FIXED-RATE MODELING OF AUDIO LUMPED SYSTEMS: A COMPARISON BETWEEN TRAPEZOIDAL AND IMPLICIT MIDPOINT METHODS

François G. Germain

Center for Computer Research in Music and Acoustics (CCRMA),
Stanford University, Stanford, CA, USA
francois@ccrma.stanford.edu

ABSTRACT

This paper presents a comparison framework to study the relative benefits of the typical trapezoidal method with the lesser-used implicit midpoint method for the simulation of audio lumped systems at a fixed rate. We provide preliminary tools for understanding the behavior and error associated with each method in connection with typical analysis approaches. We also show implementation strategies for those methods, including how an implicit midpoint method solution can be generated from a trapezoidal method solution and vice versa. Finally, we present some empirical analysis of the behavior of each method for a simple diode clipper circuit and provide an approach to help interpret their relative performance and how to pick the more appropriate method depending on desirable properties. The presented tools are also intended as a general approach to interpret the performance of discretization approaches at large in the context of fixed-rate simulation.

1. INTRODUCTION

Computational modeling of audio lumped systems (i.e., virtual analog modeling) is a major topic of interest, including emulation of electronic and acoustic systems such as vintage audio effects or acoustic instruments. This main goal of this modeling process is the design of discrete-time systems emulating the behavior of a continuous-time system. When the set of differential equations driving the dynamics of that system are known, a common procedure is to discretize it using a discretization schemes [1–3].

Those methods have a variety of advantages and drawbacks. Discretization schemes are generally designed following concepts such as order of accuracy and stability. These properties guarantee the versatility of those methods for consistently generating discrete-time models with some level of accuracy. In the context of virtual analog modeling, a large part of the literature has been developed focusing on using the trapezoidal method [3–10]. This method provides a good compromise between simplicity (as a one-step method), accuracy (as a second-order method) and behavior (as an unconditionally stable method). A less-studied numerical method with those same properties is the implicit midpoint method [11]. While they share those properties, the two methods differ in behavior for nonlinear systems, and results from previous papers have hinted to the possibility that the implicit midpoint method could produce better-behaved simulations for some classes of systems [12, 13]. One particular point of interest to compare the two methods is the oscillatory behavior for the simulation of stiff systems [14].

Virtual analog modeling is also generally focused on fixed-rate simulation, meaning that that controlling the error through the modulation of the simulation rate is seldom considered. This is mostly due to the computational cost of such control, as real-time simulation is a desirable property of virtual analog approaches [15]. In that context, typical error analysis methods have limited interpretation since they mostly describe the behavior of the methods as the simulation rate tends to infinity. Simulation rates in virtual analog systems tend to fall in regions where those asymptotic properties provide limited insight on the simulation behavior. We then want to draw alternative design methods for discretization methods that rather target the fixed-rate context [14, 16–18].

This paper presents a preliminary framework for the comparison and error analysis of the typical trapezoidal method and the lesser-used implicit midpoint method in the context of fixed-rate simulation of audio lumped systems, as well as practical information regarding their implementation in several existing frameworks. Sec. 2 shows the general state-space formalism that we use for our analysis. Sec. 3 shows the definition and the general properties of the two methods. Sec. 4 shows a discussion of strategies to implement the midpoint method using current virtual analog modeling approaches. Finally, Sec. 5 shows an empirical comparison of the two methods on a diode clipper system [15, 19, 20].

1.1. Notation

In this paper, we use bolded letters (e.g., $\boldsymbol{x}$, $\boldsymbol{f}$) to denote multi-dimensional variables and multi-output functions. Discrete-time sequences are denoted with an overline (e.g., $\overline{x}$, $\overline{\boldsymbol{x}}$). Superscripts are used to denote the number of a sample in a sequence (e.g., $\overline{\boldsymbol{x}}^n$ is the nth sample in the sequence $\overline{\boldsymbol{x}}$). To avoid confusion, power are indicated outside parentheses (e.g., $(\overline{\boldsymbol{x}}^3)^2$ is the 3rd sample of the sequence $\overline{\boldsymbol{x}}$ raised to the 2nd power). Subscripts are used to denote differentiation (e.g., $f_{xu}(x,u)$ is the 2nd-order derivative of the function $f(x,u)$ with respect to x and u).

2. STATE-SPACE SYSTEM REPRESENTATION

2.1. Continuous-time state-space representation

A common way of representing time-invariant lumped systems is in the so-called state-space representation, where the system is characterized by the equations [21]

$$\boldsymbol{x}_t(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{1a}$$

$$\boldsymbol{y}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad \text{and} \tag{1b}$$

$$\boldsymbol{x}(0) = \boldsymbol{x}_0, \tag{1c}$$

where $\boldsymbol{f}, \boldsymbol{g}$ are nonlinear functions, $\boldsymbol{x}$ is a vector of state variables, $\boldsymbol{u}$ is a vector of input variables, $\boldsymbol{y}$ is a vector of output variables, $\boldsymbol{x}_0$ is a vector of initial conditions. For conciseness, we omit repeating Eqs. (1b) and (1c) in the rest of the paper as they remain unchanged by the discretization process.

2.2. System discretization

The exact solution to Eqs. (1a) and (1c) is theoretically given by

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_0^t \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau. \tag{2}$$

but we can also express the solution at times $t^n = nT$ (with a fixed interval T) by solving iteratively the problem

$$\boldsymbol{x}(t_{n+1}) = \boldsymbol{x}(t_n) + \int_{t_n}^{t_{n+1}} \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))d\tau. \tag{3}$$

In typical cases, solving Eq. (2) or (3) analytically is intractable. One can however compute a discrete-time series $\overline{\boldsymbol{x}}^n$ approximating $\boldsymbol{x}(t_n)$. To do so, we use numerical integration methods to approximate the integral in Eq. (3) using some N past input values and approximated state values. The update equation of $\overline{\boldsymbol{x}}^n$ is then

$$\overline{\boldsymbol{x}}^{n+1} = \overline{\boldsymbol{x}}^n + T\overline{\boldsymbol{f}}(\overline{\boldsymbol{x}}^{n+1}, \dots, \overline{\boldsymbol{x}}^{n-N}, \overline{\boldsymbol{u}}^{n+1}, \dots, \overline{\boldsymbol{u}}^{n-N}). \tag{4}$$

Coupled with Eq. (1b), we can then form a sequence of approximated output values for our system of interest. Note that when $\overline{\boldsymbol{f}}$ depends on $\overline{\boldsymbol{x}}^{n+1}$, Eq. (4) becomes implicit and cannot always be solved analytically.

