SIMULATION OF THE DIODE LIMITER IN GUITAR DISTORTION CIRCUITS BY NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS

David T. Yeh, Jonathan Abel, and Julius O. Smith

Center for Computer Research in Music and Acoustics (CCRMA)
Stanford University, Stanford, CA
[dtyeh|abel|jos]@ccrma.stanford.edu

ABSTRACT

The diode clipper circuit with an embedded low-pass filter lies at the heart of both diode clipping “Distortion” and “Overdrive” or “Tube Screamer” effects pedals. An accurate simulation of this circuit requires the solution of a nonlinear ordinary differential equation (ODE). Numerical methods with stiff stability – Backward Euler, Trapezoidal Rule, and second-order Backward Difference Formula – allow the use of relatively low sampling rates at the cost of accuracy and aliasing. However, these methods require iteration at each time step to solve a nonlinear equation, and the tradeoff for this complexity must be evaluated against simple explicit methods such as Forward Euler and fourth order Runge-Kutta, which require very high sampling rates for stability. This paper surveys and compares the basic ODE solvers as they apply to simulating circuits for audio processing. These methods are compared to a static nonlinearity with a pre-filter. It is found that implicit or semi-implicit solvers are preferred and that the filter/static nonlinearity approximation is often perceptually adequate.

1. INTRODUCTION

Guitarists tend to feel that digital implementations of distortion effects sound inferior to the original analog gear. This work attempts to provide a more accurate simulation of guitar distortion and a physics-based method for designing the algorithm in the same manner physical modeling is done for acoustic instruments. Guitar effects consists of circuits that are accurately described in the audio frequency band by nonlinear ordinary differential equations (ODEs). Often the circuits are comprised of linear stages that can be efficiently implemented by infinite impulse response (IIR) digital filters. The remaining nonlinear ODEs may need to be solved by a numerical method or other approximation.

The diode clipper circuit with an embedded low-pass filter forms the basis of both diode clipping “Distortion” and “Overdrive” or “Tube Screamer” effects pedals[1]. An accurate simulation of this circuit requires the solution of a nonlinear ordinary differential equation (ODE). Numerical methods with stiff stability, Backward Euler, Trapezoidal Rule, and second order Backward Difference Formula (BDF, also called Gear) allow the use of low sampling rates at the cost of accuracy and aliasing[2]. However, these methods require iteration at each time step to solve a nonlinear equation, and the tradeoff for this complexity must be evaluated against simple explicit methods such as Forward Euler and fourth-order Runge-Kutta, which require very high sampling rates for stability.

1.1. Related work

A nonlinear system can be represented analytically as a Volterra series. There has been work to form finite-order Volterra series for simulating electronics [3, 4]. However, these are interesting only for low order, whereas for highly nonlinear systems, direct simulation by numerical methods is more computationally efficient [5].

Numerical solution of ODEs is a very mature topic in applied mathematics and many sophisticated algorithms exist for improving accuracy and speed [2, 6, 7, 8]. Backward Euler, Trapezoidal Rule, and BDF (called Gear in SPICE) are options to solve the nonlinear ODEs in circuits [9, 10]. Most algorithms are designed with a variable step size (sampling rate) to maintain a specified accuracy. The error is typically specified in the time-domain and is related to an order of the step size. Matlab features a rich suite of ODE solvers [11] that can be used for experimentation and gaining experience with the solution of ODEs.

Although not presented as such, an example of numerical simulation of ODEs for musical application is [12]. The WDF is an alternate implementation of the trapezoidal rule. It is equivalent to trapezoidal rule integration and results in the same iterations being solved because the nonlinearity is expressed in K-variables.[1]

1.2. Error criterion

Most applications for ODE solvers have a different set of requirements than those for audio. The error criterion for general solvers adaptively adjusts the step size to an excessively small value.

A fixed sampling rate is better suited for implementation in real-time audio processing. In addition, borrowing from the field of perceptual audio coding [13], the error specification for audio is best defined perceptually in the short-time frequency domain.

For audio, using a larger step size to reduce computational costs may increase aliasing, but this is tolerable if below the masking threshold of the desired audio signal. Also, the original analog electronics have a relatively high noise floor which would mask low level aliasing.

In this paper, the audio band is defined to be 0–20 kHz, where a match to the accurate solution of the ODE is desired. Subsonic frequencies are included because they may mix through the nonlinearity and cause perceptible amplitude modulation of the output. High frequencies are assumed to be sufficiently low due to roll-off of typical spectra that mixing products are negligible.

