ALAMODE: AUTOMATED LEARNING OF ACOUSTICAL MODAL PARAMETERS VIA DIFFERENTIAL EVOLUTION

Jin Woo Lee, Jatin Chowdhury, Facundo Franchino

Music Technology
Massachusetts Institute of Technology
Cambridge, MA
{jnlee | chowdsp | facundof}
@mit.edu

Soohyun Kim

CCRMA
Stanford University
Stanford, CA
soohyun@ccrma.stanford.edu

Mark Rau

Music Technology
Massachusetts Institute of Technology
Cambridge, MA
mrau@mit.edu

ABSTRACT

This paper is a technical report on the methodology submitted for Task A of the 1st DAFx Parameter Estimation Challenge. The goal of the challenge's task is to invert the multi-dimensional physical and geometric parameters of a virtual plate reverberator given a target reference impulse response. To achieve this, we present a multi-stage gradient-free optimization framework. This three-stage optimization is computed using an efficient physics-based simulator, starting with an optimization of only mode frequency-determining physical parameters, followed by a 6-DoF parameter optimization with position-determining ones and a final phase for frequency- and position-independent mode amplitude estimation.

1. INTRODUCTION

Physical modeling audio synthesis allows for high-fidelity sound emulation by exploiting the underlying physical principles of the sonifying object [1–4]. Yet, an enduring challenge is the *inverse problem*: estimating the latent physical and spatial parameters of a synthesis engine from an arbitrary reference target recording [5,6].

The 1st DAFx Parameter Estimation Challenge Task A [7] presents a formal benchmarking environment for this inversion task using a virtual plate reverberator [8–10]. The latent search space contains parameters related to material properties, aspect ratios, tension, mass density, and pickup/excitation coordinates. Performing naïve numerical optimization in this setting is challenging and often fails to yield a practical solution [11, 12], because small changes in the physical parameters can move modal frequencies by several spectral bins [13]. Furthermore, conventional spectral error objectives produce highly multimodal loss surfaces with sharp, narrow basins of attraction, making the gradient-based optimization prone to local minima in such a non-convex problem [14].

This work introduces ALAMODE, an automated learning framework of acoustical modal parameters using differential evolution (DE) [15] method, which is gradient-free. Specifically, we split the parameter set into a frequency, position, and amplitude-determining subsets, and resolve each in turn using the DE method. In each phase, the optimal parameters obtained from the preceding phase are used as initial values; as a metric for optimality, we base the evaluation on the difference in spectral characteristics between the target impulse response (IR) and the output of the physical plate model [8] using JAX [16] similarly to [17].

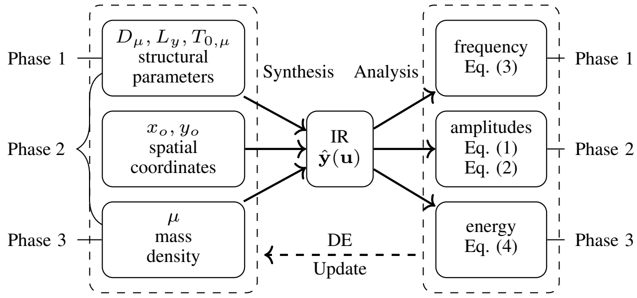


Figure 1: Schematic motivation for the proposed framework.

The physical motivation behind the multi-phase framework is summarized in Fig. 1. It has long been reported that, as the search space expands, the complexity of the function space increases, rendering the algorithm's inherent assumptions (bias) ineffective and making guarantees of global convergence exponentially difficult [18–21]. This has motivated the development of a multi-stage parameter estimation framework; we propose an approach that categorizes the physical parameters to be optimized based on the phenomena they influence and estimates them in separate stages. This can be interpreted as a form of physical modeling synthesis with analysis–synthesis system [22], adopting a strategy similar to the curriculum learning [23] but with a metaheuristic optimization algorithm [15]. We experimentally demonstrate that this facilitates fast and effective convergence.

