An Algorithm for Numerical Reference Generation in Symbolic Analysis of Large Analog Circuits

Ignacio García-Vargas, Mariano Galán, Francisco V. Fernández and Angel Rodríguez-Vázquez
IMSE, Centro Nacional de Microelectrónica, Edif. CICA, Avda. Reina Mercedes s/n, E-41012 Sevilla, Spain

Abstract
This paper addresses the problems arising in the calculation of numerical references (network function coefficients), essential for an appropriate error control in simplification before and during generation algorithms for symbolic analysis of large analog circuits. The conventional polynomial interpolation method reveals to be unable to handle the large circuit sizes needed, mainly due to the dramatic effect of round-off errors. This paper introduces a new algorithm able to accurately calculate the network function coefficients of large analog circuits in an efficient way.

1. Introduction
Symbolic circuit analysis refers to the calculation of network functions where all or part of the circuit parameters are represented by symbols. The applicability of symbolic simulation techniques to the analysis and synthesis of analog integrated circuits has been known for a long time. See for instance [1] for an actualized review of techniques and applications of symbolic analysis.

Given the exponential increase of the resulting expression length with the circuit size, symbolic expression simplification has been recognized to be essential for both: formula interpretation by human designers and computer manipulation for repetitive evaluations in design automation applications [1]. Simplification in symbolic analysis is defined as the reduction of formula complexity by eliminating insignificant terms or sub-expressions, based on numerical estimates of the symbolic parameters. Conventional simplification approaches first calculate the complete symbolic expression, and then simplify it—commonly called Simplification After Generation (SAG).

Due to the necessity of generating complete expressions, SAG techniques are constrained to low and medium complexity circuits (below about 50 symbols). Larger circuits can be analyzed by using the newest approaches of simplifying during, or before to, the analysis stage. They are called Simplification During Generation (SDG) and Simplification Before Generation (SBG) techniques respectively.

The rationale behind SDG techniques is that terms of the network function coefficients can be generated strictly in decreasing order of magnitude starting with the largest [2]-[4]. Consider a generic symbolic network function

\[ H(s, x) = \frac{N(s, x)}{D(s, x)} = \sum_{j=1}^{M} f_j(x) \cdot s^j \]

and

\[ h_k(x) = \sum_{j=1}^{N} g_j(x) \cdot s^j \]

represents either \( f_j(x) \) or \( g_j(x) \) in (1). Refs. [2]-[4] generate the \( P \) most significant terms in (2) until the sum of the generated terms represents a given fraction of the total magnitude of the coefficient,

\[ \left| h_k(x_o) - \sum_{j=1}^{P} h_j(x_o) \right| < \varepsilon_k \]

where \( x_o \) represents a design point of the circuit parameters and \( \varepsilon_k \) is an error control parameter. As shown in (3), the total magnitude of each circuit coefficient \( h_k(x_o) \) must be known a priori, without having the symbolic expression available.

On the other hand, SBG takes place in the network under analysis, replacing those elements (or subcircuits), whose contribution (appropriately measured) to the network function is negligible, with a zero-admittance or zero-impedance element. Then, the reduced circuit is much easier to analyze. As in SDG, most accurate error control criteria compare a numerical evaluation of the simplified expres-
sion with a numerical estimate of the complete (exact) expression.

Hence, both simplification methodologies require to compare with a reference. The calculation of this reference is the objective of this paper, with implies the realization of a symbolic analysis with only the complex frequency $s$ as symbolic variable.

2. Polynomial interpolation techniques

2.1 Background

The polynomial interpolation method [5],[6] constitutes a very efficient technique for the calculation of network function coefficients with only the complex frequency in symbolic form.

The polynomial interpolation starts from the idea that a $n$-th order polynomial,

$$ P(s) = p_0 + p_1 s + p_2 s^2 + \ldots + p_n s^n $$

(4)

is fully determined by its value at $(n+1)$ points.

It is widely accepted that the interpolation points should not have a wide spread because of the overflow problem. It has been shown that the use of equally-spaced interpolation points in the unit circle gives the best results concerning numerical accuracy and stability [5],[6]. Once the values of (4) at all these points $P(s_k)$ are known, the polynomial coefficients can be obtained through the Inverse Discrete Fourier Transform (IDFT),

$$ \hat{p}_i = \frac{1}{K} \sum_{k=0}^{K-1} P(s_k) e^{\frac{2\pi i k}{K}} \quad i = 0, 1, \ldots, K-1 $$

(5)

where

$$ \hat{p}_i = \begin{cases} p_i & \text{for } i \leq n \\ 0 & \text{otherwise} \end{cases} $$

(6)

The number of interpolation points, $K$, should be $(n+1)$, but in most cases, like that we are dealing with, the polynomial order is not known beforehand. Hence, an upper estimate on $K$ must be done, and (5) should be identically 0 for those coefficients over the $n$-th power. As we will see, this becomes a major problem when working with analog integrated circuits.