[1] The term “K-variable” means “Kirchoff variable,” such as a voltage or current. In contrast, WDFs use “W-variables” (“wave variables”). K-variables can be converted to W-variables and vice versa by means of a two-by-two matrix multiply as in (8).

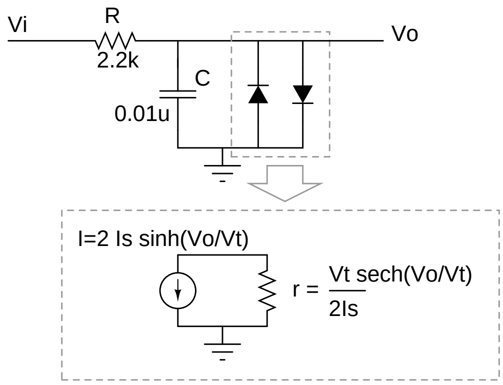


Figure 1: *Small signal approximation of diode clipper*

1.3. Diode clipper equation

Often nonlinearities for virtual analog modeling are assumed to be memoryless. The derivation below proves this to be an incorrect assumption, although cascading filters with a memoryless nonlinearity may be used as a perceptually close approximation. That the nonlinearity has memory is also suggested in [14], where the measurement technique to find the nonlinear transfer curves in a tube amp does not find a smooth nonlinearity, but rather a noisy one due to hysteresis.

The diode clipper in guitar circuits is typically a RC low-pass filter with a diode limiter across the capacitor (Fig. 1). The diode clipper limits the voltage excursion across the capacitor to about a diode drop in either direction about signal ground.

The model of the pn diode is

$$I_d = I_s(e^{V/V_t} - 1), \tag{1}$$

where the reverse saturation current I_s, and thermal voltage V_t of the device are model parameters that can be extracted from measurement.

The nonlinear ODE of the diode can be derived from Kirchhoff's laws:

$$\frac{\mathrm{d}V_o}{\mathrm{d}t} = \frac{V_i - V_o}{RC} - 2\frac{I_s}{C}\sinh(V_o/V_t), \tag{2}$$

where V_i, V_o are the input and output signals respectively.

This small-signal interpretation (Fig. 1) of this circuit contradicts the assumption of a memoryless nonlinearity because it yields a low-pass filter whose pole location changes with voltage.

2. NUMERICAL METHODS

2.1. Basic methods

The basic methods solve equations of the form

$$\frac{\mathrm{d}v}{\mathrm{d}t} = \dot{v} = f(t, v) \tag{3}$$

where the system state v is, in general, a vector, and $f(t, v)$ is a nonlinear function which encompasses any input $u(t)$ to the system. Time, t, is the independent variable of integration for ODEs that describe circuits.

In the case of a linear constant-coefficient differential equation, (3) can be written in state space form:

$$\dot{v}(t) = Av(t) + Bu(t) \tag{4}$$

where the eigenvalues of A are the poles of the system. Linear filters are efficient solvers of this special case of ODEs.

Explicit methods are those whose output depend only on state from previous time steps. Implicit formulas may depend on current state and require iteration if the ODE is nonlinear. Newton's method is the most popular solver, in part because it is scalable to higher dimensions. For single dimensional equations, bisection or bracketing provides predictable convergence [2].

In the literature for numerical methods, the methods are notated with subscripts denoting the time index and h for step size (sampling period). Here the methods are presented using square brackets to denote the time index and T for step size as for digital filters.

2.1.1. Forward Euler (FE)

$$v[n] = v[n-1] + T\dot{v}[n-1], \tag{5}$$

where $v[n]$ is the system state at discrete time n, and T is the sampling interval. This is an explicit, first-order method.

2.1.2. Backward Euler (BE)

$$v[n] = v[n-1] + T\dot{v}[n], \tag{6}$$

Note that this is similar to (5) except this is an implicit equation.

2.1.3. Trapezoidal Rule (TR)

$$v[n] = v[n-1] + \frac{T}{2}\left(\dot{v}[n] + \dot{v}[n-1]\right), \tag{7}$$

Trapezoidal rule is similar to the Euler methods, using instead the average of the derivatives at time n and $n-1$, resulting in a second-order method.

It is known that the trapezoidal rule has the smallest truncation error of any method of order two [10]. It is also equivalent to the discretization of a continuous-time transfer function by the bilinear transform. The trapezoidal rule is also the only practical order-preserving method that does not introduce artificial damping when discretizing continuous time systems.