2. OPTIMIZATION CRITERIA

Let $\mathbf{u} \in [0,1]^D = \mathcal{U}$ denote the normalized candidate parameter vector within a D-dimensional bounded unit hypercube. The forward model synthesizes an impulse response (IR) denoted as $\hat{\mathbf{y}}(\mathbf{u}) \in \mathbb{R}^{T_s}$, where $T_s = \lfloor f_s \cdot d \rfloor$ is the discrete duration at a sampling rate f_s. The target reference is represented by $\mathbf{y}_{\text{tgt}}$. The goal is to find the optimal normalized parameter vector $\tilde{\mathbf{u}} = \arg\min_{\mathbf{u}\in\mathcal{U}} \mathcal{L}(\mathbf{u})$ with $\mathcal{L}(\mathbf{u}) = \frac{1}{\sum_c w_c} \sum_c w_c \cdot \mathcal{L}_c\left(\hat{\mathbf{y}}(\mathbf{u}), \mathbf{y}_{\text{tgt}}\right)$, where $\mathcal{L}$ is defined by the composite objective function evaluating a normalized combination of specific spectral feature distances and w_c is the explicitly assigned scalar weighting factor for each metric component. As for the spectral feature distances $\mathcal{L}_c$, we employ three different criteria:

Band-Limited Smooth Log-Magnitude Loss. Let $\mathcal{M}(\cdot)$ denote the DFT magnitude operator. To prevent minor frequency misalignments from generating catastrophic gradient penalties, the spectrum is smoothed using a Hanning window $\mathbf{w}_{\text{med}}$ corresponding to a bandwidth $\Delta f = 10\,\text{Hz}$. For a frequency band delimited by bin boundaries $B = [b_{\text{lo}}, b_{\text{hi}}]$, the loss is formulated as:

$$\mathcal{L}_{\text{dft}} = \frac{1}{|B|}\sum_{k\in B}\left|\log\left(\mathcal{M}_{\text{sm}}(\hat{\mathbf{y}})_k\right) - \log\left(\mathcal{M}_{\text{sm}}(\mathbf{y}_{\text{tgt}})_k\right)\right|^2 \quad (1)$$

where $\mathcal{M}_{\text{sm}}(\mathbf{x}) = \mathcal{M}(\mathbf{x}) * \mathbf{w}_{\text{med}}$.

Spectral Optimal Transport (SOT) Loss. We compute a one-dimensional SOT loss based on the 1D Wasserstein metric [24]. Let $\bar{\mathcal{M}}(\mathbf{x})$ represent the normalized power spectral density over B, enforcing $\sum_{k\in B}\bar{\mathcal{M}}(\mathbf{x})_k = 1$. The corresponding Cumulative Distribution Function (CDF) at bin k is $\text{CDF}(\mathbf{x})_k = \sum_{m=b_{\text{lo}}}^{k}\bar{\mathcal{M}}(\mathbf{x})_m$. The SOT loss is defined as

$$\mathcal{L}_{\text{sot}} = \frac{1}{2}\sum_{k\in B}\left[\text{CDF}(\hat{\mathbf{y}})_k - \text{CDF}(\mathbf{y}_{\text{tgt}})_k\right]^2. \quad (2)$$

Parabolic Sub-Bin Modal Peak Extraction. We employ the distance in the mode frequencies via peak picking.

$$\mathcal{L}_{f_0 f_1} = \sqrt{|\hat{f}_0 - f_{0,\text{tgt}}|^2 + |\hat{f}_1 - f_{1,\text{tgt}}|^2} \quad (3)$$

RMS Scale Loss. A log distance in RMS values is employed:

$$\mathcal{L}_{\text{rms}} = \left|\log_{10}\text{RMS}(\hat{\mathbf{y}}) - \log_{10}\text{RMS}(\mathbf{y}_{\text{tgt}})\right|^2. \quad (4)$$

It is worth noting that when using DE, these losses need not to be differentiable, and we do not rely on gradient signals.

3. MULTI-PHASE INVERSION PIPELINE

The overall framework splits the parameters sequentially into three decoupled operational stages, as illustrated in Fig. 2. The parameter arrangement maps to a 6-DoF $\mathbf{u} = [D_\mu, L_y, T_{0,\mu}, \mu, x_o, y_o]^\top$.