3. NUMERICAL INTEGRATION METHODS

Numerical integration methods aims at approximating the value of the integral of a function using a finite number of function evaluations. As stated earlier, we can apply those methods to Eq, (3) form a discretized update equation as in Eq. (4). In this paper, we discuss specifically two common methods, the trapezoidal method and the implicit midpoint method.

3.1. Trapezoidal method

The trapezoidal method approximates the value of f in the interval $[t, t+T]$ as the average of the function values at the start and end point of the integral to compute the integral as [22]

$$\int_t^{t+T} \boldsymbol{f}(\tau)d\tau \approx \tfrac{T}{2}(\boldsymbol{f}(t+T) + \boldsymbol{f}(t)). \tag{5}$$

Eq. (3) is then discretized to form the implicit update equation for the trapezoidal method as

$$\overline{\boldsymbol{x}}^{n+1} = \overline{\boldsymbol{x}}^n + \tfrac{T}{2}\boldsymbol{f}(\overline{\boldsymbol{x}}^{n+1}, \overline{\boldsymbol{u}}^{n+1}) + \tfrac{T}{2}\boldsymbol{f}(\overline{\boldsymbol{x}}^n, \overline{\boldsymbol{u}}^n). \tag{6}$$

3.2. Implicit midpoint method

The midpoint method approximates the value of f in the interval $[t, t+T]$ as its value at midpoint of the integral to compute the integral as [22]

$$\int_t^{t+T} \boldsymbol{f}(\tau)d\tau \approx T\boldsymbol{f}\big(t + \tfrac{T}{2}\big). \tag{7}$$

To use this approach in Eq. (3), several implementations are possible. It can be implemented as the *explicit midpoint* method, but that method requires a way to compute the state and input values at time $t_{n+\frac{1}{2}}$. Additionally, this method (also called *leapfrog* method) has poor stability properties as its stability region is reduced to the imaginary axis in the s-plane [1]. We focus instead on the *implicit midpoint* method whose implicit update equation is

$$\overline{\boldsymbol{x}}^{n+1} = \overline{\boldsymbol{x}}^n + T\boldsymbol{f}\big(\tfrac{1}{2}(\overline{\boldsymbol{x}}^{n+1} + \overline{\boldsymbol{x}}^n), \tfrac{1}{2}(\overline{\boldsymbol{u}}^{n+1} + \overline{\boldsymbol{u}}^n)\big). \tag{8}$$

For conciseness, we will refer to the implicit midpoint method simply as "midpoint method" in the rest of the paper.

3.3. Stability analysis and pole mapping

In the following sections, we use the scalar version of Eq. (4) to simplify the notation, as the results extend readily to the multidimensional case using diagonalization and multivariable calculus.

A typical way of studying the stability of discretization methods is by studying the solution to linear time-invariant ordinary differential equations (ODEs) of the form

$$x_t(t) = \lambda x(t),\ \lambda \in \mathbb{C}, \tag{9}$$

whose update equation can typically be written in the form

$$\overline{x}(t_{n+1}) = \sum_{m=0}^{N} a_m(\lambda)\overline{x}(t_{n-m}),\ a_0(\lambda), \dots, a_N(\lambda) \in \mathbb{C}. \tag{10}$$

If we denote $\overline{\lambda}_m(\lambda)$ the $N+1$ roots of the polynomial

$$p(z) = z^{N+1} - \sum_{m=0}^{N} a_m(\lambda)z^{N-m}, \tag{11}$$

the stability region of a method is then defined as the set of λ such that $\forall m \in \{0, \dots, N\}$, we have $|\overline{\lambda}_m(\lambda)| < 1$ (i.e., $\overline{\lambda}_m(\lambda)$ is inside the unit sphere). For both the trapezoidal method and the midpoint method, we have a single root $\overline{\lambda}$ written as

$$\overline{\lambda}(\lambda) = \frac{1 + T\lambda/2}{1 - T\lambda/2}, \tag{12}$$

so that the stability region corresponds to left half of the complex plane $\{\lambda, \mathrm{Re}(\lambda) < 0\}$. This means that the two methods have the desirable property of being A-stable [23]. This also means that, for linear systems, the two methods map the system poles and zeros exactly the same way. We then expect both methods to exhibit similar qualitative behavior, such as resonant peaks near the Nyquist frequency for stiff systems (i.e., systems with strongly damped poles) and frequency warping [14].

3.4. Discretization error and order of accuracy

The discretization error of a method is typically characterized using the equation $x_t(t) = f(x(t))$ by deriving the error between $x(t_{n+1})$ solution of

$$x(t_{n+1}) = x(t_n) + \int_{t_n}^{t_{n+1}} f(x(\tau))d\tau \tag{13}$$

and $\overline{x}^{n+1}$ solution of the discretized equation

$$\overline{x}^{n+1} = x(t_n) + T\overline{f}(\overline{x}^{n+1}, x(t_n), \dots, x(t_{n-N})). \tag{14}$$

This error is typically computed in terms of a polynomial in T using the Taylor expansion of $x(t_{n+1}) - \overline{x}^{n+1}$ [1] so that

$$x(t_{n+1}) - \overline{x}^{n+1} = \sum_{n=0}^{+\infty} \epsilon(t_n)T^n, \tag{15}$$

where $\epsilon(t_n)$ can be expressed in terms of function derivatives f_{x^k}. The order of accuracy of a method is then defined as the integer n for which $\epsilon_m = 0$ for all $m \leq n$. As symmetric methods, both the trapezoidal method and the midpoint method are second-order accurate ($\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$). Beyond that, for the trapezoidal method, we have the 3rd and 4th order error terms as

$$\begin{aligned} \epsilon_3^{\text{tr}} &= -\frac{(f)^2 f_{xx} + f(f_x)^2}{12} \quad \text{and} \\ \epsilon_4^{\text{tr}} &= -\frac{(f)^3 f_{xxx} + 4(f^2) f_x f_{xx} + f(f_x)^3}{24}, \end{aligned} \tag{16}$$

and, for the midpoint method, as

$$\begin{aligned} \epsilon_3^{\text{md}} &= \frac{(f)^2 f_{xx} - 2f(f_x)^2}{24} \quad \text{and} \\ \epsilon_4^{\text{md}} &= \frac{(f)^3 f_{xxx} - (f)^2 f_x f_{xx} - 4f(f_x)^3}{48}. \end{aligned} \tag{17}$$

We then see that depending on the nonlinear function characteristics, different patterns of constructive or destructive interference between the different terms will lead to different behaviors between the two methods.

