Highly Accurate Table Representations using Chebyshev samples with Barycentric Lagrange Interpolant

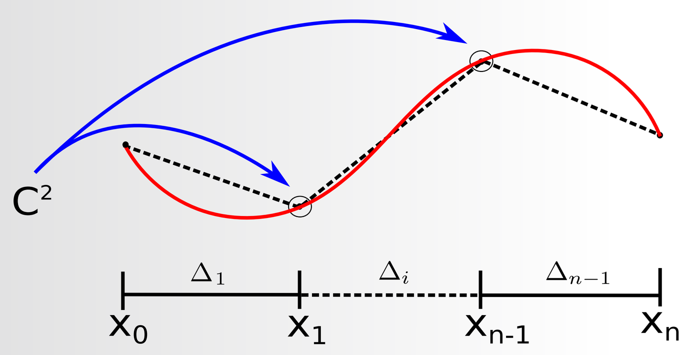
Archit Gupta
University Of California, Berkeley

Motivation

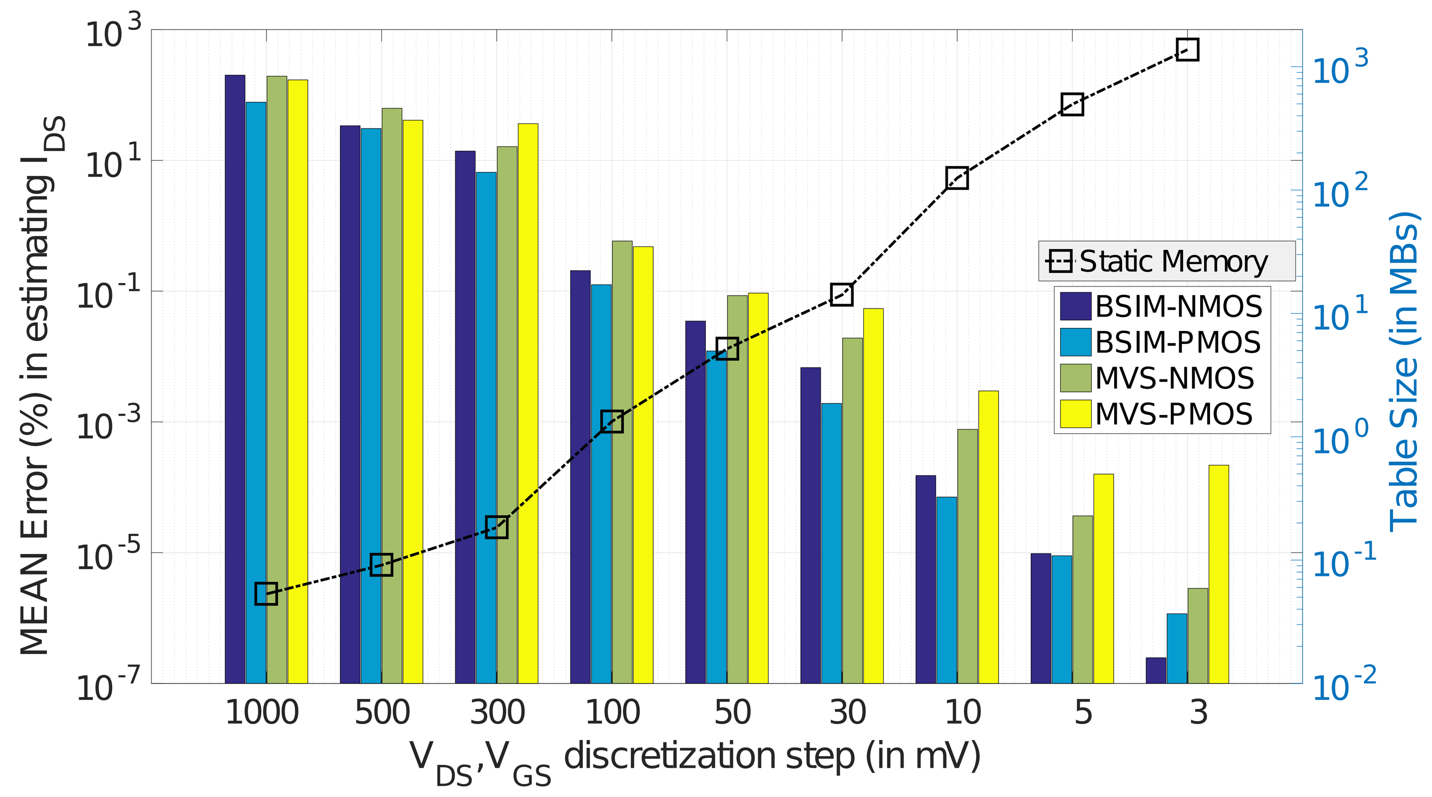
In previous work, titled "STEAM: Spline-based Tables for Efficient and Accurate device Modeling", we developed a table-modeling framework based on MODSPEC.