2.1.4. Wave Digital Filter implementation of Trapezoidal Rule integration

The wave digital filter (WDF) is an alternate implementation of the trapezoidal rule integration where K-variables V, I (values corresponding to physical voltages and currents) are replaced by W-variables a, b by the mapping [12, 15]

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & R_0 \\ 1 & -R_0 \end{bmatrix} \begin{bmatrix} V \\ I \end{bmatrix} \tag{8}$$

Summarizing the approach in [15, 16], to make a nonlinear WDF, one may use (1), which is in the form $I = I(V)$ and substitute into (8) and obtain the wave variables:

$$a = f(V) = V + RI(V) \tag{9}$$

$$b = g(V) = V - RI(V) \tag{10}$$

Given an incident wave a and an invertible $f(V)$, one may find V and then use (10) to find b. Because $I(V)$ is the nonlinear function (1), this requires an iterative method. Note that the iteration involves the K-variable V. Therefore, the resulting iterations are identical to the direct application of the trapezoidal rule to the ODE.

The trapezoidal rule is implied in the use of the bilinear transform to locally discretize capacitors and inductors. The nonlinear WDF can thus be viewed as an alternative implementation of the trapezoidal rule solver for nonlinear ODEs.

The WDF poses the advantage that the nonlinear equations are solved locally for each nonlinear element, minimizing the size of the matrix equation to be inverted, even if the circuit contains many nodes or elements.

2.1.5. Backward Difference Formula

The Backward Difference Formula of order two (BDF2) is commonly used in circuit simulation and deserves mention here. It is a multi-step method that only requires a single function evaluation of (3).

$$v[n] = \frac{4}{3}v[n-1] - \frac{1}{3}v[n-2] + \frac{2T}{3}\dot{v}[n] \tag{11}$$

2.1.6. Runge-Kutta

A popular higher-order one-step method is the explicit fourth-order Runge-Kutta formula (RK4).

$$\begin{aligned} k_1 &= Tf(n-1, v[n-1]), \\ k_2 &= Tf(n-1/2, v[n-1] + k_1/2), \\ k_3 &= Tf(n-1/2, v[n-1] + k_2/2), \\ k_4 &= Tf(n, v[n-1] + k_3), \\ f(n,v) &= \dot{v}(n,v) = \frac{V_i[n]-v}{RC} - 2\frac{I_s}{C}\sinh(v/V_t), \\ v[n] &= v[n-1] + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} \end{aligned}$$

The function evaluations at times in between samples could be inconvenient for digital audio. The only time dependence in the diode clipper ODE is the input voltage. Therefore, RK4 requires input at twice the sampling rate of the output. On the contrary, a method should have a higher output rate than input rate because the nonlinearity causes the output to have a wider bandwidth than the input.

2.1.7. Semi-implicit methods

Because the Newton method converges to the solution quickly if the initial guess is close to the final result, often one iteration is sufficient for oversampled methods. This is the semi-implicit method[2], which has predictable cost.

Another way to save computation is to compute the Jacobian once and use it several times in the iterations (chord method)[11].

2.1.8. Approximation of ODE by static nonlinearity and digital filters

This is not an ODE method but approximates the result. It provides a baseline to evaluate the significance of the memory in the

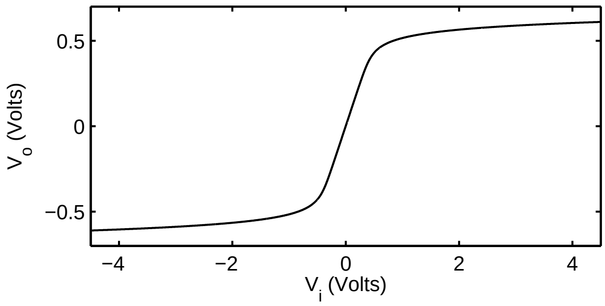


Figure 2: *Tabulated static nonlinear function*

nonlinearity. The comparison also demonstrates what is lacking when static nonlinearities are used.

The nonlinearity used is the DC approximation of the actual nonlinearity (Fig. 2), which can be derived from (2) by setting $C\frac{\mathrm{d}V}{\mathrm{d}t} = 0$. This is implemented using a lookup table as in [1] and is also known as waveshaping distortion. It is found that using a first-order low-pass filter before the nonlinearity with a cutoff frequency determined by the R and C of the diode limiter reduces aliasing while maintaining accurate output.