3.1. Phase 1: Frequency-determining Parameters

The initial phase purely optimizes the macro-structural grid parameters that dictate pole positioning, ignoring position-dependent scaling factors. The free parameter space is $\mathbf{u}_3 = [D_\mu, L_y, T_{0,\mu}]^\top$. Non-frequency parameters are locked at their mid-points. Loss functions invariant to global amplitude, achieved through normalization by the overall size, were used. The primary objective is to achieve a rough alignment in the mode frequencies when reproduced by plate $\hat{\mathbf{y}}(\mathbf{u})$ to help a faster convergence in Phase 2.

3.2. Phase 2: Full Parameters with Position-determination

Once the underlying physical grid and tension boundaries are approximately locked, the search space opens up to jointly resolve spatial pickup arrangements and material properties. The target free space is full 6-DoF vector $\mathbf{u} \in [0,1]^6$. The top candidate from Phase 1 anchors the primary coordinates, while the remaining population matrix is drawn via stratified Latin Hypercube Sampling (LHS) [25]. Optimization is initially performed on the 1–200 Hz band without spectral smoothing. If no solution is found, a multi-resolution optimization sweep is conducted using a coarse-to-fine (c2f) scheduling strategy, ranging from a broad smoothing (40 Hz width) applied to a wider frequency band (up to 2 kHz) down to a case with no smoothing. Similar multi-resolution cascaded strategies have been demonstrated in many prior studies [26–28]. The primary objective of Phase 2 is to ensure that, after the parameters are used to regenerate the IR, both the modal frequencies and the relative amplitude distributions of the modes match the target.

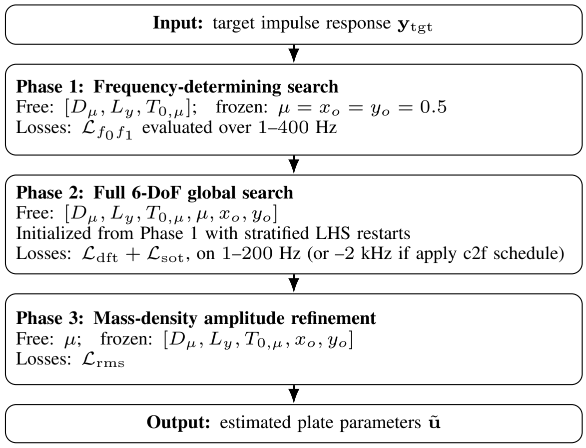


Figure 2: Overview of the proposed multi-stage pipeline.

3.3. Phase 3: Frequency-/Position-independent Parameter

Because surface mass density (μ) acts as an absolute scaling factor for time-integration amplitudes, standard spectral shapes can sometimes exhibit minor tracking inaccuracies. Therefore, the target free parameter space is a 1-DoF optimization isolating $\mu \in [0,1]$, with the remaining 5 parameters frozen firmly to the optimal vector from Phase 2.

4. EVALUATION

This section presents the evaluation results of the proposed method on both the synthetic IR dataset and the DAFx Parameter Estimation Challenge Task A test set. The results are compared with the baseline method provided in the challenge, which uses a Particle Swarm Optimization (PSO) algorithm [29]. DE adopted in our method is similar to PSO in that both are population-based, but DE uses random sampling to algebraically mutate candidate solutions. Details of the DE algorithm and hyperparameter settings are provided in Table 1. The code is implemented in Python; using JAX for the simulator and SciPy [30] for the DE.

4.1. Dataset

We evaluate ALAMODE on both a synthetic IR dataset and the challenge test set. The synthetic IR dataset consists of 128 randomly generated IRs using the virtual plate reverberator model provided in the official challenge codebase [1]. The parameters are sampled uniformly within the specified ranges for each parameter. The synthetic IRs are generated at a sampling rate of 48 kHz and a duration of 1 second. The challenge test set consists of 16 number of 5 seconds IRs with unknown ground-truth (GT) parameters.