The problem of calculating $P(s_k)$ (representing either the numerator $N(s_k)$ or the denominator $D(s_k)$) at the different interpolation points, is discussed now. Assume that an appropriate formulation method, i.e. modified nodal analysis, has been applied on the circuit so that the network equations can be written as [6]:

$$ Y_{MNA} X = E $$

(7)

where $Y_{MNA}$ is the modified nodal matrix, $X$ contains nodal voltages and auxiliary currents, and $E$ accounts for the influence of independent sources. Once any frequency-dependent element in $Y_{MNA}$ is evaluated at the interpolation point $s=s_k$, the value of the network function

$$ H(s_k) = \frac{N(s_k)}{D(s_k)} $$

(8)

can be obtained by applying LU decomposition to (7). The denominator of the network function is easily obtained as:

$$ D(s_k) = |Y_{MNA}(s_k)| $$

(9)

and the numerator $N(s_k)$ is easily obtained from (8) and (9):

$$ N(s_k) = H(s_k) \cdot D(s_k) $$

(10)

2.2 Round-off errors in DFT

One major problem in polynomial interpolation applied to analog integrated circuits is the dramatic effect of round-off errors, due to the finite precision arithmetics of computers. Let us check the associated problems through a real-world example: the calculation of the differential voltage gain as a function of $s$ in the positive feedback OTA of Fig. 1. The estimate on the upper bound of the polynomial order for this circuit is 9. The interpolated polynomial coefficients when using interpolation points located at the unit circle are given in Table 1a.

As shown in Table 1a, many coefficients have a non-zero imaginary component. This is due to the round-off errors, which avoid perfect cancellations of the imaginary parts in the DFT. But this is not a major problem. The results obtained for the real parts make us doubt about their correctness. As can be seen, most calculated coefficients have the same order of magnitude than the imaginary parts. So, one may wonder which of those coefficients can be correct and which are identically 0. As indicated by (6) the zero coefficients would indicate the real polynomial order. But now this criterion seems not valid.

On the other hand, it is known that each symbolic term is given by a product of admittances: transconductances and

![Figure 1: Positive feedback OTA.](image-url)
Table 1: (a) transfer function coefficients for the differential voltage gain of Fig.1: using interpolation points on the unit circle, (b) normalized coefficients using a frequency scale factor of $10^9$.

<table>
<thead>
<tr>
<th>$s^i$</th>
<th>Numerator</th>
<th>Denominator</th>
<th>$s^i$</th>
<th>Numerator</th>
<th>Denominator</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s^5$</td>
<td>-5.8296e-25</td>
<td>+0.0</td>
<td>-5.8296e-25</td>
<td>+0.0</td>
<td>8.9418e-30</td>
</tr>
<tr>
<td>$s^4$</td>
<td>-1.5484e-33</td>
<td>-j2.958e-41</td>
<td>-5.8296e-25</td>
<td>+0.0</td>
<td>8.9418e-30</td>
</tr>
<tr>
<td>$s^3$</td>
<td>-2.5254e-41</td>
<td>+j8.367e-41</td>
<td>-1.1010e-40</td>
<td>-j7.3880e-41</td>
<td></td>
</tr>
<tr>
<td>$s^2$</td>
<td>-5.5101e-41</td>
<td>+j1.0646e-43</td>
<td>-1.0937e-24</td>
<td>-j2.227e-40</td>
<td>+j2.9260e-40</td>
</tr>
</tbody>
</table>

3.1 Interpolation of high order polynomials

As reported in [6], the scaling procedure described above works well only for polynomials up to about tenth order (for typical magnitudes involved in analog integrated circuits). The main reason behind this is that for higher order polynomials it is impossible to find a scale factor that keeps all the coefficients within a range, where

$$P(s) = \sum_{i=0}^{n} p_i s^i = \sum_{i=0}^{n} \frac{p_i}{g} \left(\frac{1}{s^i}\right) = g^{-M} \sum_{i=0}^{n} p_i \left(\frac{1}{g}\right)^i s^i$$

where $M$ is the number of conductances involved in coefficient $p_0$. We can observe that increasing the frequency scale factor has the same effect than decreasing the conductance scale factor. The main problem is that the appropriate scale factor is not known a priori.

