LPC-based Cross Synthesis

Guglielmo Fratticioli, Elia Pirrello

Introduction

- Implements a vocoder through cross-synthesis.
- Computes LPC coefficients using Wiener-Hopf equations and Steepest Descent algorithm.

Getting Started

- Run the `lpc.py` script.
- The `res` folder should contain `piano.wav` and `speech.wav`.

Code Overview

- Main function: `perform_lpc()`.
- Signal is divided into frames, windowed, and analyzed individually.
- LPC coefficients are computed and whitening filter coefficients are obtained.
- Convolution is done through multiplication in frequency domain.
- Zero padding is done for artifact avoidance.
- The inverse FFTs are summed and written as `output.wav`.
- Data is normalized to prevent overflow errors in the Steepest Descent algorithm.

Steepest Descent Analysis

- The steepest descent is an iterative algorithm with two main parameters: μ and ε.
- the following theoretical results were considered :
  - error

$$J(w_n) = \sigma^2{}_x - w_n^H * p - p^H * w - w_n^H * R * w_n \quad (1)$$

  - stability

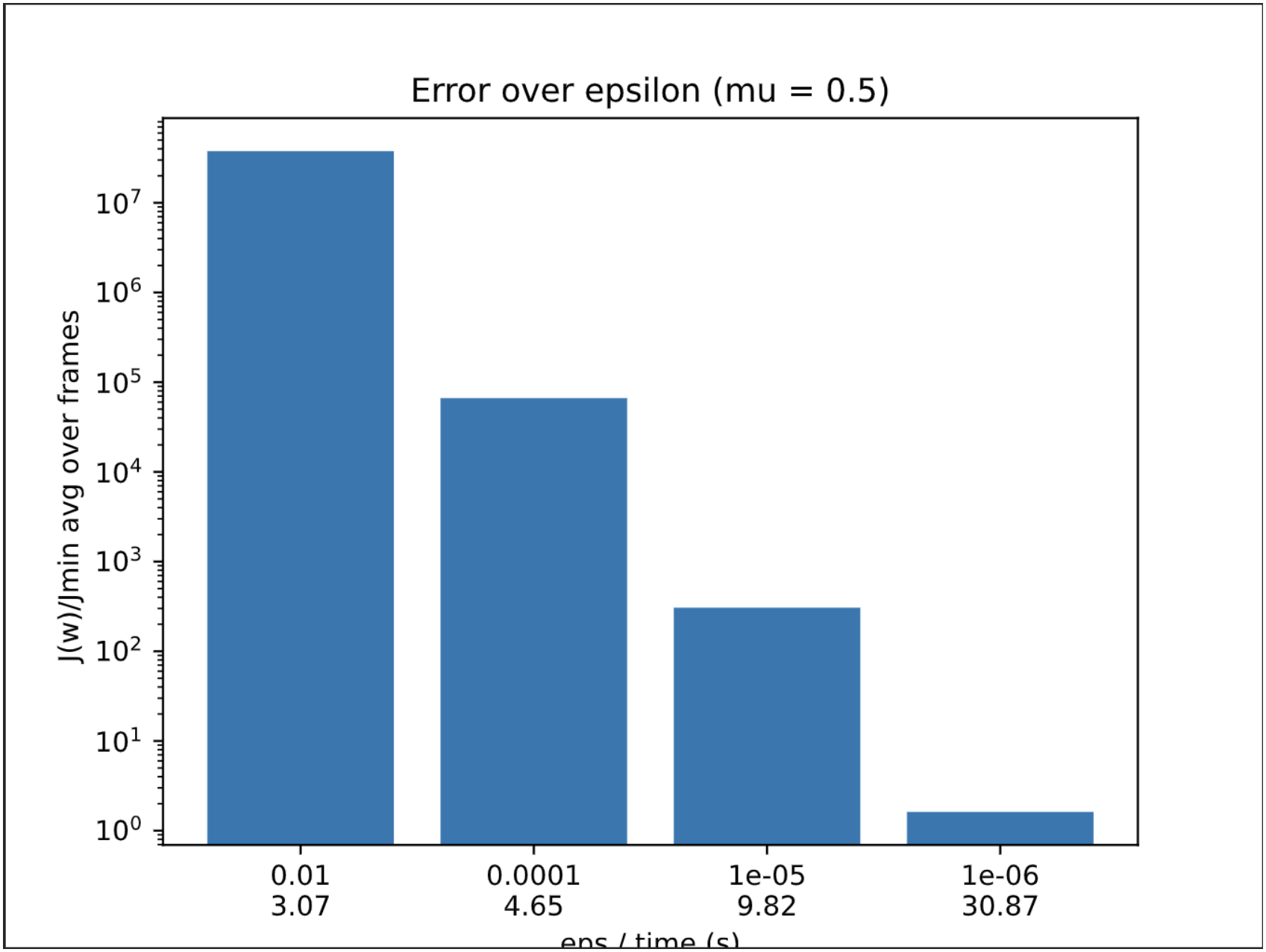
$$0 \leq \mu \leq \frac{2}{\lambda_{max}} \quad (2)$$

