and fast elocution)

Each of them contains the sentence: "Voici un segment contenant de la parole pour faire de l'analyse spectrale et cepstrale."

To access for instance test8000.wav with matlab, use the following command:

```
s8000 = wavread('test8000.wav');
```

and you will have the variable `s8000` containing the samples of the speech segment. Then you can visualize it with the command:

```
plot(s8000)
```

Start working with `test44100.wav`. Then study the differences when using the other signals.

1. **Slicing**
   Slice the signal into frames of 30 ms every 10 ms to form the matrix S. The number of samples to take to have 30 ms depends of course on the sampling frequency. Each column of S represents a frame of the signal. Visualize a few of them using the `plot` command.
2. **Windowing**
   Apply a Hamming window on each column of S to form the matrix H. Each column of H represents a windowed frame of the signal. Visualize a few of them.

3. **Spectrum**
   Apply a FFT with N = 512 to form the matrix F containing complex numbers. Take the modulus of F (function `abs`) to form the matrix M. And finally take 20 log(F) to form the matrix D. Since the spectrum is symmetric, keep only the top part of matrix D, that is the first 256 values of each column, to form the matrix D1. Each column of D1 contains a spectrum in dB (decibels) corresponding to the bandwidth [0 - Fe/2]. Represent a few spectra (columns of D1). Then represent the whole matrix using the command `imagesc`. This is a spectrogram.

4. **Filterbank**
   Design a filterbank with the following characteristics:

   - All the filters are triangular. The first and last filters are half a triangle.
   - The central frequencies of the filters are on a linear scale.
   - There are 30 filters.
   - The value of the summit of each triangle is 1.
   - Each filter has 256 values (same number of points as the spectra).

   When applying one filter to one spectrum, you obtain one value of the spectrum enveloppe. If you apply all the filters on one spectrum, you obtain 30 values representing the enveloppe of the corresponding spectrum. Apply all the filters to the matrix D1 to form the matrix B. The columns of B are the enveloppes of the spectra and have 30 components. Represent a few of them, then represent the whole matrix B using `imagesc`.

5. **Cepstrum**
   Apply an inverse Fourier transform (`ifft`) with N = 30 to the matrix B. Take the modulus to form the matrix C. The columns of C are the corresponding cepstra. Represent a few of them. Represent the whole matrix C using `imagesc`.