New interpolation and smoothing techniques for nonlinear models

R. Follmann, D. Köther and I. Wolff
IMST GmbH, D–47475 Kamp-Lintfort, Germany, e-mail follmann@imst.de

***Abstract*— Non-linear models in circuit simulation software often show a bad convergence behavior. In many cases, this behavior results from non-smooth intrinsic equivalent circuit element behavior due to measurement or extraction inaccuracies. In our paper we present a new method for smoothing and interpolating n-dimensional grids of equivalent circuit elements using an FFT approach. Furthermore, we demonstrate the influence of model smoothness to a harmonic balance simulation.**

I. Introduction

Equivalent circuit elements in dependence of control voltages and temperatures can be calculated from S-parameter measurements [1]. A straight forward approach to interpolate these values of intrinsic elements is the use of bi-cubic spline functions. This type of functions as well as its first derivatives are continuous in the complete area, where measurement values are available. Of course, this method does not smooth any ripple resulting from measurement inaccuracies. In order to overcome simulation problems (convergence, non-smooth behavior of compression curves, etc.) many non-linear transistor models are using dedicated functional descriptions. Most of these functions require a large parameter list in order to describe the measured / physical behavior in dependence of control voltages. In many cases these functions may fit for certain technology or type of transistor while slightly different devices can not be model with sufficient accuracy. The bi-cubic spline function approach has no limitations with respect to special technology or device characteristics. Its only limitation is the sensitivity on measurement uncertainties. Especially temperature dependent S-parameters, which can be obtained from short pulsed measurements with almost stable thermal conditions, may have quite high uncertainties. Therefore, smoothing of either measured data (S-parameters) or extracted data (e.g. C_{gs}) is required in order to achieve very accurate simulations.

II. Smoothing of spline functions

There are several approaches to smooth spline functions:

- Local smoothing,
- Smoothie splines [2],
- Functional description [3].

In this paper another method will be presented, which makes use of a Fourier transformation plus a filtering in the ”frequency” domain. Here ”frequency” can either be the transformed control voltages or temperature. Finally, the back transformation gives new coefficients for a functional description which may either be again bi-cubic splines or any other function (e.g. Fourier row).

In many cases these function values are very smooth while derivatives are more and more rough. Furthermore, derivatives are required to calculate e.g. Jacobian matrix (convergence improvement) or intermodulation properties. From the samples given before, the resulting function itself is almost equal for bi-cubic spline function and FFT function. Thus, differences in extracted values are more or less neglectable. Even for the first derivatives the influence of filtering is not very significant while the second derivatives of both functions show much more severe influence of measurement uncertainty resp. its filtering.

III. Mathematical description

A bi-cubic splines easily can be expressed as

$$f(x_1,x_2) = \sum_{j=1}^{4}\sum_{k=1}^{4} c_{jk} t^{j-1} u^{k-1} \tag{1}$$

with the spline coefficients c_{jk} and

$$t = \frac{x_1 - x_{1,\mathrm{table}}[j]}{x_{1,\mathrm{table}}[j+1] - x_{1,\mathrm{table}}[j]} \tag{2}$$

$$u = \frac{x_2 - x_{2,\mathrm{table}}[k]}{x_{2,\mathrm{table}}[k+1] - x_{2,\mathrm{table}}[k]} \tag{3}$$

A 2D FFT function g can be described as

$$g(x_1,x_2) = \sum_{m=0}^{M}\sum_{n=0}^{N} a_{mn}\cos(m\omega_1(x_1 - x_{1\min})) \cos(n\omega_2(x_2 - x_{2\min})), \tag{4}$$

with $\omega_1 = 2\pi/(x_{1\max} - x_{1\min})$ and $\omega_2 = 2\pi/(x_{2\max} - x_{2\min})$. For smoothing purpose, the following weighting function is defined:

$$W(\omega,k) = \begin{cases} 1 & : \quad \omega \le \omega_{\min} \\ 0 & : \quad \omega \ge \omega_{\max} \\ \cos^k\left(\frac{\omega-\omega_{\min}}{\omega_{\max}-\omega_{\min}}\frac{\pi}{2}\right) & : \quad \begin{cases} \omega_{\min} \le \omega \le \omega_{\max} \\ 0 < k. \end{cases} \end{cases} \tag{5}$$

This leads to a smoothed 2D FFT description as shown in equation (6).