3.5. First-order fixed-point behavior

Many systems of interest usually tend towards a steady-state solution after infinite time. Note however that not all systems behave this way (e.g., relaxation oscillators [24, 25]). Those steady-state (*equilibrium*) solutions x^e solve the implicit equation:

$$f(x^e) = 0. \tag{18}$$

A typical analysis of the equilibrium is to look at the value $f_x(x^e)$ to find if the equilibrium is *stable* ($f_x(x^e) < 0$) or *unstable* ($f_x(x^e) > 0$). Physical systems composed with passive and dissipative components typically present one or more stable equilibria due to the energy of the system being dissipated over time. For such equilibria, as the state variable of a system approaches x^e, the Hartman-Grobman theorem [26] guarantees they will behave similarly as the solutions of the linearized system around x^e which follow the exponentially decaying profile

$$x(t) \approx x(0)\exp(t f_x(x^e)) + x^e. \tag{19}$$

Sampled at times t_n, the update formula is expressed as

$$x(t_{n+1}) \approx (x(t_n) - x^e)\exp(T f_x(x^e)) + x^e. \tag{20}$$

By construction, the equilibria of the discretized system using either the trapezoidal rule or the midpoint rule are identical to those of the original system as they also verify Eq. (18). In the vicinity of those equilibrium, both methods then behave as the linearized:

$$\overline{x}^{n+1} \approx \alpha(\overline{x}^n - x^e) + x^e, \tag{21}$$

with α related to $f_x(e)$ following Eq. (12), meaning

$$\alpha = \frac{1 + T f_x(x^e)/2}{1 - T f_x(x^e)/2}. \tag{22}$$

The stability properties of both methods guarantees that stable equilibria ($f_x(x^e) < 0$) are necessarily stable for the discretized sequences ($|\alpha| < 1$). However, we have no guarantee on the sign of α, so that if $\alpha < 0$, the sign of $\overline{x}^n - x^e$ near the equilibrium alternates at each iteration, creating an oscillation that is not present in the original $x(t_n) - x^e$ which remains of the same sign according to Eq. (19). This oscillatory behavior matches the well-known oscillations exhibited by the solution of stiff systems discretized with the trapezoidal rule [14].

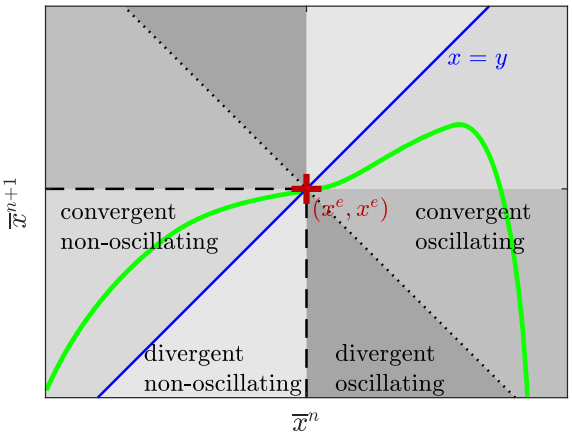


Figure 1: *Solution sequence behavior regions for $\overline{x}^{n+1}$ as a function of $\overline{x}^n$ (in green) with reference to the point (x^e, x^e).*

3.6. General fixed-point behavior

Further from an equilibrium, the first-order behavior from Eq. (21) may be insufficient to understand the behavior of the system in some regimes. Another tool we can use is to compute the transition equation $\overline{x}^{n+1}$ as a function of $\overline{x}^n$. Once that transition function is known, we can deduce regions indicating whether the method is converging or diverging, i.e., (respectively):

$$|\overline{x}^{n+1} - x^e| < |\overline{x}^n - x^e| \;\text{ or }\; |\overline{x}^{n+1} - x^e| > |\overline{x}^n - x^e| \tag{23}$$

and whether the method is oscillating or not, i.e., (respectively):

$$|\overline{x}^{n+1} - x^e| \cdot |\overline{x}^n - x^e| < 0 \;\text{ or }\; |\overline{x}^{n+1} - x^e| \cdot |\overline{x}^n - x^e| > 0 \tag{24}$$

Those properties translate graphically as shown in Fig. 1. In that representation, an quasi-exponential decay such as the one described in Eq. (21) for $\alpha > 0$ corresponds to a linear section of curve in the convergent non-oscillating region. An oscillating exponential decay ($\alpha < 0$) corresponds to a linear section of curve in the convergent oscillating region.

3.7. Discretization error with variable input

When considering the influence of the input variables, the equation of interest becomes $x_t(t) = f(x(t), u(t))$. The error terms are then expressed as function of the partial derivatives $f_{x^k u^l}$ of f.

Expectedly, with the added input variables, both methods remain second-order accurate ($\epsilon_0 = \epsilon_1 = \epsilon_2 = 0$). The 3rd-order error term for the trapezoidal method becomes

$$\begin{aligned} \epsilon_3^{\text{tr}} = -\big((f)^2 f_{xx} + 2u_t f f_{xu} + (u_t)^2 f_{uu}\big)/12 \\ - \big((f f_x + u_t f_u) f_x + u_{tt} f_u\big)/12, \end{aligned} \tag{25}$$

and one for the midpoint method becomes

$$\epsilon_3^{\mathrm{md}} = \big((f)^2 f_{xx} + 2u_t f f_{xu} + (u_t)^2 f_{uu}\big)/24 - \big((f f_x + u_t f_u) f_x + u_{tt} f_u\big)/12. \quad (26)$$

Here again, we see how the two methods will present different patterns of interference altering their behavior as a function of f as the influence of the input on the error terms differ as well.

4. IMPLEMENTATION CONSIDERATIONS

Many publications have presented various implementations of the trapezoidal method for the simulation of audio lumped systems so we refer the reader to those articles for more details [3–10]. We detail here approaches regarding the implementation of the midpoint method in typical audio circuit simulation frameworks. Generally, the implementation of the midpoint method can be derived in two ways: either through simple modifications of the system obtained using the more typical trapezoidal method or directly from the output of that trapezoidal-rule system with a simple transformation of the output as detailed below.

4.1. Direct implementation of the midpoint method

A direct implementation of the midpoint method computes the $\overline{\boldsymbol{x}}_{\mathrm{md}}^{n+1}$ by solving the implicit equation given in Eq. (8), similarly as what we do for the trapezoidal method when solving Eq. (6). Depending on the equation, similar strategies can be exploited, using analytical inverse functions when available or numerical root-finding methods otherwise. One potential downside of the midpoint method is that the expressions relative to the time step t_{n+1} and the time step t_n are grouped together inside the nonlinear function $\boldsymbol{f}$ which may complicate the derivation and use of analytical inverse functions. On the other hand, notice that the update equation does not depend on two independent input samples $\overline{\boldsymbol{u}}_{\mathrm{md}}^{n+1}$ and $\overline{\boldsymbol{u}}_{\mathrm{md}}^{n}$ but on the combined $\frac{1}{2}(\overline{\boldsymbol{u}}_{\mathrm{md}}^{n+1}+\overline{\boldsymbol{u}}_{\mathrm{md}}^{n})$ so that the dimensionality of the update equation with respect to the input can be reduced compared to the trapezoidal method.

