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 <u>S</u>implification <u>A</u>fter <u>G</u>eneration (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 <u>S</u>implification <u>D</u>uring <u>G</u>eneration (SDG) and <u>S</u>implification <u>B</u>efore <u>G</u>eneration (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

$$H(s, \boldsymbol{x}) = \frac{N(s, \boldsymbol{x})}{D(s, \boldsymbol{x})} = \frac{\sum_{i=1}^{M} f_i(\boldsymbol{x}) \cdot s^i}{\sum_{j=1}^{N} g_j(\boldsymbol{x}) \cdot s^j} \tag{1}$$

and

$$h_k(\boldsymbol{x}) = \sum_{l=1}^{T} h_{kl}(\boldsymbol{x}) \tag{2}$$

represents either $f_i(\boldsymbol{x})$ or $g_j(\boldsymbol{x})$ 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,

$$\left| h_k(\boldsymbol{x_o}) - \sum_{l=1}^{P} h_{kl}(\boldsymbol{x_o}) \right| < \varepsilon_k \left| h_k(\boldsymbol{x_o}) \right| \tag{3}$$

where $\boldsymbol{x_o}$ represents a design point of the circuit parameters and ε_k is an error control parameter. As shown in (3), the total magnitude of each circuit coefficient $h_k(\boldsymbol{x_o})$ 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-

* This work has been performed in the framework of the AMADEUS Project within the ESPRIT IV Program of the CEC.

ED&TC '97 on CD-ROM

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,

$$P(s) = p_0 + p_1 s + p_2 s^2 + \ldots + p_n s^n \tag{4}$$

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 $K \geq n+1$ 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),

$$\hat{p}_i = \frac{1}{K}\sum_{k=0}^{K-1} P(s_k)\, e^{-\frac{2\pi i k}{K}} \qquad i = 0, 1, \ldots, K-1 \tag{5}$$

where

$$\hat{p}_i = \begin{cases} p_i & \text{for } i \leq n \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

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]:

$$\boldsymbol{Y}_{MNA}\boldsymbol{X} = \boldsymbol{E} \tag{7}$$

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

$$H(s_k) = \frac{N(s_k)}{D(s_k)} \tag{8}$$

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

$$D(s_k) = \left|\boldsymbol{Y}_{MNA}(s_k)\right| \tag{9}$$

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

$$N(s_k) = H(s_k) \cdot D(s_k) \tag{10}$$

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.

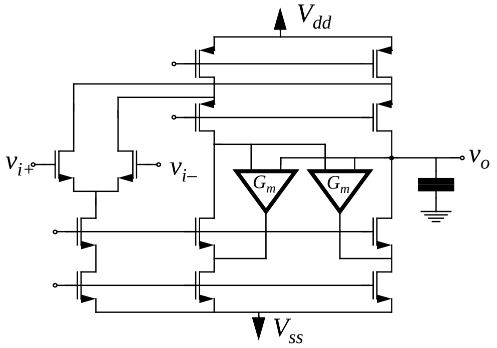

Figure 1: Positive feedback OTA.

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

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.

s^i	(a)		(b)	
	Numerator	Denominator	Numerator	Denominator
s^0	–5.8296e–25 +j0.0	8.9418e–30 +j0.0	–5.8296e–25 –j1.1106e–42	8.9418e–30 +j2.2028e–43
s^1	–1.5484e–33 –j2.2958e–41	3.8525e–36 –j7.0064e–47	–1.5484e–24 –j1.1010e–40	3.8525e–27 –j7.3880e–41
s^2	–2.5254e–41 +j1.8367e–41	2.3920e–43 –j1.4013e–46	–1.7843e–24 –j3.2266e–40	4.2042e–25 –j5.0134e–41
s^3	–5.5101e–41 +j0.0	1.0646e–43 –j1.4013e–46	–1.0937e–24 –j2.227e–40	1.3193e–24 +j2.9260e–40
s^4	7.3468e–41 +j3.6734e–41	–8.4077e–46 –j5.6051e–46	–3.2376e–25 +j1.0883e–40	1.6913e–24 +j5.7810e–40
s^5	–4.5917e–41 +j3.5695e–41	2.1019e–45 –j5.4751e–46	–9.8792e–27 +j3.2146e–40	1.0968e–24 +j3.6903e–40
s^6	5.5101e–41 +j4.1326e–41	–4.2039e–46 –j5.6051e–46	1.2567e–26 +j3.6827e–40	3.5870e–25 –j7.3804e–41
s^7	1.8826e–40 –j2.0203e–40	1.0243e–43 +j3.0828e–45	–4.6035e–40 –j9.511e–40	4.7236e–26 +j2.5243e–40
s^8	–1.1479e–40 +j5.5101e–41	–1.8020e–43 –j5.6051e–46	–6.2330e–40 +j5.1814e–40	–2.110e–40 –j5.1437e–40
s^9	–1.7448e–40 –j1.6530e–40	6.8383e–43 +j2.5223e–45	–2.2385e–39 +j8.7655e–40	–7.5904e–41 –j1.776e–39

capacitors [5][1]. The coefficient of s^i has one more transconductance and one less capacitor in each term than the coefficient of s^{i+1}. Taking into account the typical magnitudes of conductances and capacitors in analog integrated circuits, we can expect a ratio of 10^6 to 10^{12} between each pair of consecutive coefficients. This clearly does not happen in most coefficients in Table 1a.

The reason for this behavior can be found in the unavoidable numerical noise in digital computers. The numerical error level in the polynomial interpolation depends on the coefficient p_i in (4) having the largest absolute value. This error is about $10^{-13} \times max_i|p_i|$ in a computer with 16-decimal-digit accuracy [6]. The spread of values between the maximum and minimum coefficient should be well below this error to ensure numerical accuracy of the calculated coefficients. Hence, it is not difficult to see that the second and higher order coefficients in Table 1a are not valid any more.

3. Interpolation using scaling procedures

In order to reduce the spread of coefficient values in typical analog integrated circuits, the complex frequency variable (equivalently the capacitor values) should be scaled before performing the polynomial interpolation on the unit circle. Also, this suggests conductance scaling as another alternative.

For illustration's sake, Table 1b shows the correct normalized coefficients, which were obtained using a frequency scale factor of 10^9. The shadowed coefficients are well above the error level $\left(10^{-13} \times max_i|p_i|\right)$ and can be considered to be correct.

The effect of conductance and frequency scaling on the relative value of the coefficients is analogous. Assume that g is the conductance scale factor and f is the frequency scale factor, then the normalized coefficients p'_i are:

$$P(s) = \sum_{i=0}^{n} p'_i s^i = \sum_{i=0}^{n} p_i f^i g^{M-i} s^i = g^M \sum_{i=0}^{n} p_i \left(\frac{f}{g}\right)^i s^i \qquad (11)$$

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.

3.1 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 $10^{-13+\sigma} \times max_i|p_i|$ range, where σ is the number of significant digits desired in the coefficients.

A possible solution is to try a relatively large set of scale factors. For each pair of scale factors only a limited set of coefficients––those which keep above the error level––are equal in both interpolations and can be considered to be valid.

There is obviously the problem of appropriately selecting the scale factors. If the distance between them were too large, some coefficients might never be above the error level, or be above only for one scale factor. Then, those coefficients would give a different result for each scale factor and would not be possible to calculate them correctly. If the distance between the scale factors is too small, then many coefficients will be calculated several times, leading to much higher computational effort.

3.2 Adaptive scaling

The solution we propose to solve these problems is to

[1] We are considering circuits containing capacitors as the only frequency-dependent element. Circuits containing inductors can be analysed using transformation methods [5].

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 $10^{-13+6} \times max_i|p_i|$ 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 μA741 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 capacitors as frequency scale factor. 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:

$$10^{-13+6} \times 1.28095 \times 10^{124} = 1.28095 \times 10^{117} \qquad (12)$$

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:

$$g' = \frac{g}{\sqrt{q}} \qquad f\,' = f\sqrt{q} \qquad (13)$$

and q is given by:

$$|p_e|q^e = |p_m|q^m \times 10^{13+r} \qquad (14)$$

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

$$|p_b|q^b = |p_m|q^m \times 10^{13+r} \qquad (15)$$

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_3. 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. A third interpolation applying (13) and (14) to the coefficients in Table 2b gives the remaining coefficients (see Table 3).

Table 2: Denominator coefficients of the voltage gain of μA741.

(a) First interpolation			(b) Second interpolation		
s^i	Normalized	Denormalized	s^i	Normalized	Denormalized
s^0	–2.82408e+118	–1.6419e–90	s^0...		
s^1	–7.32222e+122	–1.45352e–92	s^{13}	–3.52987e+91	–4.3694e–176
s^2	–8.26327e+123	–5.60064e–98	s^{14}	–2.04859e+92	–4.25375e–184
s^3	–1.28095e+124	–2.96432e–104	...	...	...
s^4	–1.20867e+124	–9.55018e–111	s^{18}	–7.57228e+93	–1.24499e–217
s^5	–7.46903e+123	–2.015e–117	s^{19}	–8.54909e+93	–2.35783e–226
s^6	–3.17468e+123	–2.92428e–124	s^{20}	–7.25477e+93	–3.35638e–235
...	...	...	...	...	...
s^{12}	–2.13624e+118	–3.11759e–168	s^{30}	–2.7438e+87	–2.23949e–329
s^{13}	–8.7689e+116	–4.3694e–176	s^{31}	–2.02556e+86	–2.7733e–339
... s^{48}			... s^{48}		

Table 3: Denominator coefficients of the voltage gain of μA741.

Third interpolation		
s^i	Normalized	Denormalized
s^0 ...		
s^{31}	–9.64426e+99	–2.7733e–339
...	...	...
s^{39}	–3.61933e+103	–1.52373e–421
s^{40}	–3.9978e+103	–3.13903e–432
s^{41}	–3.48434e+103	–5.10259e–443
...	...	...
s^{47}	–2.18689e+100	–1.34792e–510
s^{48}	–9.75596e+98	–1.1215e–522

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 μA741 and those obtained through a commercial electrical simulator. Such comparison is illustrated in Fig. 2 where perfect matching can be observed.

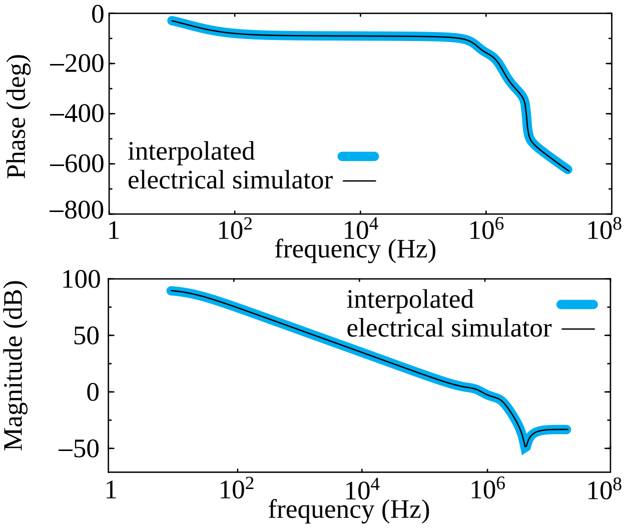

Figure 2: Bode diagrams of the voltage gain of µA741 using the interpolated coefficients and an electrical simulator.

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,

$$\frac{f_{new}}{g_{new}} = 10^{\frac{\log\frac{f_1}{g_1} + \log\frac{f_2}{g_2}}{2}} \qquad g_{new} = 10^{\frac{\log\left(g_1\right) + \log\left(g_2\right)}{2}} \tag{16}$$

Notice that in the algorithm above simultaneous scaling of both, frequency and conductance is used. This technique is used to avoid using too large (>~10^{18}) frequency or conductance scale factors. These high values occasionally occur when using a single scale factor and are responsible for an increase of the error in the calculation of numerator and denominator of the transfer function at the interpolation points.

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 $p_0...p_{k-1}$ and $p_{l+1}...p_n$ have already been calculated, then the polynomial is transformed as follows,

$$\begin{aligned} P'(s) &= p_k + \ldots + p_l \cdot s^{l-k} \\ &= \frac{P(s) - \sum_{i=0}^{k-1} p_i \cdot s^i - \sum_{i=l+1}^{n} p_i \cdot s^i}{s^k} \end{aligned} \tag{17}$$

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.

Conclusions

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.

References

[1] A. Rodríguez-Vázquez, F.V. Fernández, J.L. Huertas and G. Gielen, eds., *Symbolic Analysis Techniques and Applications to Analog Design Automation*. IEEE Press, 1997.

[2] F.V. Fernández, P. Wambacq, G. Gielen, A. Rodríguez-Vázquez and W. Sansen: "Symbolic Analysis of Large Analog Integrated Circuits by Approximation During Expression Generation," *Proc. IEEE ISCAS*, Vol. CAD, pp. 25-28, 1994.

[3] P. Wambacq, F.V. Fernández, G. Gielen, W. Sansen and A. Rodríguez-Vázquez, "Efficient Symbolic Computation of Approximated Small-Signal Characteristics of Analog Integrated Circuits," *IEEE J. Solid-State Circuits*, Vol. 30, No. 3, pp. 327-330, March 1995.

[4] Q. Yu and C. Sechen, "Approximate Symbolic Analysis of Large Analog Integrated Circuits," *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 664-671, 1994.

[5] P.M. Lin, *Symbolic Network Analysis*. Elsevier, 1991.

[6] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design.* Van Nostrand Reinhold, 1994.