# Web Audio Guitar Tube Amplifier vs Native Simulations

Michel Buffa, Jerome Lebrun

# Web Audio Guitar Tube Amplifier vs Native Simulations

Michel Buffa
Université Côte d'Azur,
CNRS, INRIA
buffa@i3s.unice.fr

Jerome Lebrun
Université Côte d'Azur,
CNRS
lebrun@i3s.unice.fr


**Figure 1: Web based tube amp simulation, GUI with visualization tools**

## ABSTRACT

We propose to present a tube guitar amplifier simulation we've been designing using the Web Audio API with the aim to faithfully reproduce the main parts of the Marshall JCM 800 amplifier schematics. Each stage of the real amp has been recreated (preamp, tone stack, reverb, power amp and speaker simulation). We've also added an extra multiband EQ. This "classic rock" amp simulation we've been building has been used in real gigs and can be favorably compared with some native amp simulation both in terms of latency, sound quality, dynamics and comfort of the guitar play. The amp is open source[1] and can be tested online[2], even without a real guitar plugged-in. It comes with an audio player, dry guitar samples and a wave generator that can be used as inputs. Figure 1 shows the current GUI, with some optional frequency analyzers and oscilloscopes that we've been using to probe the signal at different stages of the simulation. One purpose was to evaluate the limits of the Web Audio API and see if it was possible to design a web-based guitar amp simulator that could compete with native simulations.

**Figure 2: Typical setup for a demo / comparison with native amp simulations**

## 1. Introduction

Guitar amplifier digital models became popular with devices such as the pod series by Line 6 in the early 2000s. More recently, Fractal Audio Systems introduced the Axe FX-II amp modeler, an all-in-one preamp/effects digital processor which contains a vast

[1] https://github.com/micbuffa/WebAudio-Guitar-Amplifier-Simulator-3

[2] https://wasabi.i3s.unice.fr/AmpSim3 and a version with measure tools activated: https://wasabi.i3s.unice.fr/AmpSimFA

virtual inventory of hundreds of vintage and modern guitar amps. In 2002, Amplitube, an audio plugin commercialized by IK multimedia, was the first successful software amp simulation on the market, followed soon by Guitar Rig from Native Instruments. Today, we can find hundreds of native plugins (commercial or freeware, only a few are open source) for digital audio workstations that simulate existing guitar amps, or are based on an original design by their authors. Software amp simulations are popular on-the-run solutions, or when the production budget for recording is low, as they are cheaper and more flexible than their digital hardware equivalents, whereas some may claim they'll never be the same as the real thing.

## 2. Our framework

In 2012, Google Chrome was the first to propose low-latency access to live audio from a microphone or other audio input on MacOS, followed by a Windows implementation (with a longer latency). Soon Opera, Firefox and more recently Microsoft Edge also implemented this features that relies on the Media Capture and Streams API from W3C. Chris Wilson's "Input Effects" demo[3] was one of the first to show real time sound processing effects written with Web Audio. He proposed implementations of some famous effects such as delay, distortion, wah, etc. His demo did not allow to chain effects but proved that low latency processing could be achieved. However, getting close to the sound of a real guitar amplifier is a real challenge that Chris Wilson's examples did not address. Many papers have been written about vacuum-tube guitar amplifiers modeling [1][6], and about the particularities of linear and non-linear distortion effects suited for guitar [2][3][4][5]. More generally, works such as James J. Clark's "Advanced programming techniques for modular synthesizers" book, are not focused on guitar but discuss thoroughly the different approaches for achieving a distortion effect.

There are two main approaches for simulating the different parts of a guitar amplifier: one is called the technique of virtual analog (or physical modeling) and consists of entering the electronic schema in a tool like the industry standard SPICE analog circuit simulator to translate the circuit into equations to be solved. These general equations are typically nonlinear differential algebraic equations that may be solved using integration methods, roots solver algorithms, and sparse matrix techniques. SPICE can produce C++ code ready to be executed. However, it is often necessary to make simplifications and optimizations to obtain a solution suited for real-time processing. This is particularly the case with the modeling of vacuum tubes used in guitar amplifiers and their interactions with other parts of the circuitry (see [1] for a review of common techniques).

Another technique consists in a higher-level emulation, in which "logical" parts are identified (filters, tubes, etc.) and individually emulated using separate models. This is in theory less accurate as some effects and interactions such as the current feedback effect of overloaded tubes or the action of the speaker impedance on the sound tone will not be considered. However, this approach is simpler and more adapted to Web Audio and its current limitations (custom processing on audio samples with the Script Processor node is not usable without introducing latency or glitches, for example). Furthermore, Web Audio proposes some high-level nodes (such as the Wave Shaper node, the biquad filter node) that can be used for modeling tubes and filters, and it has been shown that when properly used, wave shaping techniques associated with appropriate filtering can give good results. The famous pod XT by Line 6 effect processor uses such techniques[1].

As far as we know there is no previous work where authors have tried to simulate a complete guitar amp using Web Audio. Pedals.io[4] is a JavaScript recreation of some classic audio effect pedals for guitarists (delay, chorus, overdrive, etc.), and we find nearly the same implementations of these effects in many Web Audio JavaScript libraries such as toneJS[5], tunaJS[6], pizzicatoJS[7], etc. The most advanced work we've found is a Google Chrome application named GuitarFX[8] that proposes simple amp models with a set of audio effect pedals, but does not fully recreate each stages of a real amp (only one wave shaper per amp, for example).

Our proposed demonstration is also a good test bench to assess the current limitations of Web Audio in terms of latency (driver, audio buffer size, sample rate frequency, etc.).

## 3. Settings for the demo

Our simulation can be played real-time with a real guitar. As of today, we recommend for the best experience to use MacOS and a low latency sound card. By using a real guitar, we propose to compare our Web Audio based tube amp simulation with native simulations such as Guitar Rig by Native Instruments (used by many musicians and guitarists), with GarageBand amp simulations and with Guitarix, an open source native amp simulator.

## 4. REFERENCES


[1] Pakarinen, J. and Yeh, D. T. (2009). A review of digital techniques for modeling vacuum-tube guitar amplifiers. *Computer Music Journal*, *33*(2), 85-100.

[2] Holmes, B. and van Walstijn, M. (2015, Dec.). Improving the robustness of the iterative solver in state-space modeling of guitar distortion circuitry. In *Proc. 18th Int. Conference on Digital Audio Effects (DAFx-15), Trondheim, Norway*.

[3] Chang, C. H. (2011). DESC9115: Digital Audio Systems-Final Project Overdrive/Distortion. In *Repository of Tech. Reports, University of Sydney, Australia.* http://hdl.handle.net/2123/7608

[4] Macak, J. and Schimmel, J. (2010, Sep.). Real-time guitar tube amplifier simulation using an approximation of differential equations. In *Proc. 13th International Conference on Digital Audio Effects (DAFx-10)*, *Graz, Austria.*

[5] Yeh, D. T., Abel, J. S., Vladimirescu, A. and Smith, J. O. (2008). Numerical methods for simulation of guitar distortion circuits. *Computer Music Journal*,*32*(2), 23-42.

[6] Yeh, D. T. and Smith, J. O. (2006, Sep.). Discretization of the'59 Fender Bassman tone stack. In *Proc. 9th Int. Conference on Digital Audio Effects (DAFx-06), Montreal, Canada.*


[3] https://webaudiodemos.appspot.com/input/index.html

[4] https://pedals.io

[5] https://github.com/Tonejs/Tone.js

[6] https://github.com/Theodeus/tuna

[7] https://alemangui.github.io/pizzicato

[8] https://tinyurl.com/ljdhuqh