4.2. Midpoint method using the trapezoidal rule

The midpoint method and the trapezoidal method are conjugate methods [11]. More precisely, if the sequence $\overline{\boldsymbol{x}}_{\mathrm{md}}^{n}$ verifies Eq. (8) for the input sequence $\overline{\boldsymbol{u}}_{\mathrm{md}}^{n}$, we can show that the sequence $\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n} = \frac{1}{2}\big(\overline{\boldsymbol{x}}_{\mathrm{md}}^{n} + \overline{\boldsymbol{x}}_{\mathrm{md}}^{n-1}\big)$ verifies Eq. (6) for the modified input sequence $\overline{\boldsymbol{u}}_{\mathrm{tr}}^{n} = \frac{1}{2}\big(\overline{\boldsymbol{u}}_{\mathrm{md}}^{n} + \overline{\boldsymbol{u}}_{\mathrm{md}}^{n-1}\big)$ [27]. This means that if we want to implement the midpoint rule to simulate the state equation

$$\boldsymbol{x}_t(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \quad (27)$$

we can do so by simulating that same equation using the trapezoidal rule replacing the input $\boldsymbol{u}(t)$ by $\frac{1}{2}(\boldsymbol{u}(t) + \boldsymbol{u}(t-T))$. As a consequence, if a system has been designed to generate a simulated sequence $\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n}$ of equation (27) using the trapezoidal rule, we can obtain a simulated sequence $\overline{\boldsymbol{x}}_{\mathrm{md}}^{n}$ for the input sequence $\overline{\boldsymbol{u}}_{\mathrm{md}}^{n}$ iteratively as follows:

- Generate a new sample $\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1}$ from the trapezoidal rule system using the input samples defined as

$$\begin{cases} \overline{\boldsymbol{u}}_{\mathrm{tr}}^{n+1} = \frac{1}{2}\big(\overline{\boldsymbol{u}}_{\mathrm{md}}^{n+1} + \overline{\boldsymbol{u}}_{\mathrm{md}}^{n}\big), \\ \overline{\boldsymbol{u}}_{\mathrm{tr}}^{n} = \frac{1}{2}\big(\overline{\boldsymbol{u}}_{\mathrm{md}}^{n} + \overline{\boldsymbol{u}}_{\mathrm{md}}^{n-1}\big). \end{cases} \quad (28)$$

- Generate a new sample $\overline{\boldsymbol{x}}_{\mathrm{md}}^{n+1}$ using either the (recurrent) equation:

$$\overline{\boldsymbol{x}}_{\mathrm{md}}^{n+1} = 2\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1} - \overline{\boldsymbol{x}}_{\mathrm{tr}}^{n} \quad (29)$$

or the (direct) equation:

$$\overline{\boldsymbol{x}}_{\mathrm{md}}^{n+1} = \overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1} + \tfrac{T}{2}\boldsymbol{f}\big(\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1}, \overline{\boldsymbol{u}}_{\mathrm{tr}}^{n+1}\big). \quad (30)$$

We also need to use the appropriate initial conditions for the trapezoidal-rule system based on the initial conditions of the desired midpoint sequence. In a typical case, those initial conditions correspond to $\overline{\boldsymbol{x}}_{\mathrm{md}}^{0} = \boldsymbol{x}(0)$ for the state variables and $\overline{\boldsymbol{u}}_{\mathrm{md}}^{0} = \boldsymbol{u}(t_0)$ for the input variables, typically based on a specified input function $\boldsymbol{u}(t)$ defined for $t \geq 0$. In addition to specifying the initial state value $\overline{\boldsymbol{x}}_{\mathrm{tr}}^{0}$, the trapezoidal-rule system requires to also define $\overline{\boldsymbol{u}}_{\mathrm{tr}}^{0}$ which depends on the unspecified $\overline{\boldsymbol{u}}_{\mathrm{md}}^{-1}$. Note that $\overline{\boldsymbol{x}}_{\mathrm{tr}}^{0}$ and $\overline{\boldsymbol{u}}_{\mathrm{tr}}^{0}$ cannot be chosen independently. We can pick any initial condition pair $(\overline{\boldsymbol{x}}_{\mathrm{tr}}^{0}, \overline{\boldsymbol{u}}_{\mathrm{tr}}^{0})$ as long as it verifies

$$\overline{\boldsymbol{x}}_{\mathrm{tr}}^{0} + \tfrac{T}{2}\boldsymbol{f}(\overline{\boldsymbol{x}}_{\mathrm{tr}}^{0}, \overline{\boldsymbol{u}}_{\mathrm{tr}}^{0}) = \overline{\boldsymbol{x}}_{\mathrm{md}}^{0}. \quad (31)$$

A typical assumption is that the system was in a steady-state condition before the simulation started, so that the initial condition are given as a function of $\overline{\boldsymbol{x}}_{\mathrm{md}}^{0}$ by solving

$$\begin{cases} \overline{\boldsymbol{x}}_{\mathrm{tr}}^{0} = \overline{\boldsymbol{x}}_{\mathrm{md}}^{0}, \\ \boldsymbol{0} = \boldsymbol{f}(\overline{\boldsymbol{x}}_{\mathrm{md}}^{0}, \overline{\boldsymbol{u}}_{\mathrm{tr}}^{0}). \end{cases} \quad (32)$$

4.3. Trapezoidal method using the midpoint rule

A similar principle can be used to generate a simulation based on the trapezoidal rule using a system designed to simulate that same system using the midpoint rule using the procedure:

- Generate a new sample $\overline{\boldsymbol{u}}_{\mathrm{md}}^{n+1}$ from the midpoint rule system using the input sample defined recursively as

$$\overline{\boldsymbol{u}}_{\mathrm{md}}^{n+1} = 2\overline{\boldsymbol{u}}_{\mathrm{tr}}^{n+1} - \overline{\boldsymbol{u}}_{\mathrm{md}}^{n}. \quad (33)$$

- Generate a new sample $\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1}$ using either

$$\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1} = \tfrac{1}{2}\big(\overline{\boldsymbol{x}}_{\mathrm{md}}^{n+1} + \overline{\boldsymbol{x}}_{\mathrm{md}}^{n}\big) \quad (34)$$

or solving the implicit equation

$$\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1} + \tfrac{T}{2}\boldsymbol{f}\big(\overline{\boldsymbol{x}}_{\mathrm{tr}}^{n+1}, \overline{\boldsymbol{u}}_{\mathrm{tr}}^{n+1}\big) = \overline{\boldsymbol{x}}_{\mathrm{md}}^{n+1}. \quad (35)$$