3.2 Adaptive scaling

The solution we propose to solve these problems is to...
select at each interpolation a region of valid coefficients. Our objective is to make as few interpolations as possible. For this, the scale factors must be selected trying to meet two conditions:

- The generated regions of valid coefficients must be separated—the smaller overlapping the better.
- Joining all valid coefficients must give all polynomial coefficients.

The proposed algorithm performs successive interpolations. At each interpolation, if coefficients with 6 significant digits need to be calculated, then all coefficients which prior to denormalization are smaller than \(10^{-13} + 6 \times \max_j |p_j|\) must be neglected. For each interpolation, the appropriate scale factors are calculated using information from previous interpolations.

For a detailed explanation of the proposed algorithm we will use the \(\mu A 741\) opamp. Due to the limited space and without loss of generality we will limit ourselves to the denominator of its voltage gain.

The first interpolation is performed using the inverse of the mean value of the conductances. Analogously, the conductance scale factor is calculated through the inverse of the mean value of the conductances. The objective of these heuristics is to first generate the widest region of valid coefficients.

For the example above we can observe in Table 2a how these first frequency and conductance scale factors generate a region of valid coefficients which goes from \(p_0\) to \(p_{12}\) (light-shadowed). This region is determined from the coefficient with largest normalized absolute value (dark-shadowed): all coefficients larger than:

\[
10^{-13} + 6 \times 1.28095 \times 10^{124} = 1.28095 \times 10^{117}
\]

are considered correct.

To proceed to coefficients with higher powers of \(s\) a new conductance scale factor \(g'\) and a new frequency scale factor \(f'\) are calculated by normalizing the previous ones:

\[
g' = \frac{g}{\sqrt[q]{\Delta q}} \quad f' = f \sqrt[q]{\Delta q}
\]

and \(q\) is given by:

\[
|p_e|q^r = |p_m|q^m \times 10^{13+r}
\]

where \(p_e\) and \(p_m\) are the last and maximum coefficients within the last valid region respectively, and \(r\) is a tuning factor.

A new polynomial interpolation with the scale factors calculated in (13) gives a new region of valid coefficients. The objective of (14) is that the \(e\)-th coefficient of the previous interpolation is one of the first coefficients in the new region of valid coefficients and, hence, the new region has the smallest overlap with the previous one.

To proceed to coefficients of smaller powers of \(s\), then factor \(q\) in (13) is calculated through

\[
|p_e|q^b = |p_m|q^m \times 10^{13+r}
\]

where \(p_b\) is the first coefficient of the previous valid region.

The results of the first interpolation in the example above (see Table 2a) are used to calculate new scale factors \(f'\) and \(g'\) which provide a region of valid coefficients of higher powers of \(s\). For this, we apply (13) and (14) using \(p_e=p_{12}\) and \(p_m=p_5\). The generated valid coefficients are shown in Table 2b, where the region of valid coefficients has shifted to the region between the 13-th and the 30-th coefficient.

### Table 2: Denominator coefficients of the voltage gain of \(\mu A 741\).

<table>
<thead>
<tr>
<th>(s^2)</th>
<th>Normalized</th>
<th>Denormalized</th>
<th>(s^4)</th>
<th>Normalized</th>
<th>Denormalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s^0)</td>
<td>(-2.82408e+118)</td>
<td>(-1.6419e-90)</td>
<td>(s^2)</td>
<td>(-7.32222e+122)</td>
<td>(-1.45352e-92)</td>
</tr>
<tr>
<td>(s^1)</td>
<td>(-8.26327e+123)</td>
<td>(-5.60064e-98)</td>
<td>(s^3)</td>
<td>(-2.04859e+92)</td>
<td>(-2.45375e-184)</td>
</tr>
<tr>
<td>(s^2)</td>
<td>(-1.28095e+124)</td>
<td>(-2.96432e+104)</td>
<td>(s^4)</td>
<td>(-7.57228e+93)</td>
<td>(-1.24499e-217)</td>
</tr>
<tr>
<td>(s^3)</td>
<td>(-1.20867e+124)</td>
<td>(-9.55018e-111)</td>
<td>(s^5)</td>
<td>(-8.5409e+93)</td>
<td>(-2.35783e-226)</td>
</tr>
<tr>
<td>(s^4)</td>
<td>(-3.17468e+123)</td>
<td>(-5.61933e+103)</td>
<td>(s^6)</td>
<td>(-7.25477e+93)</td>
<td>(-3.35638e-235)</td>
</tr>
<tr>
<td>(s^5)</td>
<td>(-8.7689e+116)</td>
<td>(-4.3084e+176)</td>
<td>(s^7)</td>
<td>(-2.02556e+86)</td>
<td>(-2.7733e-339)</td>
</tr>
</tbody>
</table>