$$q_e(x), f_e(x), q_i(x), f_i(x)$$

Key Idea: 4 functions completely describe a device model. An approximation of the functions is an approximation for the device model itself.



Previous method used low order spline interpolants for accurate and fast interpolation. The memory requirements for any degree of accuracy are shown.



Can we do significantly better?
Say floating point precision!

Barycentric Lagrange Interpolation

$$p(x) = \frac{\sum_i \frac{f_i w_i}{(x - x_i)}}{\sum_j \frac{w_j}{(x - x_j)}}$$

BLI's performance depends largely on the choice of the set of sample points

It is known to be very stable if Chebyshev points are used as the sample points (Available for use in a package called CHEBFUN)

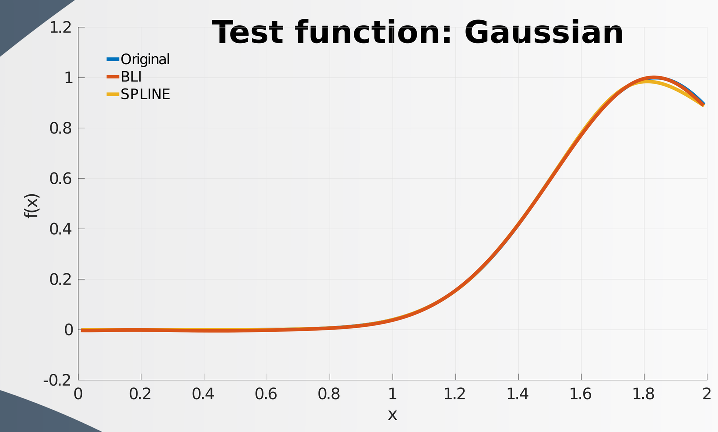
$$w_i = (-1)^i \binom{n}{i}$$

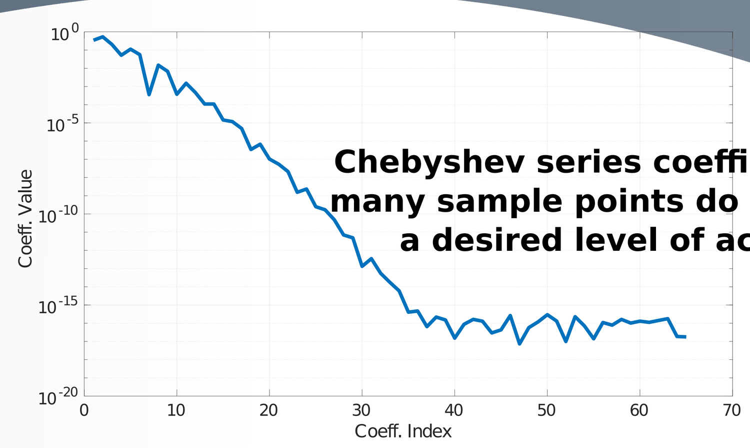
for uniform points, whereas for Chebyshev Points:

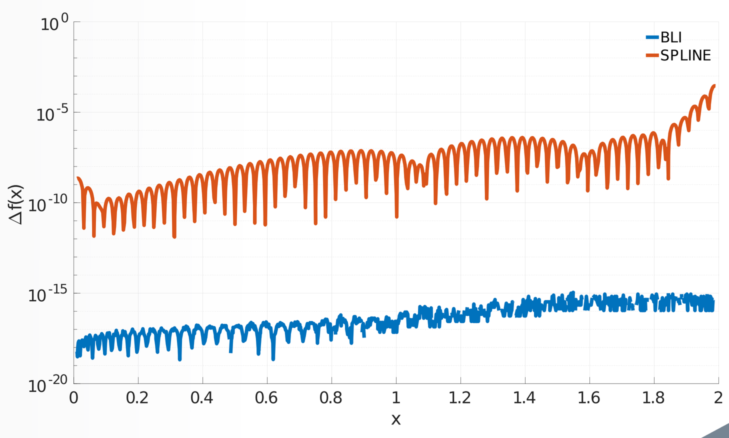
$$w_i = (-1)^i \delta_i$$

$$\delta_1, \ldots, \delta_{n-1} = 1$$

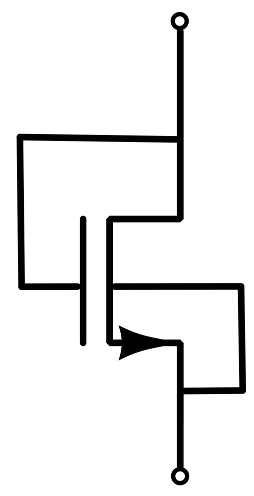
$$\delta_0 = \delta_n = \frac{1}{2}$$



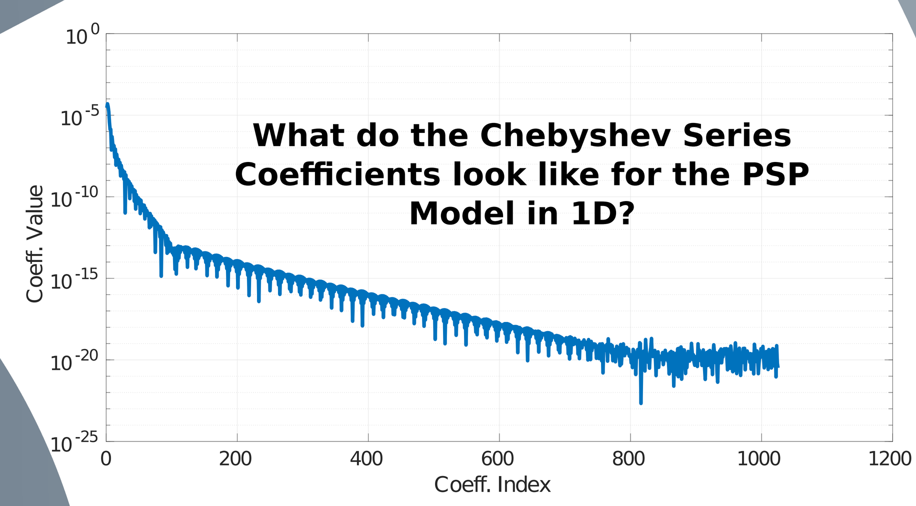


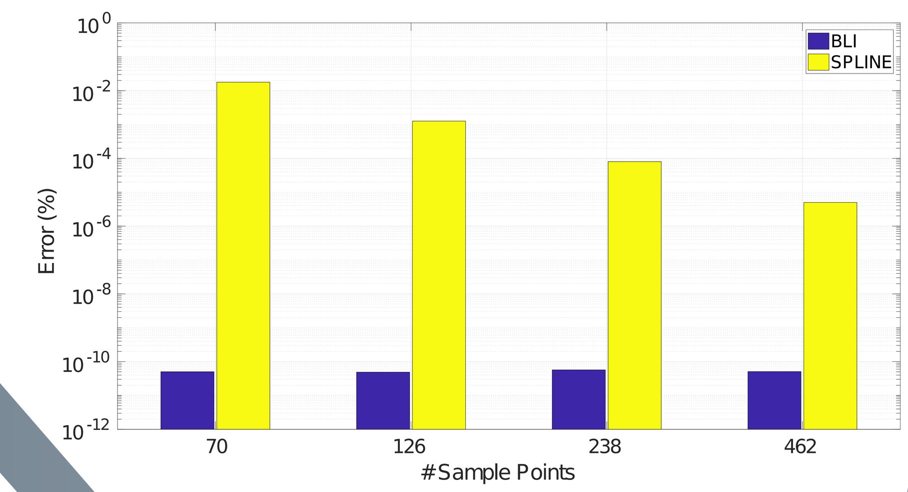


Reconstruction Error with Spline and BLI Compared for f(x)