4.2. Evaluation Metrics

The following evaluation metrics are used: (1) Normalized Mean Squared Error (NMSE): Measures the average squared difference

[1] https://github.com/LOGUNIVPM/1st-DAFx-Challenge

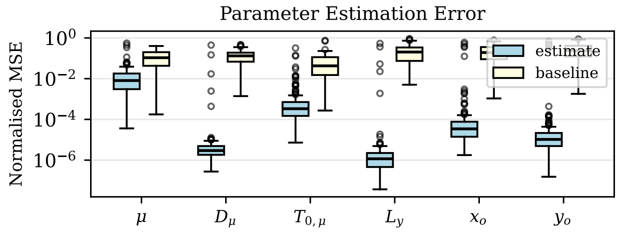
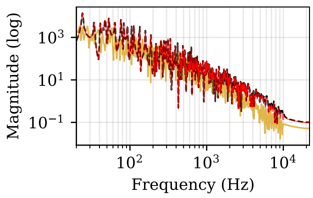
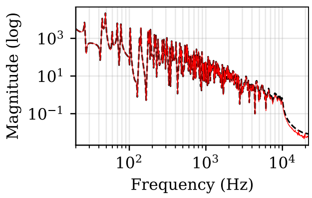


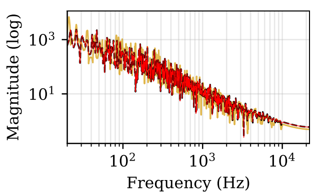
Figure 3: Boxplots of the normalized mean squared error (NMSE) for each parameter across the synthetic IR dataset. The boxes represent the interquartile range (IQR), with the median indicated by a black horizontal line. Whiskers extend to 1.5 times the IQR and outliers are shown as circles.



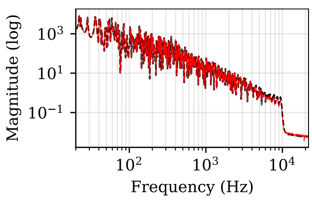
(a) Synthetic Data Sample 1



(b) Challenge Test Sample `0013`



(c) Synthetic Data Sample 2



(d) Challenge Test Sample `0009`

Figure 4: Log-magnitude spectra comparison between the target IR (black dashed line - - -) and the synthesized IR using ALAMODE (red solid line ——). For the synthetic dataset, the baseline IRs are shown in yellow solid lines (——) for reference.

between the estimated and ground-truth parameters, normalized by the range of each parameter provided in [7]. (2) Spectral Distance Error (SDE): Computes the log-magnitude spectral distance of the target IR and the IR resynthesized from the estimated parameters.

4.3. Evaluation Results

Fig. 3 shows the boxplots of the normalized mean squared error (NMSE) for each parameter across the synthetic IR dataset. The results indicate that ALAMODE achieves low NMSE values for most parameters, outperforming the baseline method. Regarding μ, which shows the most marginal performance improvement among the parameters, we have observed that increasing the population size of Phase 3 (at the cost of some additional time) could further enhance the performance; however, we settle for a satisfactory level of improvement considering the overall efficiency.

Close spectral match between target and resynthesized IRs. Looking at Fig. 4, the estimated spectra by ALAMODE do not deviate from the target spectra line within the audible frequency range. Considering this level of similarity as a *nearly perfect* reconstruction, ALAMODE shows nearly perfect performance on:

- 96% of the synthetic validation sets (123/128), and
- 100% of the challenge test sets (16/16).

Fig. 4a and Fig. 4b show the results with an average number of function evaluations, while Fig. 4c and Fig. 4d show the results with the highest number of function evaluations with multiple restarts on Phase 2 (see Fig. 5 for the corresponding parameter trajectories and the number of function evaluations). To speak in terms of quantitative scores, the average SDE is approximately 0.05 for the synthetic dataset, and 0.09 for the challenge test set. These SDE serve merely as a reference, as the ultimate goal of Task A is parameter estimation, making parameter accuracy the critical factor. However, these can be a key indicator in cases where the ground-truth parameters are unknown. We have not observed instances where a high SDE coincided with small parameter NMSE, nor the high NMSE coinciding with a sufficiently low SDE. The five synthetic data samples for which the spectra have not been properly reconstructed appear to be marked as outliers in Fig. 3 with the avearge NMSE values exceeding 0.1.