2.2. Order and Accuracy

The traditional measure of accuracy is Local Truncation Error (LTE), which is the lowest order difference between the Taylor series expansion of the solution and the result of the method. Other manifestations of error are aliasing and frequency warping. Oversampling reduces error by the order of the method. Because aliasing is more a function of the nonlinearity than of the method, it determines the minimum oversampling factor required. Frequency warping is not a concern in the audio band at these high sampling rates.

2.3. Stability

Consider the linear system described by (4) with $B = 0$. Substituting this into the Forward Euler method (5), for example, yields:

$$y_n = (I + TA)y_{n-1} + TBu_{n-1} \tag{12}$$

This is stable if $|1 + T\lambda| < 1$ for each eigenvalue λ of matrix A. The stability of an ODE method depends on the ratio between the sampling frequency and the largest eigenvalue of the system A.

2.3.1. Explicit methods

The plot of the region of stability on the $T - \lambda$ plane often forms a bounded region where the method is stable. Explicit methods result in polynomial stability conditions [6], which trace out a stability region that is a subset of the left half s-plane. Consequently, this places a limit on the largest magnitude negative eigenvalue the system may assume to assure bounded behavior.

2.3.2. Implicit methods

For implicit methods, the stability region extends to infinity in the negative half-plane, thereby placing no limit on the maximum

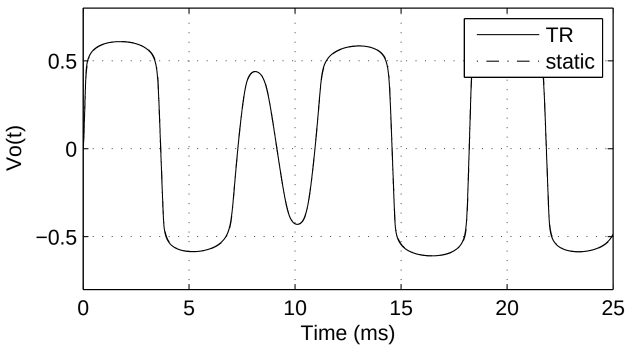


Figure 3: *Time-domain waveform for 110Hz + 155 Hz input, Trapezoidal (TR) and static approx., 8x oversampling. They are indistinguishable in the figure.*

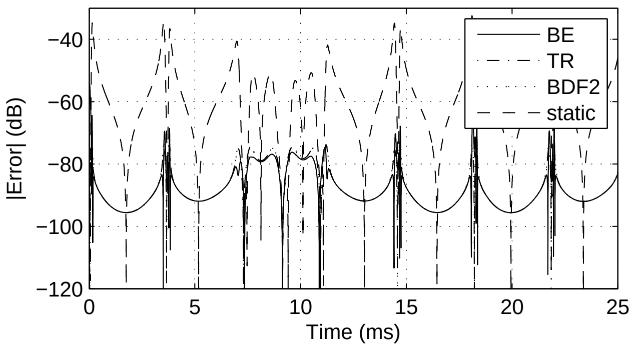


Figure 4: *Time-domain dB error for 110Hz + 155 Hz input, 8x oversampling. Backward Euler, Trapezoidal, BDF2, and static approx.*

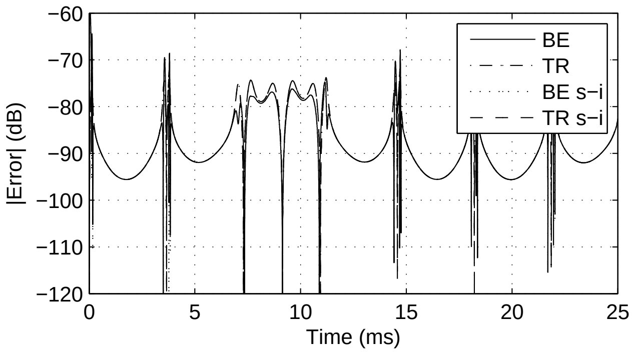


Figure 5: *Time-domain dB error for 110Hz + 155 Hz input, BE, TR and semi-implicit versions, 8x oversampling.*

magnitude of an eigenvalue of a system, if it is not complex, and allows a low sampling rate. For Trapezoidal, Backward Euler and BDF2, the regions encompass all of the left half-plane, so all stable systems will map to stable discrete time systems ("A-stability"). Backward Euler and BDF2 will introduce artificial damping to higher-frequency poles, while Trapezoidal Rule applies the bilinear transform to the poles and introduces no additional damping.