Finding the initial conditions is less complex in that case. The initial source sample $\overline{\boldsymbol{u}}_{\mathrm{md}}^{0}$ can be chosen arbitrarily. If the input function $u(t)$ is known for $t \geq 0$, a possible choice would be $\overline{\boldsymbol{u}}_{\mathrm{md}}^{0} = \boldsymbol{u}\big(\frac{T}{2}\big)$. An alternative option is $\overline{\boldsymbol{u}}_{\mathrm{md}}^{0} = \frac{1}{2}\big(\overline{\boldsymbol{u}}_{\mathrm{tr}}^{1} + \overline{\boldsymbol{u}}_{\mathrm{tr}}^{0}\big)$, which does not require explicit knowledge of $\boldsymbol{u}(t)$. The initial state $\overline{\boldsymbol{x}}_{\mathrm{md}}^{0}$ is obtained as

$$\overline{\boldsymbol{x}}_{\mathrm{md}}^{0} = \overline{\boldsymbol{x}}_{\mathrm{tr}}^{0} + \tfrac{T}{2}\boldsymbol{f}(\overline{\boldsymbol{x}}_{\mathrm{tr}}^{0}, \overline{\boldsymbol{u}}_{\mathrm{tr}}^{0}). \quad (36)$$

4.4. Considerations for typical audio systems

A large share of the audio systems presented in the literature are characterized by having the dynamical elements only be linear (e.g., capacitor, inductor), and having the nonlinear elements only be memoryless (e.g., diode, transistor, operational amplifier). Several simulation frameworks for audio systems (e.g., wave digital filters [9, 28], nodal DK method [29], generalized state space [8])

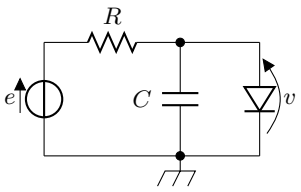


Figure 2: *Diode clipper circuit.*

segregate the system between sets of memoryless nonlinear equations and sets of linear equations between the system variables.

For the midpoint method, discretizing the linear equations is the same process as for the trapezoidal method, so the discrete update equations are identical. Memoryless nonlinear equations (e.g., voltage–current characteristics) are often of the form $\boldsymbol{b}(t) = \boldsymbol{h}(\boldsymbol{a}(t))$ (e.g., with $\boldsymbol{a}$ voltages and $\boldsymbol{b}$ currents for a voltage–current characteristic). On the contrary to the trapezoidal method where the update equations remain memoryless as $\overline{\boldsymbol{b}}^n = \boldsymbol{h}(\overline{\boldsymbol{a}}^n)$, the midpoint method equations become

$$\overline{\boldsymbol{b}}^{n+1} = -\overline{\boldsymbol{b}}^n + 2\boldsymbol{h}\big(\tfrac{1}{2}\big(\overline{\boldsymbol{a}}^{n+1} + \overline{\boldsymbol{a}}^n\big)\big). \tag{37}$$

This transformation would then be applied to the nonlinear elements at the root of a wave digital filter tree [9, 28], or to the nonlinear characteristic equations in the nodal DK approach [3] and in the generalized state-space [8].

Alternatively, we can use the process described in Sec. 4.2 to exploit simulations that use the trapezoidal method, applying the appropriate transformation to the input sequence (following Eq. (28)), and adding the conversion equations (following Eqs. (29) and (30)) to get the state variables for the midpoint simulation.

5. CASE STUDY

We study the diode clipper [14, 19, 20, 30] as shown in Fig. 2 to illustrate the concepts developed in the previous sections, as we study and compare the behavior of both methods in several scenarios and provide tools to understand and forecast such behavior.

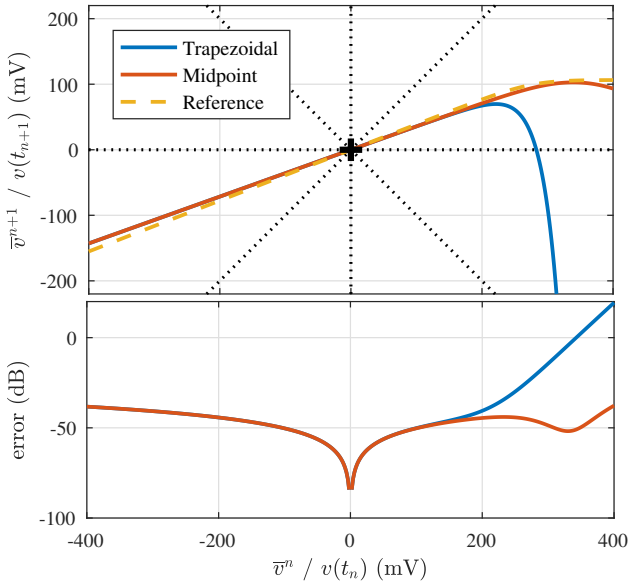
5.1. Circuit description

The circuit consists of a voltage input (i.e., a source), a resistor, a capacitor and a diode. The diode and capacitor are in parallel, their combination is in series with the resistor and the voltage source, and we wish to simulate the voltage v across the diode as a function of the source voltage e. The system behavior can then be summarized as the state-space system

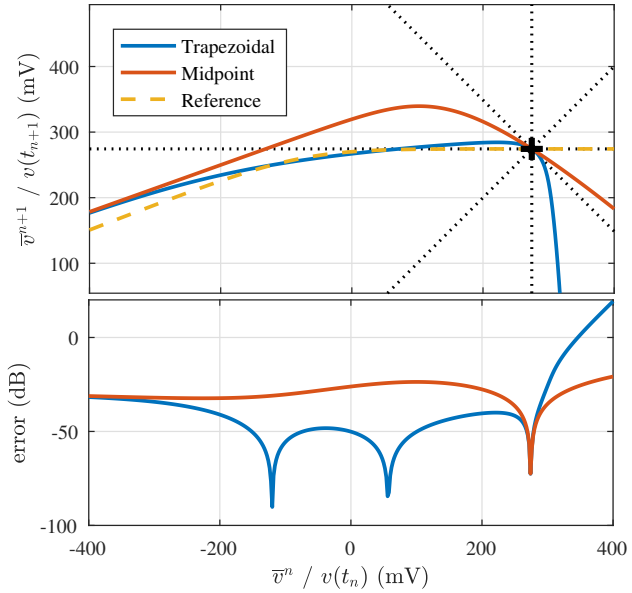
$$v_t(t) = \frac{e(t) - v(t)}{RC} - \frac{g(v(t))}{C}, \tag{38}$$

Table 1: *Simulation parameters.*

Name	Value	Description
f_s	48 kHz	sampling frequency
T	20.83 µs	sampling period
R	2.2 kΩ	resistor
C	10 nF	capacitor
I_S	2.52 nA	N914 saturation current
V_T	25.85 mV	thermal voltage