  - stop condition

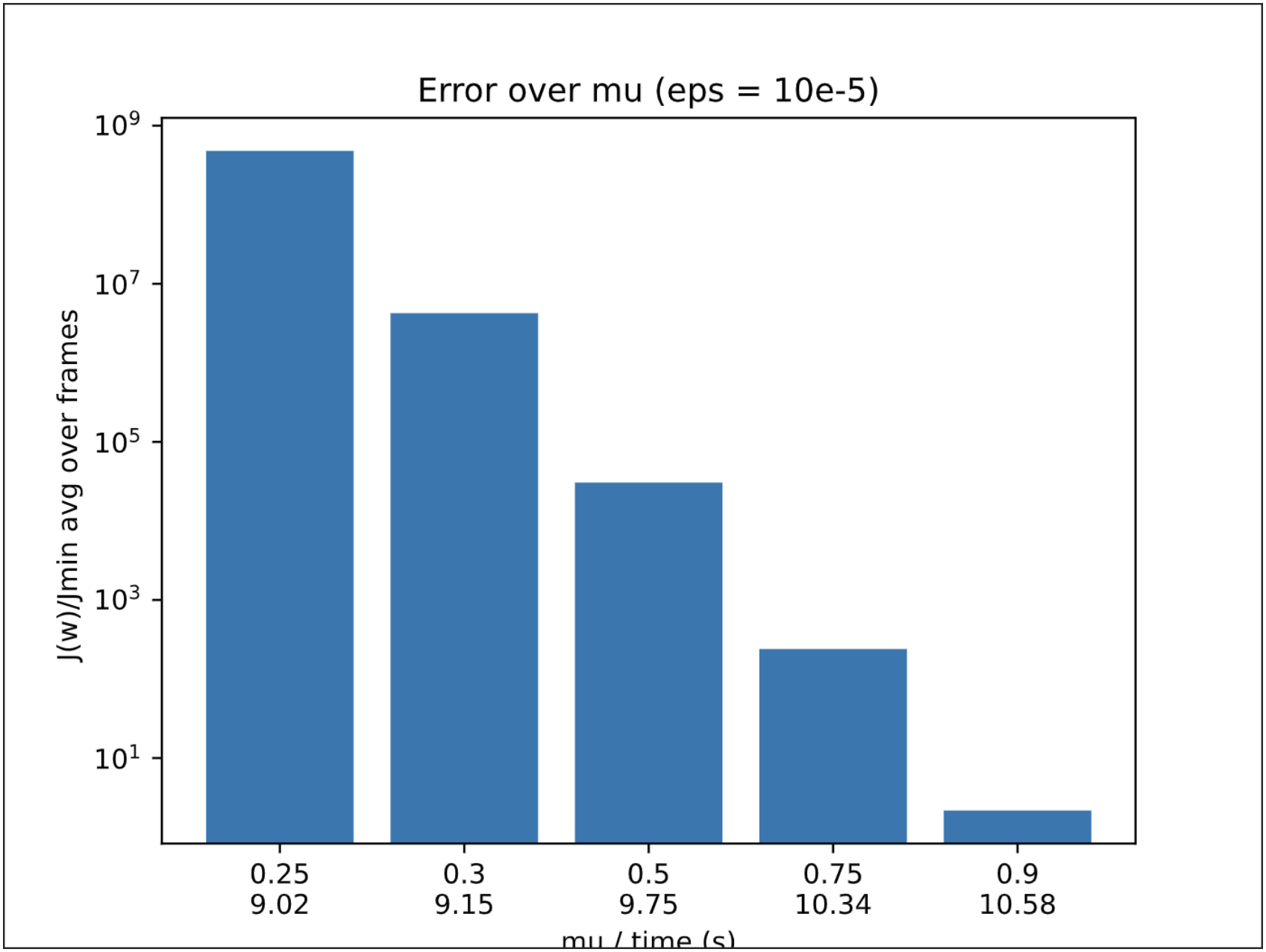
$$\Delta_J = J(w_{n+1}) - J(w_n) \leq \varepsilon \ , \ \varepsilon \in \{10^{-3}, 10^{-5}, 10^{-7}, ...\} \quad (3)$$

- the `steepest_descent_analysis()` function performs experiments by varying μ , ε

Result 1/2


Error over epsilon (mu = 0.5)
J(w)/Jmin avg over frames
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0
0.01
3.07
0.0001
4.65
1e-05
9.82
1e-06
30.87
eps / time (s)

Result 2/2


Error over mu (eps = 10e-5)
J(w)/Jmin avg over frames
10^9
10^7
10^5
10^3
10^1
0.25
9.02
0.3
9.15
0.5
9.75
0.75
10.34
0.9
10.58
mu / time (s)