2.3.3. Stiff stability

For the ODEs found in circuits, it has been found in practice that implicit methods drastically reduce the sampling rate requirement relative to explicit methods, and are ultimately more efficient[10]. In the ODE literature this property is known as "stiffness." Stiffly stable solvers place no requirement on the minimum sampling rate needed to ensure a bounded solution. Instead, considerations for accuracy, in this case aliasing, govern the choice of step size.

All explicit methods are not stiffly stable [6] and they require a minimum sampling rate to operate properly. The left half plane eigenvalue of the diode clipper can be found from a small-signal linearization of the circuit. The linearized circuit is shown in Fig. 1. When V_o is large the linearized diode resistance will dominate over R, making this eigenvalue approximately

$$\lambda_{\mathrm{clip}} \approx -C\frac{V_t}{2I_s}\mathrm{sech}(V_o/V_t). \tag{13}$$

3. COMPARATIVE RESULTS AND DISCUSSION

3.1. Single-frequency sine

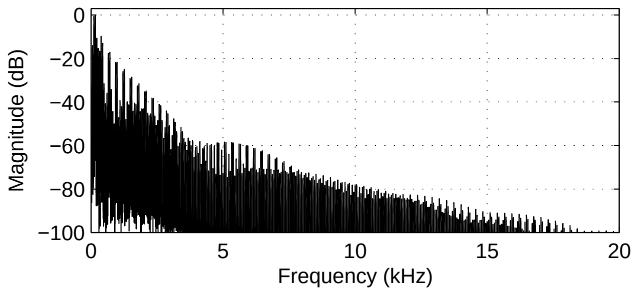
The implicit and semi-implicit methods at $8\times$ oversampling generate almost identical time-domain responses in response to a dual-tone excitation (110 and 155 Hz, 4.5 V peak), so only the Trapezoidal Rule (TR) and static approximation are shown in Fig. 3. Figs. 4-5 plot the error relative to an accurate solution computed with an oversampling factor of 32. All of the numerical methods exhibit similar error profiles with very low error. The approximation shows noticeably larger error than the numerical solvers, but it is typically less than -40 dB.

A spectral comparison better represents the audible differences. The numerical solvers all produce similar output spectra and are represented in Fig. 6 by the semi-implicit trapezoidal rule, which is very accurate at eight times oversampling as indicated by the time domain error. This is compared to the static approximation, which is a close approximation that reproduces most of the major peaks and contour of the spectrum.

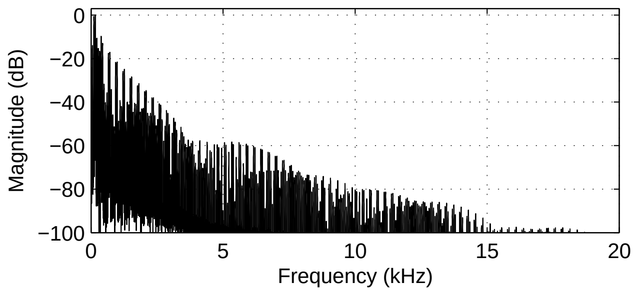
A high-level, high-frequency sine-wave excitation (4.5 V, 15001 Hz) reveals inadequacies in the semi-implicit methods, which exhibit overshoot in the time-domain plots (Fig. 7) and spurious tones in the frequency domain. The static nonlinearity produces phase error at high frequencies, although the magnitude of the fundamental is correct. The spectra of trapezoidal, semi-implicit trapezoidal, and static approximation are plotted in Fig. 8 with a reference spectrum generated by a trapezoidal rule with oversampling of 32.

3.2. Sine sweep

A high-amplitude, sinusoidal, exponential-frequency sweep from 0 to 20 kHz was processed by the methods. The oversampling factor of eight is chosen so the method is well-behaved throughout the test. The output is downsampled to 96 kHz and displayed as a log spectrogram [17]. All of the ODE methods produce almost identical output if stable, so only the spectrograms for trapezoidal rule, its semi-implicit version, and the static nonlinearity are shown in