(a) $e = 0\,\mathrm{V}$



(b) $e = 0.5\,\mathrm{V}$

Figure 3: *$v(t_{n+1})/\overline{v}^{n+1}$ and error for as function of $v(t_n)/\overline{v}^n$ for two input conditions. The equilibria are indicated with a black +.*

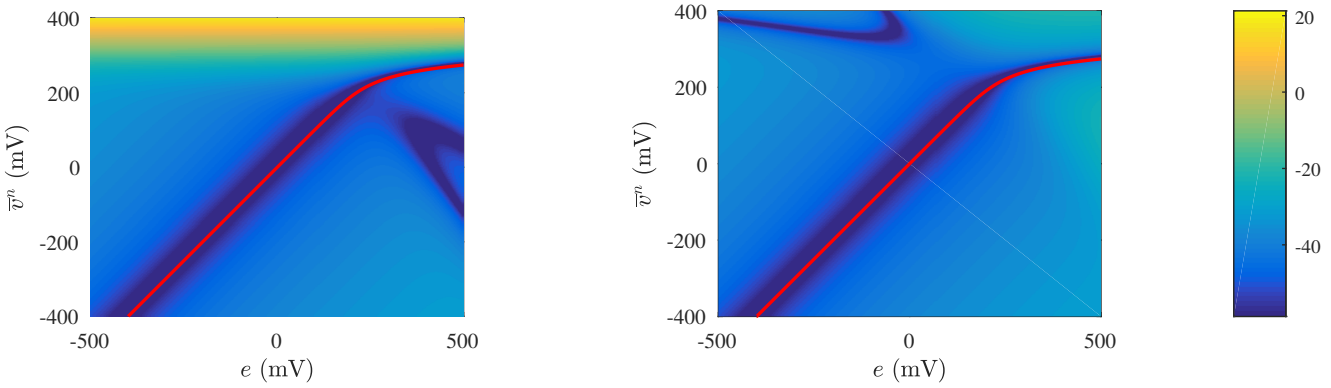
with g is the diode voltage–current characteristic. We then have:

$$\begin{aligned} v_t(t) &= f(v(t), e(t)), \quad \text{with} \\ f(x, u) &= (u - x)/RC - g(x)/C. \end{aligned} \tag{39}$$

We use the common Shockley diode model [31] for which

$$g(x) = I_S(e^{x/V_T} - 1), \tag{40}$$

with I_S the saturation current and V_T the thermal voltage.



(a) Trapezoidal method

(b) Midpoint method

Figure 4: *Error (in dB) for both methods. The equilibria are indicated in red.*

5.2. Simulation parameters

We adopt typical parameter values for fixed-rate virtual analog simulations as shown in Tab. 1. The response of the reference system is approximated using the MATLAB adaptive solver `ode45` set to work with a target error at machine precision. The equilibria and the implicit equations for both numerical methods are computed using the MATLAB root finder `fzero` set to work with a target error at machine precision.

5.3. Constant-input study

Many typical inputs are characterized by a constant-input regime. For example, an impulse response will lead to a constant zero input after the impulse. Similarly, a step response will lead to a constant non-zero input after the transient. In such a case, the system can be interpreted as an ODE (i.e., as a system without input), with the input voltage becoming a constant in the equation such that

$$v_t(t) = h(v(t)) = f(v(t), e(t)). \tag{41}$$

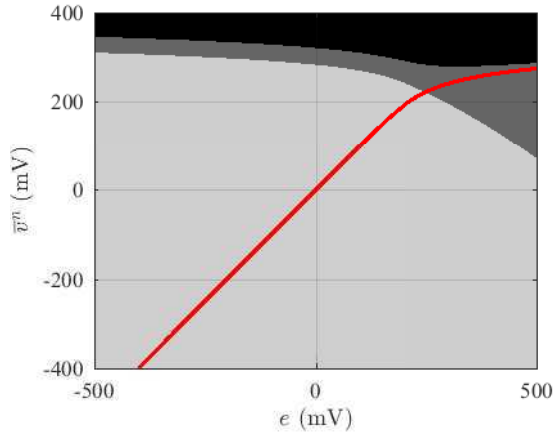
First, we investigate the error made by each method on a single simulation step, i.e. when computing the new state $\overline{v}^{n+1}$ (or $v(t_{n+1})$) from a current state $\overline{v}^n$ (or $v(t_n)$). Fig. 3 show the new state $\overline{v}^{n+1}$ and the error $|v(t_{n+1}) - \overline{v}^{n+1}|$ as a function of the current state $\overline{v}^n$ (or $v(t_n)$) for two constant voltage source values (respectively $e = 0$ and 0.5V). The plots show that while the two methods expectedly compare with one another and the reference for lower $v(t_n)$ values where the system behaves quasi-linearly, the error for the trapezoidal rule rises sharply for higher values (roughly $200\,\mathrm{mV}$ and higher). That region corresponds to the region where the system is highly non-linear and stiff. Those observations extend readily to other choices of input voltage and state values as shown in Fig. 4 where we display the error as a function of the current state $v(t_n)$ and the constant input value e.

5.4. Fixed-point analysis

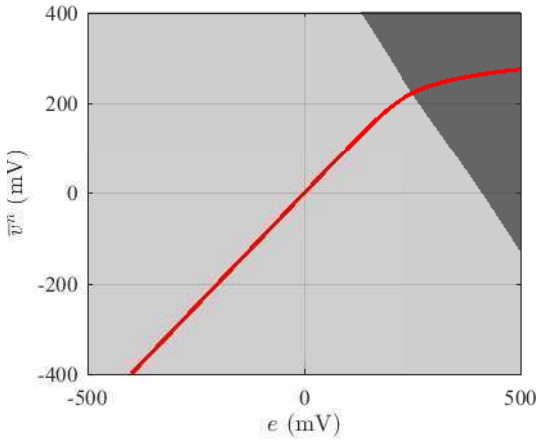
Using the analysis described in Secs. 3.5 and 3.6 for Fig. 1, we further understand the behavior for the two methods and the reference. In Fig. 3a, we see that for both methods and the reference, the first-order behavior around the system equilibrium is a non-oscillating decaying exponential. Furthermore, the transition functions are quasi-linear functions for a large region of state values around the equilibrium, where we have a quasi-exponential decay of the state sequence once a state value falls within that region. For high state voltages however, while the midpoint method stays close to the reference in the convergent non-oscillating region, the trapezoidal method drops sharply in the oscillating region. This means that a state sample in that region will iterate to a sample with a much lower voltage than the true solution (i.e., overshoot the true voltage) before entering the convergent non-oscillating regime.