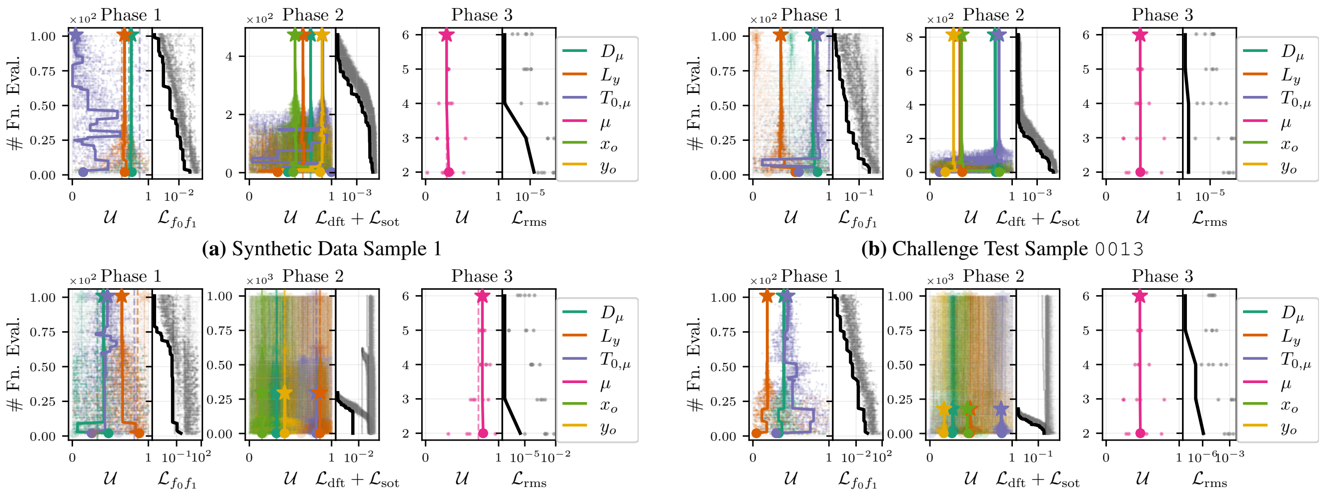
Multi-staged optimization facilitates effective convergence. We have found that, rather than optimizing all six parameters simultaneously, fitting a subset of them first and then fitting the remainder based on that facilitates faster convergence. As previously described, the frequency-determining parameter values approximated in Phase 1 serve as the initial conditions for Phase 2. In Fig. 5a, we can see that D_μ and L_y are identified close to the GT using only Phase 1; while $T_{0,\mu}$ does not converge well in Phase 1, but successfully converged toward the GT in Phase 2. While it is natural to question if Phase 2 alone (skipping Phase 1) would suffice, but, Fig. 5c illustrates a counterexample. In this sample, convergence failed during Phase 1, and the parameters ultimately selected in Phase 1 differed from the GT (indicated by the dashed line). Under such circumstances, optimization in Phase 2 becomes extremely difficult. Although convergence is eventually achieved in the case shown in Fig. 5c, it requires two restarts with the redefinition of initial conditions to reach that point.

Restarting with c2f scheduling can lead to fast convergence. As shown in Fig. 5c and Fig. 5d, the proposed method restarts with a new population when the loss does not improve for a certain number of iterations. In contrast to the Phase 2 of Fig. 5a and Fig. 5b, which converges within a few hundred function evaluations, the cases in Fig. 5c and Fig. 5d fails to converge even after a thousand evaluations; very rapid and successful convergence is achieved after 2–3 number of restarts. For the multiple restarts in Phase 2, we assume that the global optimum lies sparsely within a sharp loss landscape and employ the c2f scheduling strategy described in subsection 3.2. Applying spectral smoothing with progressively finer widths at each restart enables faster successful convergence; in practice, this manifests as a loss curve that becomes steeper with each restart. Yet, we design ALAMODE to first attempt without spectral smoothing, and indeed, we have observed successful convergence in many cases: Within the entire Task A test set, restarts occur just in two cases (Fig. 5b and Fig. 5d).

5. RUNNING TIME AND COMPUTATIONAL COST

ALAMODE iterates until the loss function satisfies the convergence criterion of each phase. Therefore, the total number of iterations and the time required vary depending on the convergence difficulty of each IR. The time taken to process all 16 IRs was 50 minutes and 33 seconds. Details regarding the duration and number of repetitions for each phase are summarized in Table 1 and Fig. 6.