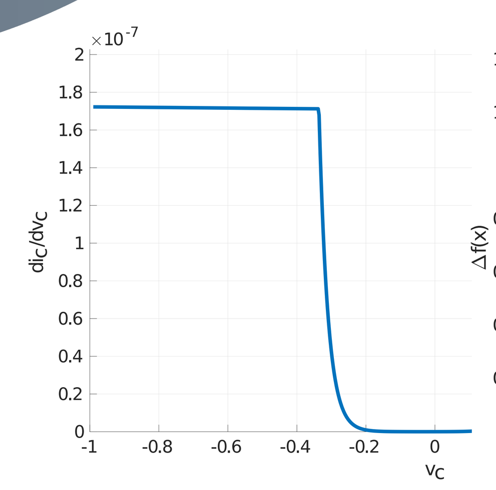


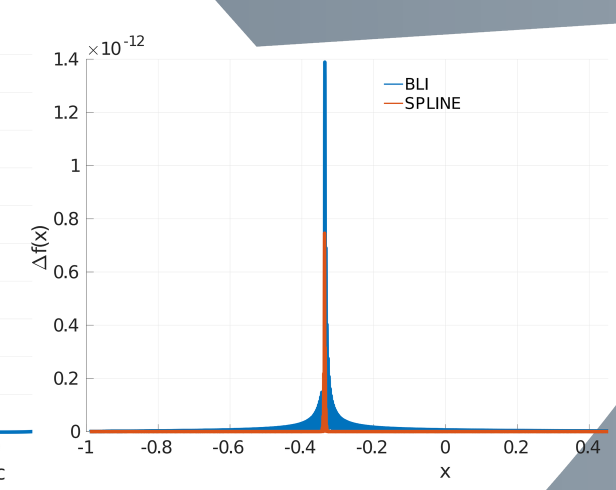
A 1D device model for illustration





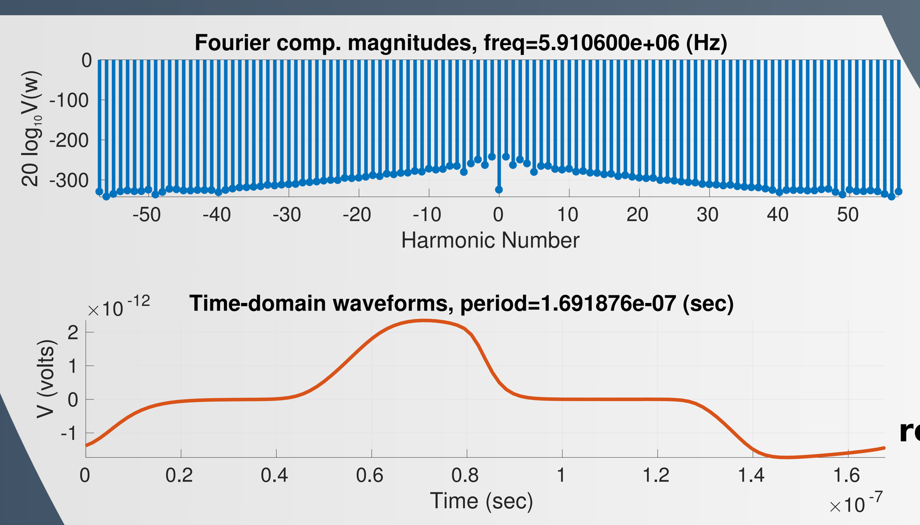
At best, how much better is BLI as compared to Splines for the PSP Model in 1D!



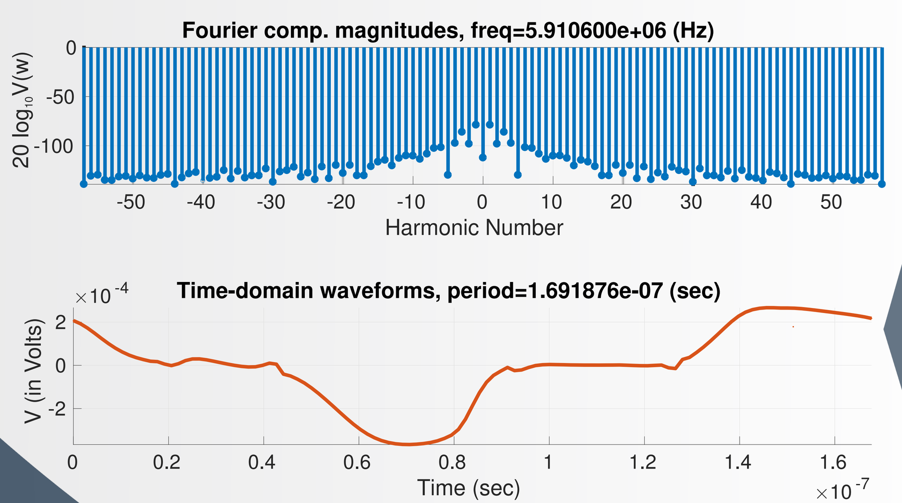


Need for Piecewise Polynomials as the device functions may not be smooth

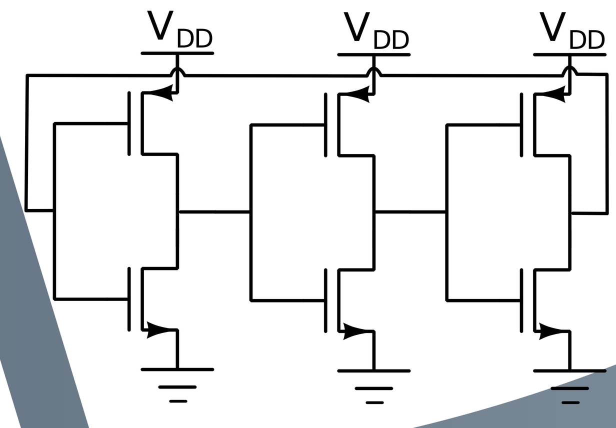
Experiments and Results

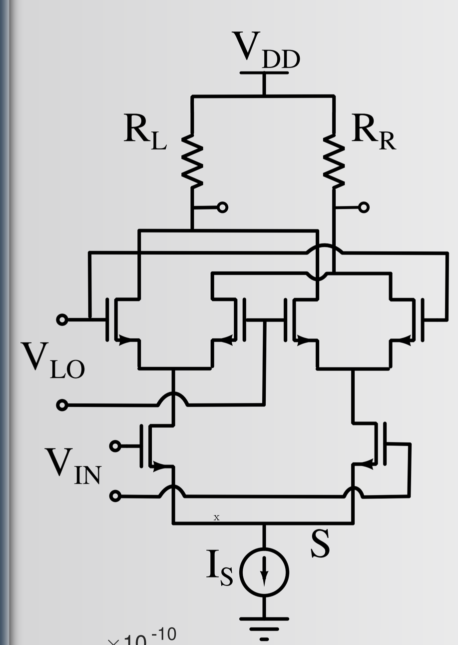


Harmonic Balance results for a 3-Stage ring oscillator with BLI

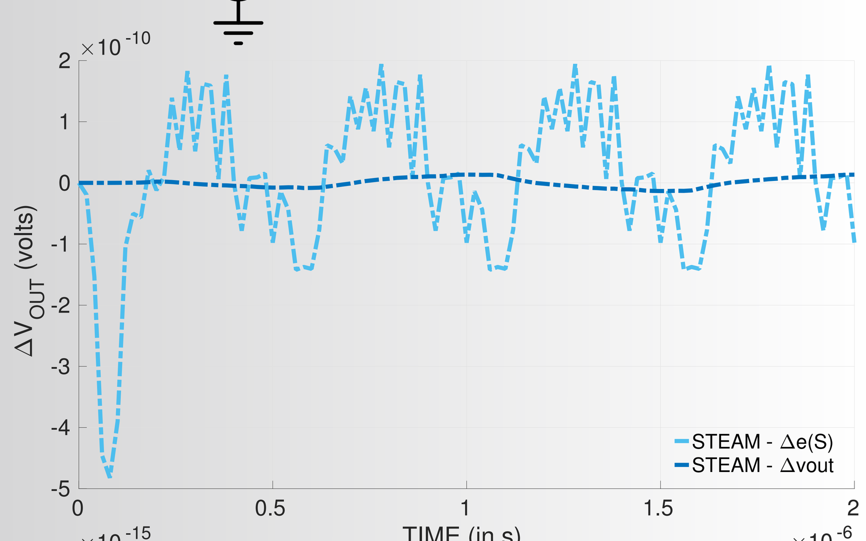


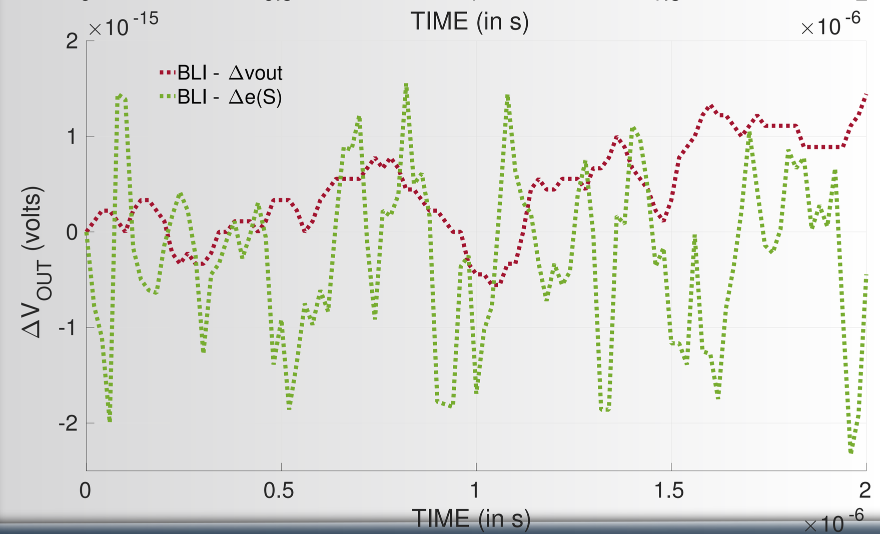
Harmonic Balance results for a 3-Stage ring oscillator with STEAM

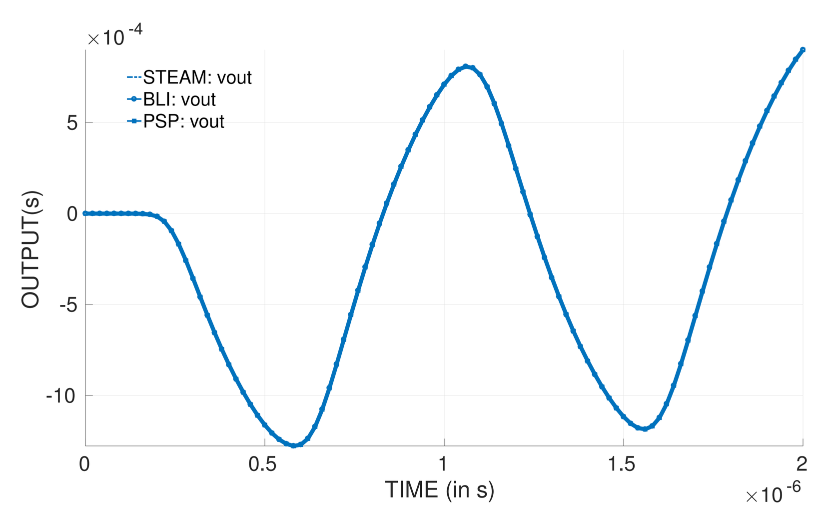




Mixer Circuit with Transient Analysis







Chebyshev Polynomials

Chebyshev polynomials are of the form

$$T_n(x) = \cos(n \cdot \cos^{-1}(x))$$

Their roots and extremas are Chebyshev Points

$$e_i = \cos\left(\frac{i \cdot \pi}{n}\right)$$

Any function can be approximated by a series of weighted Chebyshev polynomials, called a Chebyshev Series

$$p(x) = \sum_i c_i \cdot T_i(x)$$

Coefficients for Chebyshev series can be computed using DCT
Convergence properties are similar to a Fourier series
Can be easily expressed using BLI

This work was supported in part by Semiconductor Research Corporation (SRC)
Poster template based on design by Felix Breuer.