In Fig. 3b, we see clearly that the first-order behavior near the equilibrium is substantially different for the reference on the one hand and the two methods on the other hand. The reference follows again a non-oscillating exponential decay in that region, while the two methods present an oscillating exponential decay as shown by the negative slope of the transition function at the equilibrium. Beyond that region, the behavior of the two methods are actually very different. The midpoint method shows a decaying oscillating behavior on a large section of state values both above and below the equilibrium and only match the non-oscillating exponential decay behavior of the reference solution for low state values ($\lesssim -100\,\mathrm{mV}$). On the other hand, the trapezoidal method presents a non-oscillating quasi-exponential decay close to the reference for almost all state values below the equilibrium, with small oscillation only close to the equilibrium. However, the behavior above the equilibrium is again characterized by a transition function dropping again sharply in the oscillating region.

Fig. 5 shows the behavior of both methods for a wide range of e and $\overline{v}^n$ values. Knowing that the reference is always converging non-oscillating (i.e., light gray), its behavior is matched by the trapezoidal method only if the state variable does not exceed a value with a similar order of magnitude across the tested e values. The method becomes however oscillating above those values, becoming even diverging oscillating for very high $\overline{v}^n$. The midpoint method is convergent non-oscillating over a wide range of values for e and $\overline{v}^n$ values. As hinted in Fig.3b, we also see that high values of $\overline{v}^n$, for which the trapezoidal method is oscillating, keep a converging non-oscillating behavior if the input value e is low. However, the midpoint method is oscillating for an increasing range of high $\overline{v}^n$ values as the input value e becomes high.



(a) Trapezoidal method



(b) Midpoint method

Figure 5: *Behavior regions for both methods (Light gray: convergent non-oscillating, dark gray: convergent oscillating, black: divergent oscillating). Equilibria are indicated in red.*

5.5. Step response

Finally, we look at the step response of the system for various initial states $v(0) = v^0$ and input values e in Fig. 6. The behavior matches the results from the previous section with all systems converging to the equilibrium v^e (with $f(v^e, e) = 0$). In particular, we can observe how, in the case of a low input voltage with a high initial state, the trapezoidal method systematically overshoots the true response, resulting in a significant simulation error. We also observe in Fig. 6d that for a high input voltage and a high initial state, both the trapezoidal method and the midpoint method oscillate significantly around the equilibrium. Finally, Fig. 6c shows how for a high input voltage and a low initial state, the oscillation amplitude for the midpoint is much greater. In general, we see that each method shows regions of that neither method is absolutely superior in all scenarios, so that the best method should rather be determined with a scenario-dependent approach. It also shows how some behaviors are shared among both methods and cannot be avoided, such as the oscillations in Fig. 6b and 6d.

6. CONCLUSION

In this paper, we presented a comparison of the implicit midpoint method with the more commonly used trapezoidal method as discretization methods for time-invariant audio lumped system simulation, with a specific focus on fixed-rate simulation. We presented some of the relevant theoretical similarities and differences between the two methods. We particularly focused on quantifying of the oscillatory behavior that both methods exhibit for stiff systems. We then discussed the practical implementation of the implicit midpoint method and how an implicit midpoint solution sequence could be computed from a trapezoidal-based implementation and vice versa. Finally, we compared the behavior of those two methods on a simple diode clipper system, drawing from our earlier theoretical analysis, in order to predict their behavior and identify cases of better performance for each method.

From a larger perspective, this paper is also meant as the preliminary of tools for the design and comparison of discretization methods for the specific purpose of fixed-rate simulation of audio lumped systems. Future work will focus on extending and improving those tools, as well as integrating into our analysis additional considerations relevant to the audio field, such as aliasing. Ultimately, we intend on applying those tools to a wide class of numerical methods to derive heuristics for the systematic design of accurate and efficient fixed-rate simulations.

Acknowledgment—The author wishes to thank Dr. Kurt J. Werner for his thoughtful comments and suggestions.

7. REFERENCES

[1] P. Moin, *Fundamentals of engineering numerical analysis*, Cambridge Univ. Press, New York, NY, 2010.

[2] S. Bilbao, *Numerical sound synthesis: Finite difference schemes and simulation in musical acoustics*, J. Wiley, Chichester, UK, 2009.

[3] D. T. Yeh, J. S. Abel, A. Vladimirescu, and J. O. Smith, "Numerical methods for simulation of guitar distortion circuits," *Comput. Music J.*, vol. 32, no. 2, pp. 23–42, 2008.

[4] A. Fettweis, "Wave digital filters: Theory and practice," *Proc. IEEE*, vol. 74, no. 2, pp. 270–327, 1986.

[5] G. Borin, G. De Poli, and D. Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," *IEEE Trans. Speech and Audio Process.*, vol. 8, no. 5, pp. 597–605, 2000.

[6] S. Bilbao, *Wave and scattering methods for numerical simulation*, John Wiley & Sons, 2004.

[7] G. De Sanctis and A. Sarti, "Virtual analog modeling in the wave-digital domain," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 715–27, 2010.