$$\tilde{g}(x_1,x_2) = \sum_{m=0}^{\tilde{M}} \sum_{n=0}^{\tilde{N}} a_{mn} W(m\omega_1, K_1) W(n\omega_2, K_2)$$
$$\cos(m\omega_1(x_1 - x_{1\min}))\cos(n\omega_2(x_2 - x_{2\min})) \qquad (6)$$

with the new indices $\tilde{M}$ and $\tilde{N}$

$$\tilde{M} = \min(M, \frac{\omega_{1\max}}{\omega_1}) \quad \text{and} \quad \tilde{N} = \min(N, \frac{\omega_{2\max}}{\omega_2}). \qquad (7)$$

Only cosine-terms are used in FFT description in order to prevent unsteadiness. In this way the function is defined beyond measured respective extracted range by mirroring data at $x_{\max}$ (for FFT transformation only). This steady function (extended range) reduces higher frequency amplitudes, which would result from unsteady functional extension.

Even a cosine functional extension has unsteady derivatives at mirror point $x_{\max}$. Introducing a guard interval with adjustable length between initial function range and mirror range and continuous interpolation in between results a much better behavior in ”frequency” domain. These little tricks guarantee that the data used for FFT transformation has continuous derivatives. Otherwise, unsteady functions would cause a decrease of amplitudes in frequency domain proportional to a *si*-function ($\sin(x)/x$). That is a quite poor decay and high frequency parts are required at achieve smooth behavior. If high frequency amplitudes are required, filtering in the time domain will become very critical. The reduction of high frequency contribution in this case would be related to oscillation of the function in the definition range.

FFT coefficients simply can be determined by calculating the function values at required points by use of the bi-cubic spline function. Then, discrete FFT transformation provide FFT coefficients.

Filtering of coefficients for FFT function can easily be done. Using the exponential cosine filter with adjustable exponent, frequency length and corner frequency offers proper improvement of smooth behavior.

Considering all these opportunities in the right way, the coefficients of an FFT function can be obtained. This function can directly be used for interpolation between $x_{\min}$ and $x_{\max}$. Beyond these values special extrapolation methods have to be considered. The FFT function itself never can be used for this purpose. Guard interval and mirror range are not physically defined.

Furthermore, the described method can be improved. It is possible to define optimization criteria as sum of deviations from measured function values plus contributions from higher order derivatives. The deviation of FFT interpolated values g from measured values y_{i} can be defined as

$$e_0 = \sum_{i=0}^{I} (y_i - g(x_{1i}, x_{2i}))^2. \qquad (8)$$

As a second criterion a minimization of derivatives can be used:

$$e_{ij} = \int_{x_{1\min}}^{x_{1\max}} \int_{x_{2\min}}^{x_{2\max}} \left(\frac{\partial^{(i+j)} g}{\partial x_1^{(i)} \partial x_2^{(j)}} \right)^2. \qquad (9)$$

The differences between measured respective extracted data and FFT function is calculated as well as the integral over the square of the derivatives of the FFT function across the entire period in order to achieve the total error E

$$E = w_0 e_0 + \sum_{i=0}^{I} \sum_{\substack{j=0 \\ j+i\neq 0}}^{J} w_{ij} e_{ij} \qquad (10)$$

with the user defined weighting factors w_{ij} and the spectral amplitude coefficients a_{mn} to be optimized. The advantage of this integration is that these quantities can be calculated analytically. By this approach all error contributions can be calculated by simples sums according to number of frequency dimensions. Nevertheless, due to quite high number of parameters (frequency amplitudes) the optimization process takes some time. On the other side, this has to be done just once per non-linear model or device. After this optimization and smoothing, within the CAD program fast functions like bi-cubic splines or even FFT functions can be used. Parameters of optimization can easily be adjusted.

IV. Examples

Fig. 1 shows an extracted gate-source capacitance for a low noise HEMT. As can be seen, the extracted grid is quite smooth. Nevertheless, when looking at the second

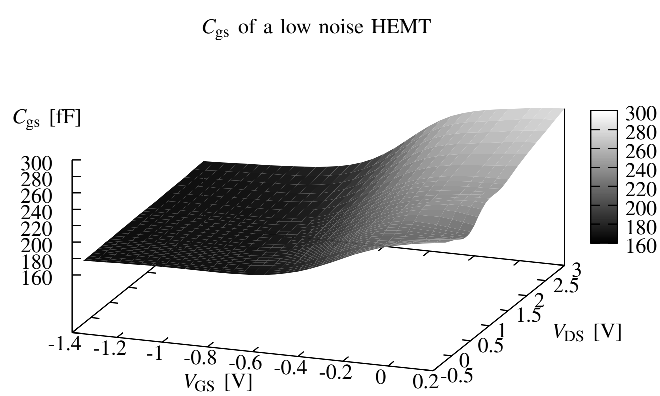

Fig. 1. Extracted gate-source capacitance C_{gs} for an low noise HEMT device.

derivative, large variations can be obtained (fig. 2). This is due to the fact that the underlying spline function simulates each measurement outlier correctly.