(a) Semi-implicit trapezoidal



(b) Static nonlinearity approximation

Figure 6: *Spectra of responses for 110Hz + 155 Hz input.*

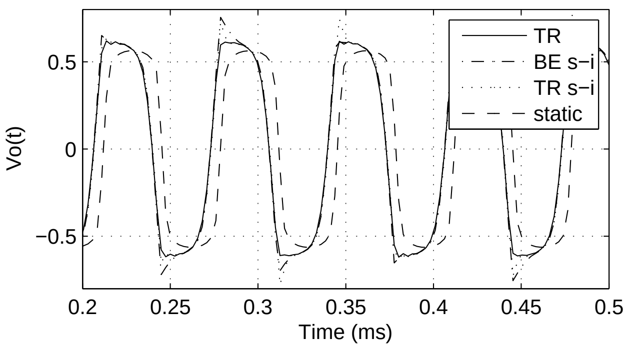


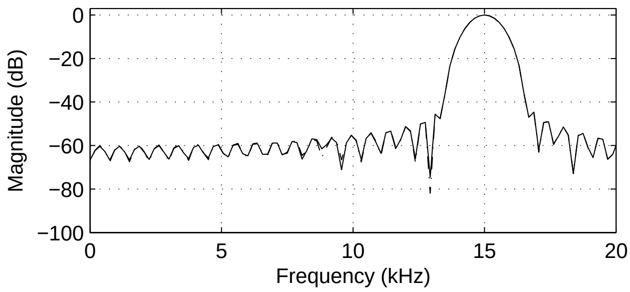
Figure 7: *Time domain waveform for 15001 Hz input, implicit and semi-implicit methods,* $8\times$ *oversampling.*

Fig. 9.

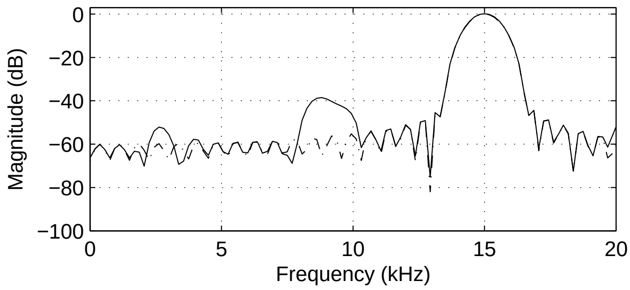
3.3. Computational cost

Because the number of iterations in an ODE solver that employs Newton's method is related to the input in a complicated way, an empirical measurement of cost is made. The number of iterations for several test signals is given in Table 1. The inputs used are an exponential sine sweep from 0 to 20 kHz, an E power chord on the low strings, and a riff with a bend.

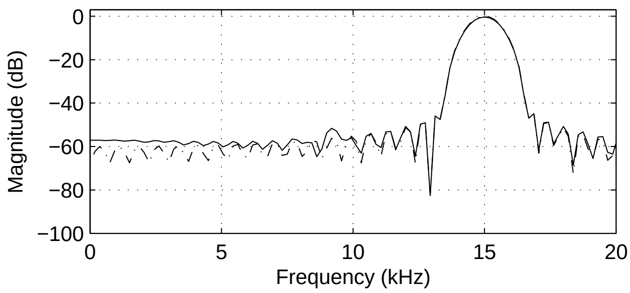
The cost per sample in terms of function calls to compute the derivative (3) or the Jacobian in the iterative methods is shown in Table 2. The cost of computing the derivative is assumed to be similar to the cost of computing the Jacobian. The cost is normalized per audio sample at the base sampling rate of 48 kHz. For iterative methods, the number of iterations n is assumed to be 1.2 as suggested by the actual guitar signals in Table 1.



(a) Solid: Trapezoidal, Dash-dot: reference



(b) Solid: Semi-implicit trapezoidal, Dash-dot: reference



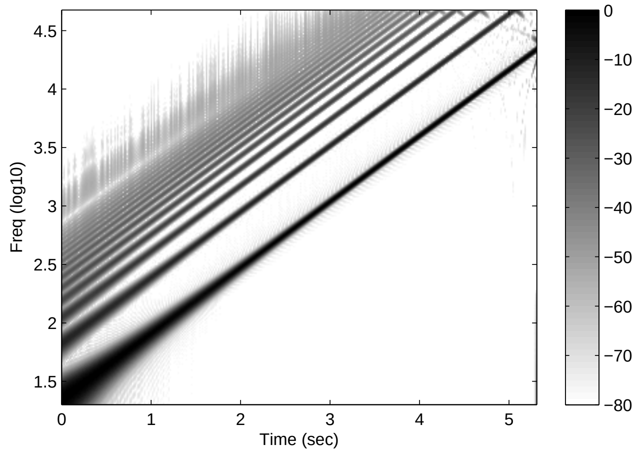
(c) Solid: Static approx., Dash-dot: reference