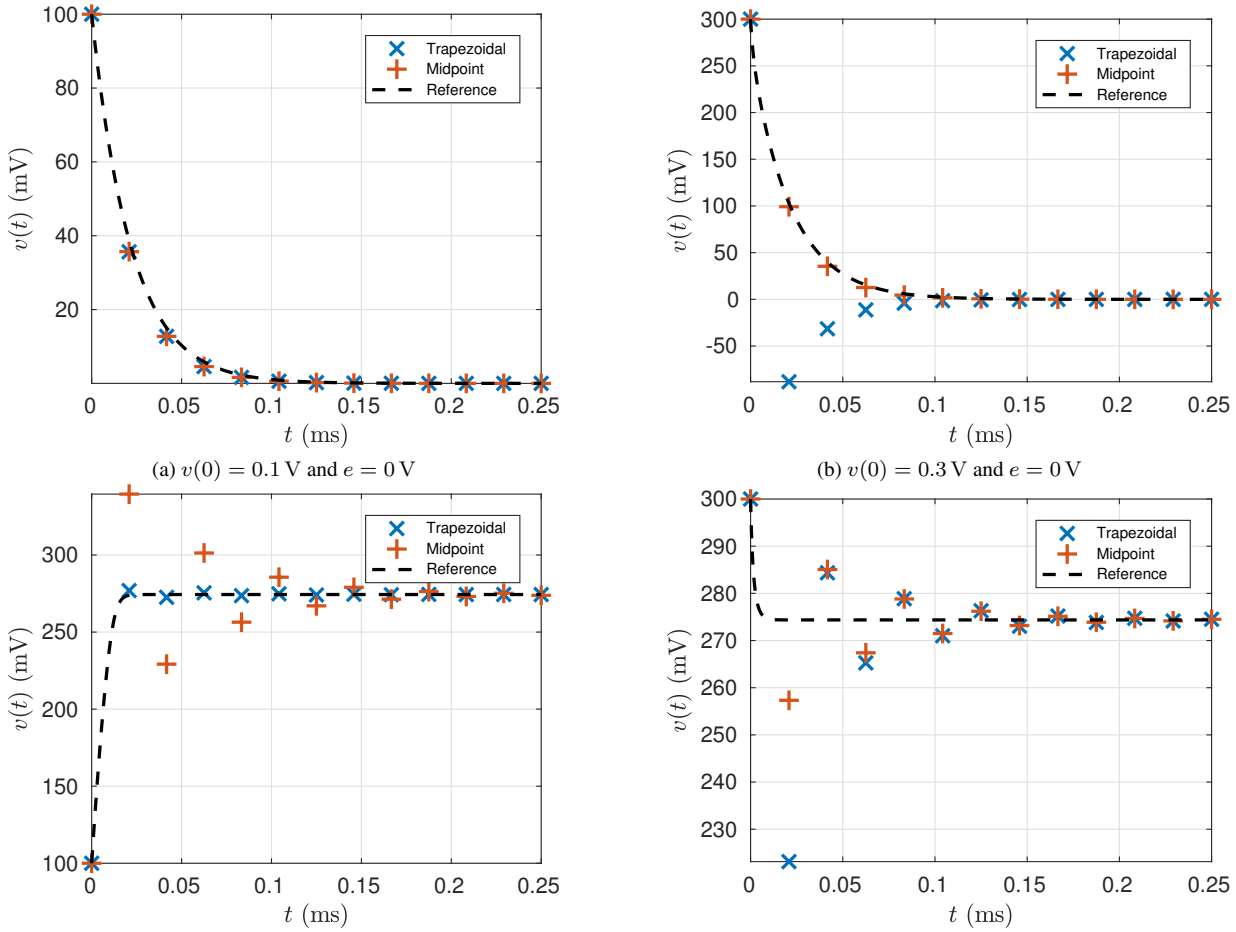
[8] M. Holters and U. Zölzer, "A generalized method for the derivation of non-linear state-space models from circuit schematics," in *Proc. 23rd European Signal Process. Conf.*, 2015.

[9] K. J. Werner, V. Nangia, J. O. Smith, and J. S. Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. 18th Int. Conf. Digital Audio Effects*, 2015.

[10] K. J. Werner, J. O. Smith, and J. S. Abel, "Wave digital filter adaptors for arbitrary topologies and multiport linear elements," in *Proc. 18th Int. Conf. Digital Audio Effects*, 2015.

[11] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, Springer, 2006.

[12] A. Falaize, *Modélisation, simulation, génération de code et correction de systèmes multi-physiques audios: Approche par réseau de composants et formulation Hamiltonienne à Ports*, Ph.D. diss., Université Pierre et Marie Curie, Paris, France, 2016.



(a) $v(0) = 0.1$ V and $e = 0$ V

(b) $v(0) = 0.3$ V and $e = 0$ V

(c) $v(0) = 0.1$ V and $e = 0.5$ V

(d) $v(0) = 0.3$ V and $e = 0.5$ V

Figure 6: *Step response simulation $\overline{x}^n$ and reference $x(t)$ for different initial conditions and constant input values.*

[13] A. Falaize and T. Hélie, "Passive guaranteed simulation of analog audio circuits: A port-Hamiltonian approach," *Appl. Sci.*, vol. 6, no. 10, 2016.

[14] F. G. Germain and K. J. Werner, "Design principles for lumped model discretisation using Möbius transforms," in *Proc. 18th Int. Conf. Digital Audio Effects*, 2015.

[15] D. T. Yeh, J. S. Abel, and J. O. Smith, "Simulation of the diode limiter in guitar distortion circuits by numerical solution of ordinary differential equations," in *Proc. 10th Int. Conf. Digital Audio Effects*, 2007.

[16] T. Stilson, *Efficiently-variable non-oversampled algorithms in virtual-analog music synthesis*, Ph.D. diss., Stanford Univ., 2006.

[17] F. G. Germain and K. J. Werner, "Joint parameter optimization of differentiated discretization schemes for audio circuits," in *Proc. 142 Conv. Audio Eng. Soc.*, 2017.

[18] F. G. Germain and K. J. Werner, "Optimizing differentiated discretization for audio circuits beyond driving point transfer functions," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2017.

[19] P. Daly, "A comparison of virtual analogue Moog VCF models," M.S. thesis, Univ. of Edinburgh, Edinburgh, UK, 2012.

[20] R. C. D. Paiva, S. D'Angelo, J. Pakarinen, and V. Välimäki, "Emulation of operational amplifiers and diodes in audio distortion circuits," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 59, no. 10, pp. 688–92, 2012.

[21] K. M. Hangos, J. Bokor, and G. Szederkényi, *Analysis and control of nonlinear process systems*, Springer, 2006.

[22] E. Süli and D. F. Mayers, *An introduction to numerical analysis*, Cambridge Univ. Press, 2003.

[23] G. G. Dahlquist, "A special stability problem for linear multistep methods," *BIT Numer. Math.*, vol. 3, no. 1, pp. 27–43, 1963.

[24] P. Howoritz and W. Hill, *The Art of Electronics*, Cambridge Univ. Press, 2nd edition, 1989.

[25] M. J. Olsen, K. J. Werner, and F. G. Germain, "Network variable preserving step-size control in wave digital filters," in *Proc. 20th Int. Conf. Digital Audio Effects*, 2017.

[26] E. A. Coayla-Teran, S.-E. A Mohammed, and P. R. C. Ruffino, "Hartman-Grobman theorems along hyperbolic stationary trajectories," *Discrete Contin. Dyn. Syst.*, vol. 17, no. 2, pp. 281–92, 2007.

[27] G. G. Dahlquist and B. Lindberg, "On some implicit one-step methods for stiff differential equations," Tech. Rep. TRITA-NA-73-02, The Royal Institute of Technology, Stockholm, Sweden, 1973.

[28] K. J. Werner, W. R. Dunkel, and F. G. Germain, "A computational model of the Hammond organ vibrato/chorus using wave digital filters," in *Proc. 19th Int. Conf. Digital Audio Effects*, 2016.

[29] D. T. Yeh, J. S. Abel, and J. O. Smith, "Automated physical modeling of nonlinear audio circuits for real-time audio effects—part I: Theoretical development," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 4, pp. 728–37, 2010.

[30] J. Macak and J. Schimmel, "Nonlinear circuit simulation using time-variant filter," in *Proc. 12th Int. Conf. Digital Audio Effects*, 2009.

[31] A. S. Sedra and K. C. Smith, *Microelectronic circuits*, New York: Oxford University Press, 1998.