Phase 2 is the computational bottleneck as is the core stage. While Phase 2 involves optimizing six parameters compared to three in Phase 1, effectively doubling the parameter dimension, both the number of function evaluations and the wall time increased



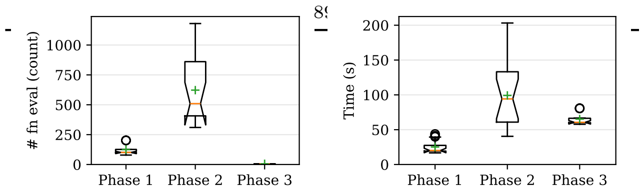
(a) Synthetic Data Sample 1 **(b)** Challenge Test Sample `0013`

(c) Synthetic Data Sample 2 **(d)** Challenge Test Sample `0009`

Figure 5: Parameter estimation trajectories. Each plot shows the evolution of parameters across the three optimization phases, where the vertical axis represents the function evaluation counts. For each phase, the normalized parameters of $\mathcal{U}$ are shown in colors and the loss values are displayed in black solid lines. The ground truth parameter is marked with a vertical dashed line if available. Trajectories for the best parameter vector, chosen based on the lowest loss value, are highlighted in solid lines with markers (●: initial, ⋆: final value.) Trajectories that were restarted after failing to converge, are also marked but with reduced opacity.

Table 1: Computational Cost Breakdown per Phase

Phase	# Function Evaluations Avg.	Min-Max	Wall Time (s) Avg.	Total	Popuplation Size	Max Iter.
1	125	79-202	25.1	402	10	100
2	623	310-1180	99.2	1587	15	1000
3	6	6-6	65.2	1043	5	5



(a) Function evaluation count per IR **(b)** Wall time per IR

Figure 6: Boxplots of the computational costs. Median values are indicated by orange lines and mean values by green plus signs.

fourfold. This exponential increase in time falls within a somewhat predictable range due to the 'curse of dimensionality' [19]. Given that Phase 2 is the core of the entire optimization process, this duration is considered manageable.

Gradient descent hybrid has not shown significant gain. By leveraging JAX's automatic differentiation capabilities, it is possible to perform the back-propagation through the simulator after each phase of the DE, blending with the gradient descent family such as L-BFGS algorithm [31] on a GPU. However, this has not been adopted because the performance has not been significantly superior, and the DE-only approach already shows low reconstruction error. Therefore, ALAMODE has become purely gradient-free.

Faster simulator is the key to further improving speed and performance. As observed from Fig. 6, Phase 3 takes nearly double wall time than Phase 1, despite having 20 times fewer function evaluations. This is because the plate model used to synthesize the IR from the parameters optimized in Phase 3 differs from the one used in Phases 1 and 2. The synthesizer used in Phases 1 and 2 is a simplified model that truncates the modes to a maximum of 256 for the sake of efficiency, therefore we have observed discrepancies in the IRs' RMS values even when using the same μ value with the ground truth. Consequently, we employ an explicit time integration for synthesizing IR in Phase 3, which necessitates allocating a longer duration to this phase. Although increasing the population size for Phase 3 and investing more time would have improved μ estimation performance, we have chosen to sacrifice some performance in Phase 3 to strike an optimal balance between efficiency and efficacy trade-off, and adhere to the challenge's running time requirements. It appears to be a viable future direction for further enhancing μ accuracy within the same timeframe by developing a faster simulator for Phase 3.

6. FUTURE WORK

This paper serves as a technical report for the challenge, leaving room for further academic investigation. Future works could explore generalizing the plate reverberation model to accommodate diverse shapes or material properties or testing generalization against measured real-world data. Implementing more efficient simulator is crucial for such endeavors, and achieving robust generalization may require precise physical modeling, such as accounting for the measurement noise and/or material anisotropy. Furthermore, while fully disentangled physical parameters are rare in reality, identifying which subset of parameters can be considered sufficiently disentangled (and be separated for the multi-stage framework) remains an open question.

7. SUMMARY

This paper presented a multi-stage DE framework for inverting the physical parameters of a virtual plate reverberator from a target IR. By decoupling the parameter set along physical axes of mode frequency-determining, mode amplitude-determining, and global energy-determining subsets, the pipeline avoids the local minima that trap joint optimization over the full 6-DoF space. The use of a JAX-based simulation model makes the approach computationally manageable, compatible with the challenge's test case scenario.

8. REFERENCES

[1] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual analog effects," *DAFX: Digital Audio Effects*, pp. 473–522, 2011.

[2] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I," *Journal of the Audio Engineering Society*, vol. 19, 1971.

[3] J. O. Smith III, *Physical Audio Signal Processing*, online book, 2010 edition, https://ccrma.stanford.edu/~jos/pasp/.

[4] S. Bilbao, *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*. John Wiley & Sons, 2009.

[5] J. O. Smith III, *Techniques for digital filter design and system identification, with application to the violin*. Stanford University, 1983.

[6] H. Han, V. Lostanlen, and M. Lagrange, "Learning to solve inverse problems for perceptual sound matching," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2605–2615, 2024.

[7] M. Ducceschi and L. Gabrielli, "1st DAFx parameter estimation challenge: Task and evaluation details," *27th Int. Conf. on Digital Audio Effects (DAFx26)*, 2026.

[8] S. Bilbao, K. Arcas, and A. Chaigne, "A physical model for plate reverberation," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5. IEEE, 2006.

[9] K. F. Graff, *Wave motion in elastic solids*. Courier Corporation, 2012.

[10] R. Szilard and W. Nash, "Theory and analysis of plates, classical and numberical methods," 1974.

[11] J. W. Lee, J. Park, M. J. Choi, and K. Lee, "Differentiable modal synthesis for physical modeling of planar string sound and motion simulation," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 1058–1081.

[12] B. Hayes, J. Shier, G. Fazekas, A. McPherson, and C. Saitis, "A review of differentiable digital signal processing for music and speech synthesis," *Frontiers in Signal Processing*, vol. 3, p. 1284100, 2024.

[13] J. Chowdhury, E. K. Canfield-Dafilou, and M. Rau, "Water bottle synthesis with modal signal processing," in *Int. Conf. Digital Audio Effects (DAFx)*, vol. 5, 2020.

[14] D. L. Sun and J. O. Smith III, "Estimating a signal from a magnitude spectrogram via convex optimization," *Proceedings of the Audio Engineering Society 133rd Convention*, 2012.

[15] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, 1997.

[16] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, Y. Katariya, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/jax-ml/jax

[17] R. Diaz Fernandez, M. Sandler *et al.*, "Fast differentiable modal simulation of non-linear strings, membranes, and plates." International Conference on Digital Audio Effects (DAFx25), 2025.

[18] D. H. Wolpert, W. G. Macready *et al.*, "No free lunch theorems for search," Technical Report SFI-TR-95-02-010, Santa Fe Institute, Tech. Rep., 1995.

[19] M. Verleysen and D. François, "The curse of dimensionality in data mining and time series prediction," in *International work-conference on artificial neural networks*. Springer, 2005, pp. 758–770.

[20] W. C. Rheinboldt, "A unified convergence theory for a class of iterative processes," *SIAM Journal on Numerical Analysis*, vol. 5, no. 1, pp. 42–63, 1968.

[21] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *Ussr computational mathematics and mathematical physics*, vol. 4, no. 5, pp. 1–17, 1964.

[22] X. Serra and J. Smith, "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition," *Computer Music Journal*, vol. 14, no. 4, pp. 12–24, 1990.

[23] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.

[24] B. Torres, G. Peeters, and G. Richard, "Unsupervised harmonic parameter estimation using differentiable dsp and spectral optimal transport," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1176–1180.

[25] M. Stein, "Large sample properties of simulations using latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[26] L. Gabrielli, S. Tomassetti, S. Squartini, C. Zinato *et al.*, "Introducing deep machine learning for parameter estimation in physical modelling," in *Proceedings of the 20th international conference on digital audio effects*, 2017.

[27] J. R. Beltrán, "Additive synthesis based on the continuous wavelet transform: A sinusoidal plus transient model," *DAFX03*, 2003.

[28] S. Hainsworth and M. Macleod, "On sinusoidal parameter estimation," in *Proc. DAFx Conference*, 2003, pp. 151–156.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. ieee, 1995, pp. 1942–1948.

[30] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[31] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical programming*, vol. 45, no. 1, pp. 503–528, 1989.