Figure 8: *Magnitude spectra of responses to 15001 Hz normalized to (32*$\times$*) oversampled reference.*

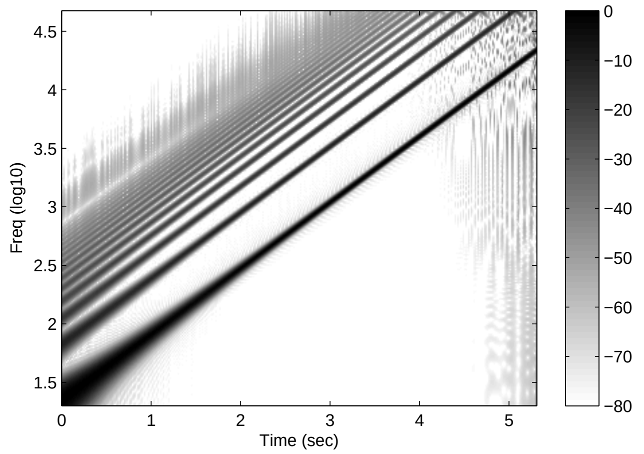
3.4. Discussion

For audio-frequency input, the differences between the methods are negligible in the audio band because the process is oversampled to reduce aliasing. The oversampling causes the various order errors to be very low in the audio band and makes the effect of frequency warping insignificant. It would seem then that a stable method of low order would be sufficient while guaranteeing bounded output.

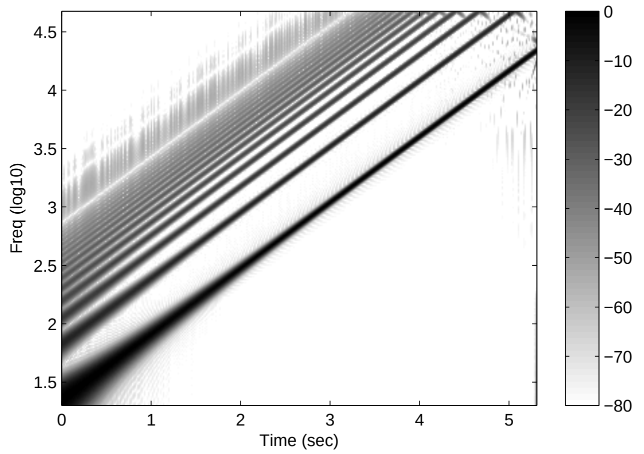
The time-domain outputs of the semi-implicit methods show significant ringing for high-frequency inputs, but this is an extreme case because high amplitudes at these frequencies are rarely encountered in practical guitar signals.



(a) Trapezoidal



(b) Semi-implicit trapezoidal



(c) Static approximation

Figure 9: *Log spectrogram of sine sweep, processed at* $8\times$ *over-sampling,* $f_s = 48$ *kHz, 80-dB dynamic range.*

Input	BE	BE s-i	TR	TR s-i	BDF2
sine sweep	1.381	1	1.370	1	1.367
power chord	1.035	1	1.035	1	1.035
riff	1.1821	1	1.1814	1	1.1815

Table 1: *Number of iterations for (semi-)implicit methods: Backward Euler, semi-implicit BE, Trapezoidal, semi-implicit TR, BDF2*

Method	X	f-calls	f-calls/sample
FE	38	1	38
RK4	30	4	120
BE	8	$2n$	19.2
BE s-i	8	2	16
TR	8	$1+2n$	27.2
TR s-i	8	3	24
BDF2	8	$2n$	19.2
static	8	lookup	-

Table 2: *Cost comparison of methods: Oversampling (X) required. Calls to derivative or Jacobian function. Calls per audio sample, $n = 1.2$ for iterative methods. n = # iterations; base sampling rate = 48 kHz*

The explicit methods, while simple, do not produce reliably accurate results in the frequency domain unless they are highly oversampled. The evaluation of cost validates prior findings in the circuit simulation literature that implicit or semi-implicit methods are preferred over explicit ones due to the large oversampling required for the explicit methods to be stable.

4. CONCLUSIONS

One important factor in deciding the sampling rate is the inevitable problem of bandwidth expansion caused by a nonlinearity, which may result in aliasing in sampled systems. Because the nonlinearities are strong, bandwidth is expanded by a large factor, necessitating large oversampling rates. This constraint on the sampling rate causes the different methods to be negligibly different in the audio frequency band. One can conclude that the simplest possible method that produces stable output should be used to save computational cost.

The amount of iterations taken by implicit methods depends on the frequency of the input signal. Quickly moving signals relative to the step size cause the initial estimate of the state to be far from the solution, requiring many iterations. Because the process is already highly oversampled relative to the bandwidth of realistic guitar signals to reduce aliasing, semi-implicit methods may work well. Future work could improve upon the convergence of the semi-implicit methods by limiting the overshoot for fast signals.

The static approximation of the ODE, which involves a prefilter and the DC transfer curve, is seen to be a cheap and effective approximation of the ODE, validating its widespread use in digital implementations of distortion effects. For future work, it is desirable to find other low-cost approximations of the ODE that result in better accuracy in the output spectrum.

At the very least, the application of ODE solvers to nonlinear musical effects builds insight into the problem and provides a reference point by which to evaluate more efficient approximations.

5. ACKNOWLEDGMENTS

David Yeh is supported by the Stanford, NDSEG and NSF graduate fellowships. Thanks to Andrei Vladimirescu for his advice and expertise on SPICE circuit simulation. Thanks to Stefan Bilbao for spending an afternoon and some commute time to work out ODEs and to explain numerical methods to me.

6. REFERENCES

[1] D. T. Yeh, J. Abel, and J. O. Smith, "Simplified, physically-informed models of distortion and overdrive guitar effects pedals," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-07)*, Bordeaux, France, Sept. 10–14, 2007.

[2] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, MA, second edition, 1992.

[3] T. Hèlie, "On the use of Volterra series for real-time simulations of weakly nonlinear analog audio device: application to the Moog ladder filter," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, Montreal, Quebec, Canada, Sept. 18–20, 2006, pp. 7–12.

[4] J. S. Abel and D. P. Berners, "A technique for nonlinear system measurement," in *Proc. 121st AES Convention*, San Francisco, CA, Oct. 2006.

[5] S. Bilbao, "Personal communication," Nov. 2006.

[6] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer-Verlag, New York, third edition, 2002.

[7] W. C. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[8] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman and Hall, New York, 1994.

[9] A. Vladimirescu, *The Spice Book*, Wiley, New York, 1994.

[10] W. J. McCalla, *Fundamentals of Computer-Aided Circuit Simulation*, Kluwer Academic Publishers, Boston, 1987.

[11] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE Suite," *SIAM Journal on Scientific Computing*, vol. 18, pp. 1–22, 1997.

[12] M. Karjalainen and J. Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *IEEE ICASSP 2006 Proc.*, Toulouse, France, 2006, vol. 5, pp. 153–156.

[13] M. Bosi and R. Goldberg, *Introduction to Digital Audio Coding and Standards*, Klewer Academic Publishers, 2003.

[14] S. Möller, M. Gromowski, and U. Zölzer, "A measurement technique for highly nonlinear transfer functions," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, Sep. 26-28 2002, pp. 203–206.

[15] K. Meerkötter and R. Scholz, "Digital simulation of nonlinear circuits by wave digital filter principles," in *Proc. IEEE Int Symp on Circuits and Systems*, May 1989, vol. 1, pp. 720–723.

[16] A. Sarti and G. De Poli, "Toward nonlinear wave digital filters," *IEEE Trans. Signal Process.*, vol. 47, pp. 1654–1668, June 1999.

[17] J. C. Brown, "Calculation of a constant-Q spectral transform," *J. Acoust. Soc. Am*, vol. 89, pp. 425–434, 1991, Matlab code available at http://web.media.mit.edu/ brown/cqtrans.htm.

[18] D. T. Yeh and J. O. Smith, "Discretization of the '59 Fender Bassman tone stack," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06)*, Montreal, Quebec, Canada, Sept. 18–20, 2006, pp. 1–6.

[19] A. Huovilainen, "Nonlinear Digital Implementation of the Moog Ladder Filter," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-04)*, Naples, Italy, Oct. 5–8, 2004.

[20] U. Zölzer, Ed., *DAFX – Digital Audio Effects*, J. Wiley & Sons, 2002.

[21] M. Karjalainen, T. Mäki-Patola, A. Kanerva, and A. Huovilainen, "Virtual air guitar," *Journal of the AES*, vol. 54, no. 10, pp. 964–980, Oct. 2006.