Thus, we applied the FFT smoothing algorithm proposed

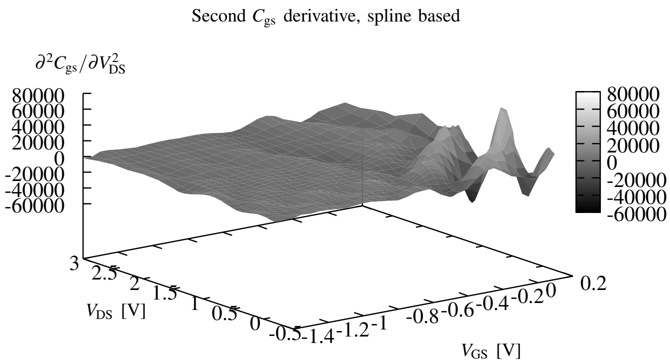

Fig. 2. Second derivative (spline based) of extracted C_{gs}.

earlier in this paper. After optimizing the FFT coefficients for an order of 15, no difference between the original extracted grid (fig. 1) and the FFT interpolated one could be obtained. Using the FFT smoothed C_{gs}-grid delivers

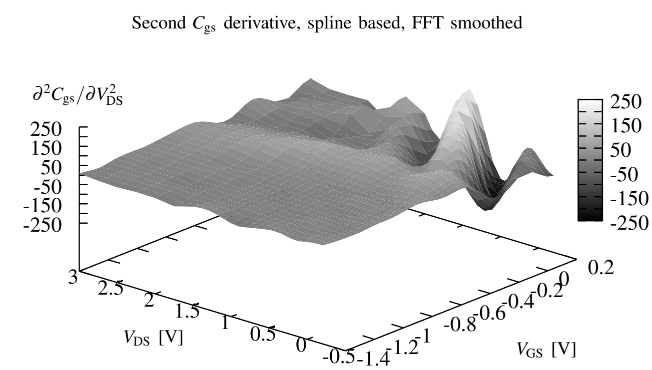

Fig. 3. Second derivative (spline based, fft smoothed) of extracted C_{gs}.

now a very smooth second derivative, as can be seen in fig. 3. Here, the scale has been adjusted.

Table I shows, how important smooth intrinsic element grids are for the simulation convergence speed under large signal conditions.

Extraction	non smoothed	medium smoothed	highly smoothed
Sim. time	48.26 s	38.12 s	36.57 s

TABLE I

HARMONIC BALANCE SIMULATION TIME.

The table shows the computation time for a two-tone intermodulation simulation (-10 dBm $\ldots$+ 10 dBm step 0.5 dB RF power, $f = 2$ GHz, 100 kHz spacing, 10 harmonics) for a LDMOS device [4]. Despite of the total computation time it is obvious that harmonic balance within the circuit simulation software ADS converges quicker, the more smooth the extracted values are. A non-smoothed model converges 25% slower than a model based upon smoothed extraction data.

REFERENCES

[1] R. Follmann, J. Borkes, P. Waldow, and I. Wolff, “Extraction and Modeling Methods for FET devices,” *IEEE Microwave Magazine*, vol. 1, no. 3, pp. 49–55, 2000.

[2] V. Cuoco, M. P. v.d. Heijden, and L. de Vreede, “The ”Smoothie” Data Base Model for the Correct Modeling of Non-Linear Distortion in FET Devices,” *IEEE International Microwave Symposium Seattle, Washington*, vol. IF-TH-42, pp. 2149–2152, 2002.

[3] W. R. Curtice, J. A. Pla, D. Bridges, T. Liang, and E. E. Shumate, “A New Dynamic Electro-Thermal Nonlinear Model For Silicon RF LDMOS FETs,” *IEEE Trans. Microwave Theory Tech.*, vol. 1, 1999.

[4] R. Follmann, D. Köther, R. Stahlmann, I. Wolff, J. Gajadharsing, M. Versleijen, and J. Sveshtarov, “A new precise large signal LDMOS(T) model including time delay effects,” *IEEE International Microwave Symposium Fort Worth, Texas*, vol. IFWE-9, 2004.