### Table 3: Denominator coefficients of the voltage gain of \(\mu A 741\).

<table>
<thead>
<tr>
<th>(s^2)</th>
<th>Normalized</th>
<th>Denormalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>(s^0)</td>
<td>(-9.44426e+99)</td>
<td>(-2.7733e-339)</td>
</tr>
<tr>
<td>(s^1)</td>
<td>(-3.67933e+103)</td>
<td>(-5.2373e-421)</td>
</tr>
<tr>
<td>(s^2)</td>
<td>(-3.9978e+103)</td>
<td>(-3.13903e+432)</td>
</tr>
<tr>
<td>(s^3)</td>
<td>(-3.8434e+103)</td>
<td>(-5.10259e+443)</td>
</tr>
<tr>
<td>(s^4)</td>
<td>(-2.1869e+100)</td>
<td>(-1.34792e+510)</td>
</tr>
<tr>
<td>(s^5)</td>
<td>(-9.75596e+98)</td>
<td>(-1.1215e+522)</td>
</tr>
</tbody>
</table>

The described algorithm has been implemented using sparse matrix techniques. Each iteration for the example in Table 2 and Table 3 spends 3.9s on a SPARC Station10.

The accuracy of the results obtained in this example is demonstrated through the comparison of the Bode diagrams obtained from the interpolation of numerator and denominator of the voltage gain of \(\mu A 741\) and those obtained through a commercial electrical simulator. Such comparison is illustrated in Fig. 2 where perfect matching can be observed.
If between two consecutive valid regions, with scale factors \( f_1, g_1 \) and \( f_2, g_2 \), some incorrect coefficients remain, then new scale factors \( f_{new}, g_{new} \) are calculated as follows,

\[
\begin{align*}
\frac{f_1}{g_1} + \frac{f_2}{g_2} &= \frac{\log(g_1) + \log(g_2)}{\log(g_1) + \log(g_2)}
\end{align*}
\]

\[
\begin{align*}
f_{new} &= 10^{\frac{1}{2}}
g_{new} &= 10^{\frac{1}{2}}
\end{align*}
\]  

The new polynomial contains the coefficients that still have to be calculated and need only \( l-k+1 \) interpolation points. This reduction of the problem complexity has been applied to the example above. The CPU time to get the same results using (17) reduces to 3.9s for the first iteration, 2.3s for the second one and 0.9s for the third one.

Also, it has been shown in the example above that for a given scale factor some coefficients are smaller than the error level. This means that for this scaling, these coefficients affect the polynomial value less than the error level, and, hence, can be neglected. Neglecting high order coefficients is useful because it allows to handle the polynomial as of smaller order, reducing in this way the number of points needed in the interpolation.

3.3 Increasing efficiency of the adaptive scaling mechanism

In each polynomial interpolation the computational effort depends on the number of interpolation frequencies needed. The problem complexity can be reduced at subsequent iterations of previous algorithm, once the coefficients of the largest or smallest powers of \( s \) have been calculated. Assume the coefficients \( p_0 \ldots p_{k+1} \) and \( p_{n+1} \ldots p_n \) have already been calculated, then the polynomial is transformed as follows,

\[
P'(s) = p_k + \ldots + p_1 s^{1-k}
\]

\[
P(s) - \sum_{i=0}^{k-1} p_i s^i - \sum_{i=k+1}^{n} p_i s^i
\]

This reduces the problem complexity by \( l-k+1 \) times. The reduced polynomial \( P'(s) \) is then interpolated, and the process is repeated.

The fundamental topic of numerical reference generation for error control in simplification before and during generation algorithms for symbolic analysis has been addressed. A new adaptive scaling mechanism has been introduced in the polynomial interpolation technique. This technique efficiently solves the problems arising when analyzing medium and large analog integrated circuit. A real world example has been shown to demonstrate the validity of the proposed approach.

Conclusions

The references consist of five papers: