SPICE 3 DISTORTION ANALYSIS

by

Jaijeet S. Roychowdhury

Memorandum No. UCB/ERL M89/48

20 April 1989

127

SPICE 3 DISTORTION ANALYSIS

by

Jaijeet S. Roychowdhury

Memorandum No. UCB/ERL M89/48

20 April 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SPICE 3 DISTORTION ANALYSIS

by

Jaijeet S. Roychowdhury

Memorandum No. UCB/ERL M89/48

20 April 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

A new implementation of distortion analysis has been integrated into SPICE3 using the Volterra series technique applied to the MNA formulation of a circuit's equations. Volterra series are a natural choice for analyzing harmonic and intermodulation distortion in a circuit simulator with an existing routine for small-signal a.c. analysis. The nonlinearities of each device supported by SPICE3 are presented. Conclusions and comparisons of numerical results from *.disto* and other methods are made.

Contents

Table of Contents 2

List of Figures 4

List of Tables 6

Acknowledgments 7

1 Introduction 8

2 Volterra Series Analysis 12
- 2.1 Volterra Series 12
- 2.2 Application of Volterra series to a circuit 13
  - 2.2.1 Introduction 13
  - 2.2.2 Matrix Formulation 14
- 2.3 Summary 18

3 Distortion Generators in Devices 20
- 3.1 The DIOde model 20
- 3.2 The BJT model 22
- 3.3 The JFET and MESFET models 24
- 3.4 The MOS models 27

4 Results 29

5 Conclusion and Observations 32

Bibliography 35

A Volterra expansions for useful inputs 37
- A.1 Input = $e^{j\omega t}$ 37
- A.2 Input = $e^{j\omega_1 t} + e^{j\omega_2 t}$ 37
- A.3 Input = $A\cos(\omega t + \phi_1)$ 37
- A.4 Input = $A\cos(\omega_1 t + \phi_1) + B\cos(\omega_2 t + \phi_2)$ 38

B Distortion Source Formulae 39
B.1 Distortion source at 2ω . . . 39
B.2 Distortion source at 3ω . . . 40
B.3 Distortion source at $\omega_1 + \omega_2$ . . . 40
B.4 Distortion source at $2\omega_1 + \omega_2$ . . . 41

C How to use SPICE3 distortion analysis 42

D Results 44
D.1 Diode-Resistor circuit . . . 45
D.2 BJT differential pair circuit . . . 52
D.3 MOSFET differential pair circuit . . . 59
D.4 JFET single stage circuit . . . 67
D.5 JFET differential pair circuit . . . 74
D.6 MESFET single-stage circuit . . . 81
D.7 ua741 op-amp . . . 89
D.8 ua733 wideband amplifier circuit . . . 97
D.9 MOSAMP2 mos amplifier . . . 105
D.10 Iee modulated Mixer . . . 112
D.11 Double Balanced Mixer . . . 119

List of Figures

3.1 DIODE model 21
3.2 BJT model 23
3.3 JFET model 26
3.4 MOS model 28

D.1 Diode-Resistor circuit - 2nd harmonic 47
D.2 Diode-Resistor circuit - 3rd harmonic 48
D.3 Diode-Resistor circuit - IM : $f_1 + f_2$ 49
D.4 Diode-Resistor circuit - IM : $f_1 - f_2$ 50
D.5 Diode-Resistor circuit - IM : $2f_1 - f_2$ 51
D.6 BJT differential pair circuit - 2nd harmonic 54
D.7 BJT differential pair circuit - 3rd harmonic 55
D.8 BJT differential pair circuit - IM : $f_1 + f_2$ 56
D.9 BJT differential pair circuit - IM : $f_1 - f_2$ 57
D.10 BJT differential pair circuit - IM : $2f_1 - f_2$ 58
D.11 MOS differential pair circuit - ac analysis 61
D.12 MOS differential pair circuit - 2nd harmonic 62
D.13 MOS differential pair circuit - 3rd harmonic 63
D.14 MOS differential pair circuit - IM : $f_1 + f_2$ 64
D.15 MOS differential pair circuit - IM : $f_1 - f_2$ 65
D.16 MOS differential pair circuit - IM : $2f_1 - f_2$ 66
D.17 JFET single-stage circuit - 2nd harmonic 69
D.18 JFET single-stage circuit - 3rd harmonic 70
D.19 JFET single-stage circuit - IM : $f_1 + f_2$ 71
D.20 JFET single-stage circuit - IM : $f_1 - f_2$ 72
D.21 JFET single-stage circuit - IM : $2f_1 - f_2$ 73
D.22 JFET differential pair circuit - 2nd harmonic 76
D.23 JFET differential pair circuit - 3rd harmonic 77
D.24 JFET differential pair circuit - IM : $f_1 + f_2$ 78
D.25 JFET differential pair circuit - IM : $f_1 - f_2$ 79
D.26 JFET differential pair circuit - IM : $2f_1 - f_2$ 80
D.27 MESFET single-stage circuit - ac analysis 83
D.28 MESFET single-stage circuit - 2nd harmonic 84

D.29 MESFET single-stage circuit - 3rd harmonic 85
D.30 MESFET single-stage circuit - IM : $f_1 + f_2$ 86
D.31 MESFET single-stage circuit - IM : $f_1 - f_2$ 87
D.32 MESFET single-stage circuit - IM : $2f_1 - f_2$ 88
D.33 ua741 op-amp - 2nd harmonic 92
D.34 ua741 op-amp - 3rd harmonic 93
D.35 ua741 op-amp - IM : $f_1 + f_2$ 94
D.36 ua741 op-amp - IM : $f_1 - f_2$ 95
D.37 ua741 op-amp - IM : $2f_1 - f_2$ 96
D.38 ua733 wideband amplifier circuit - 2nd harmonic 100
D.39 ua733 wideband amplifier circuit - 3rd harmonic 101
D.40 ua733 wideband amplifier circuit - IM : $f_1 + f_2$ 102
D.41 ua733 wideband amplifier circuit - IM : $f_1 - f_2$ 103
D.42 ua733 wideband amplifier circuit - IM : $2f_1 - f_2$ 104
D.43 MOSAMP2 mos amplifier - 2nd harmonic 107
D.44 MOSAMP2 mos amplifier - 3rd harmonic 108
D.45 MOSAMP2 mos amplifier - IM : $f_1 + f_2$ 109
D.46 MOSAMP2 mos amplifier - IM : $f_1 - f_2$ 110
D.47 MOSAMP2 mos amplifier - IM : $2f_1 - f_2$ 111
D.48 Iee modulated Mixer - 2nd harmonic 114
D.49 Iee modulated Mixer - 3rd harmonic 115
D.50 Iee modulated Mixer - IM : $f_1 + f_2$ 116
D.51 Iee modulated Mixer - IM : $f_1 - f_2$ 117
D.52 Iee modulated Mixer - IM : $2f_1 - f_2$ 118
D.53 Double Balanced Mixer - 2nd harmonic 122
D.54 Double Balanced Mixer - 3rd harmonic 123
D.55 Double Balanced Mixer - IM : $f_1 + f_2$ 124
D.56 Double Balanced Mixer - IM : $f_1 - f_2$ 125
D.57 Double Balanced Mixer - IM : $2f_1 - f_2$ 126

List of Tables

4.1 Comparison of SPICE3 *.disto* with other methods 30

Acknowledgements

I would like to thank Profs. Richard Newton, Donald Pederson and Robert Meyer for the advice and inspiration they provided me. The extension of *.disto* to handle multiple inputs is based on the ideas of Prof. Donald Pederson. I would also like to express my appreciation to Tom Quarles, Pranav Ashar, Umakanta Choudhury and Wayne Christopher for their help and suggestions. This work was supported in part by Raytheon, the California State MICRO program and by the Defense Advanced Research Projects Agency. Computing equipment used was provided by the Digital Equipment Corporation. Their support is gratefully acknowledged.

Discussions with Gerry Marino, Raytheon, were also helpful in this work and his continued encouragement of our circuit simulation work is much appreciated.

Chapter 1

Introduction

Nonlinear distortion is a critical issue in the design of long-distance frequency division multiplexed analog communication systems. The distortion exists because the components of any electronic system are not linear as they should ideally be. The nonlinearities stem from the inherently nonlinear constitutive relationships of the active devices. As the bandwidth of a communication system increases, the number of intermodulation products increases. These intermodulation products degrade the SNR of the signal. In order to meet overall system noise specifications, stringent requirements are set on the degree of linearity of any system. In addition to intermodulation, nonlinearities cause cross-modulation, desensitization, spurious responses, and gain compression/expansion. It is important, therefore, to take distortion into account during the design of a circuit.

In practice, the distortions are much smaller than the fundamental signals - for example, it is common for the third order intermodulation product to be 90dB below the fundamental. Therefore, the calculation of these distortions require precise modelling of the active devices, as well as an appropriate mathematical tool to extract the small distortions from the large signal.

Several methods have been used for calculating distortion.

- **Power Series:** A simple approach is the power series method [WS80]. The fundamental assumption here is that the circuit to be analysed consists of only *zero-memory nonlinearities* , preceded and followed by *linear* circuits which may have memory. The nonlinearities are then expressed in power-series form for the analysis. Although this model appears very limited, there do exist many physical systems for which it is an adequate representation; this method was popular with vacuum-tube circuits, for which the assumption is valid. However, the assumption breaks down when one deals with semiconductor devices, for example the BJT, in which there are important nonlinear charge storage effects.

- **Transient-Fourier:** Another method, one that is commonly used, is to do a transient analysis of the circuit with a sinusoidal input, followed by a Fourier analysis of the output to determine the spectral components arising from harmonics and intermodulation. This method, though quite general, has some disadvantages. Firstly, the method is inaccurate for small (as is typically the case) distortion levels. This is because the transient and Fourier analyses are done with the much larger signal level in consideration, for which the tolerances are typically larger than the values of the distortion components, which get washed out. Secondly, any transient analysis produces a transient response superimposed on the desired steady-state response. The Fourier components of the transient response add onto the components of the steady-state response making the result inaccurate. Thirdly, transient analyses are expensive in terms of computation, and may have convergence problems. This is compounded by the fact that a reasonably long transient analysis is required in order that a subsequent Fourier analysis yield meaningful results. For large distortions, however, the first disadvantage does not apply, and this method is superior.

- **Harmonic Balance:** A recent approach to calculating distortions is the harmonic balance method, as implemented in the program SPECTRE [KS86,KSS88,Kun87]. SPECTRE finds the periodic steady state of a circuit with sinusoidal inputs. The technique consists of expressing each circuit variable in a series of orthogonal eigenfunctions, the coefficients of which are calculated by the program. In SPECTRE the eigenfunctions used are sinusoidal waves at the frequencies of all possible harmonics and intermodulation frequencies. The magnitudes of the calculated coefficients equal the distortion components.

  This technique is very accurate for both large and small distortion levels in the circuit; it can also handle several inputs. However, it is not very suitable for inclusion in SPICE3 because it operates in the frequency domain and cannot take advantage of existing routines in SPICE3. Also, the order of complexity of the method is greater than linear with respect to circuit size; and there exists the possibility of convergence problems, though these are usually not significant.

- **Volterra Series:** The Volterra series method [Nar67,Kuo73,WS80] is a general-purpose method which yields accurate results when applied to any circuit with *small* distortion levels. It is not suitable for large distortions (as explained later), but in view of the preceding discussion, this is not a limitation in typical applications. The technique is particularly suited for implementation in a circuit simulator that does small-signal ac analyses. The computation time required is of the linear w.r.t. the size of the circuit, and there is no convergence involved. These factors make it a desirable method for doing distortion analysis in SPICE3. Details of the Volterra series method are given in Chapter 2.[1]

  The Volterra series method has been used in other simulators for distortion analysis. NODAP (NOnlinear Distortion Analysis Program) [Kuo73] is one such program. NODAP was written specifically for distortion analysis, but does not support a wide range of devices. Distortion analysis using Volterra series is also available in the very widely used SPICE2. SPICE2 distortion analysis has serious disadvantages, however, that have made it virtually unusable [CN73]. The code used in SPICE2 was transferred from SPICE1; also, as new models were added to SPICE2 or as existing models were updated, the distortion code was not modified so that errors exist in the code that yield wrong results. SPICE2 distortion supports only the diode and BJT models and, in addition, the distortion code for the BJT implements only the basic exponential nonlinearity, not making full use of the complexity of the Gummel-Poon integral charge control model.

  SPICE3 distortion analysis has the following features:

  - It implements all the semiconductor devices available in SPICE3 - DIODEs, BJTs, JFETs, MOS (levels 1, 2, 3 and BSIM) and MESFETs. This makes the routine applicable to a wide range of circuits - for example, BIMOS circuits[2].

[1] The Volterra series technique is equivalent to an intuitive circuit-based approach [Ped87] for which it provides a mathematical framework. Derivations using perturbation theory [Kuo73] also yield the same results.

[2] The only nonlinear device that cannot be implemented is the switch - because the switch is an abrupt (strong) nonlinearity, not amenable to analysis by the Volterra series technique. Distortion analysis for a circuit containing switches will yield accurate results, however, if the (small) signal levels used do not cause the switches to change state.

- Multiple small-signal sinusoidal input sources can be handled, making the program useful in the design and analysis of mixer circuits.
- The equations used for defining a device for distortion are taken from the device's *load* routine code, ensuring compatibility with other SPICE3 analyses. All nonlinearities of the device are implemented. Each internal current or charge of the device is expressed as a function of upto three controlling voltages. For example, in a MOS device, the drain current I_D is a nonlinear function of V_{GS}, V_{DS} and V_{SB}.
- Like the rest of SPICE3, the code is highly modular[Qua89b]. This is especially important in distortion analysis because the expressions for distortion formulae are long and convoluted, increasing the possibility of coding errors. The modular structure also makes the addition of new devices (and the modification of existing ones) possible with very little change and effort.

The syntax for the new distortion command of SPICE3 is given in Appendix C.

Chapter 2

Volterra Series Analysis

Volterra series were first proposed by Volterra around 1910[Vol59]. Since then, an enormous amount of work has been undertaken on its theory and applications [WS80, page 81]. This chapter deals with the simple case of Volterra series as applied to a circuit with one input, which may be either a single sinusoid or a sum of two sinusoids of different frequencies. Moreover, all Volterra kernels and transforms are assumed symmetric, and it is the symmetrised transform that all the $H_k(\cdot, \ldots, \cdot)$ refer to.

2.1 Volterra Series

Given a system with input $x(t)$ and output $y(t)$, where $x(t)$ is a sinusoidal small-signal input in the present context, $y(t)$ can be represented as:

$$y(t) = y_1(t) + y_2(t) + y_3(t) + \cdots \tag{2.1}$$

where:

$$y_n(t) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \ldots, \tau_n)\, x(t - \tau_1) \cdots x(t - \tau_n)\, d\tau_1 \cdots d\tau_n \tag{2.2}$$

Equation 2.1 is called a Volterra series expansion for the output $y(t)$.
If $x(t)$ is a pure exponential, i.e., $x(t) = e^{j\omega t}$, then

$$y_n(t) = e^{jn\omega t} \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \ldots, \tau_n)\, e^{-j\omega(\tau_1 + \cdots + \tau_n)}\, d\tau_1 \cdots d\tau_n \tag{2.3}$$

An expression of the general form:

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h_n(\tau_1, \ldots, \tau_n)\ e^{-j(\omega_1 \tau_1 + \cdots + \omega_n \tau_n)}\, d\tau_1 \cdots d\tau_n \tag{2.4}$$

is called a **Volterra transform** of the n^{th} degree and is represented by $H_n(\omega_1, \ldots, \omega_n)$. Then, Equation 2.3 becomes:

$$y_n(t) = e^{jn\omega t}\, H_n(\omega, \ldots, \omega) \tag{2.5}$$

It can be seen that for a simple exponential input, an n^{th} degree Volterra term $y_n(t)$ has a frequency of $n\omega$ and amplitude $|H_n(\omega, \ldots, \omega)|$.

If the input $x(t)$ is a sum of exponential terms, i.e., $x(t) = E_1\, e^{j\omega_1 t} + \cdots + E_Q\, e^{j\omega_Q t}$, then application of Equation 2.2 yields (not derived here, but may be found in [WS80]):

$$y_n(t) = \frac{1}{2^n} \sum_{q_1=1}^{Q} \cdots \sum_{q_n=1}^{Q} E_{q_1} \cdots E_{q_n}\, H_n(\omega_{q_1}, \ldots, \omega_{q_n})\, e^{j(\omega_{q_1} + \cdots + \omega_{q_n})} \tag{2.6}$$

Equation 2.6 can be used to derive the expression for $y_n(t)$ if the input is a sum of sinusoids, by expressing each sinusoid as a sum of exponentials. The expressions for $y_n(t)$ when $x(t)$ is a simple sinusoid, and a sum of two sinusoids, are given in Appendix A.

2.2 Application of Volterra series to a circuit

2.2.1 Introduction

As can be seen from Equation 2.1, the Volterra series is a functional series in terms of an input $x(t)$. In applying the technique to a circuit, an input must be identified, in terms of which the output(s) may be expressed in a Volterra series. Given any circuit being analysed, it is meaningful to express all unknowns in Volterra series about the input signal[1].

It is convenient to consider all the *unknown circuit variables used for modified nodal analysis of a circuit as the outputs for a Volterra series analysis.* Thus each *node voltage* and each *branch current of each voltage source and current-controlled element* is considered to be an output of the circuit and, *for each of these outputs, a Volterra series is*

[1] A circuit may have more than one input signal, as in a mixer - such circuits can be dealt with using vector Volterra series. The following sections assume a single input for purposes of simplicity

written in terms of the input. Since the series is an infinite one, it must be truncated to a finite length for any practical implementation; in this work, the first three terms are used. In most practical applications in weakly nonlinear circuits, it is found that three terms give sufficiently accurate results for small signal levels. The form of the series for any output is given in Appendix A for the cases of the input being a single sinusoid and a sum of two sinusoids. The object of the analysis is to determine the *Volterra transforms for each output variable.* Once these are known, the formulae in Appendix A can be used to obtain the magnitude and phase of each frequency component for each output and the strengths of the harmonics and cross-frequencies are thus determined.

2.2.2 Matrix Formulation

The Volterra transforms appearing in the series expansions are obtained in the following manner. First, the DC operating point of the circuit is obtained after setting the small-signal source $x(t) = 0$. Each nonlinear branch relationship in the circuit is then expanded in a Taylor series about the operating point values of its controlling variable(s). The following examples illustrate the procedure for different types of devices.

In a simple two-terminal device like the diode, there is only one controlling variable, the voltage across the diode V_d, and the branch relationship is $I_d = f(V_d)$. When this is expanded in a Taylor series about the operating point (I_{d0}, V_{d0}), the branch relationship becomes:

$$I_{d0} + i_d = f(V_{d0}) + \left.\frac{\partial f}{\partial V_d}\right|_{V_{d0}} \cdot v_d + \frac{1}{2!}\left.\frac{\partial^2 f}{\partial V_d^2}\right|_{V_{d0}} \cdot v_d^2 + \frac{1}{3!}\left.\frac{\partial^3 f}{\partial V_d^3}\right|_{V_{d0}} \cdot v_d^3 + \cdots \quad (2.7)$$

where i_d and v_d are the small-signal excursions of I_d and V_d about I_{d0} and V_{d0} respectively. Since $I_{d0} = f(V_{d0})$, we have

$$i_d = g_1 \cdot v_d + g_2 \cdot v_d^2 + g_3 \cdot v_d^3 \quad (2.8)$$

where g_1, g_2, and g_3 are the constant coefficients of v_d, v_d^2 and v_d^3 in Equation 2.7. The series is truncated to three terms here. It is important to note that the same Taylor expansion is used when the circuit is linearized for an a.c. analysis; only the first term is kept, and g_1 becomes the small-signal conductance of the nonlinear element.

As another example, consider a simple two-terminal nonlinear voltage-controlled capacitor, the branch relationship is $Q_c = f(V_c)$. This is expanded in a Taylor series to give

$$q_b = c_1 \cdot v_c + c_2 \cdot v_c^2 + c_3 \cdot v_c^3 \tag{2.9}$$

Similarly, simple two-terminal nonlinear current-controlled inductors will have a Taylor series expansion of the flux in terms of the controlling current.

If the device is an *n*-terminal device, its branch relationships may have upto *n-1* controlling variables. For example, in a MOSFET, I_D is a function of V_{GS}, V_{DS} and V_{SB}. In such cases, the branch relationship is expanded as a Taylor series in all its controlling variables. This implementation supports upto 3 controlling variables for each branch relationship, so any 4-terminal device can be supported.

Now consider the MNA equations of the linearized circuit, given by:

$$\mathbf{A}(\mathbf{p})\,\mathbf{x} = \mathbf{I_S} \tag{2.10}$$

Here $\mathbf{A}(\mathbf{p})$ is the MNA admittance matrix, and will contain the differentiation operator p $= \frac{\partial}{\partial t}$ if there are energy-storage elements in the circuit. $\mathbf{x}$ is the vector of unknown circuit variables to be solved for, i.e.:

$$\mathbf{x} = \begin{bmatrix} v_1 \\ \vdots \\ v_k \\ i_{k+1} \\ \vdots \\ i_n \end{bmatrix}$$

and $\mathbf{I_S}$ is the known vector of sources in the circuit.

When the second and the third Taylor terms are included in the branch relationship, Equation 2.10 is no longer valid. The additional square and cube terms *add* to the linear term in the Taylor-expanded branch equations 2.8 and 2.9. Suppose the controlled variable is a current from node *l*. The l^{th} row of $\mathbf{A}(\mathbf{p})$ and the l^{th} element of $\mathbf{I_S}$ comprise the node equation for the l^{th} node. The node equation can be modified to account for the extra current i_l^{non} specified by the nonlinear terms by adding i_l^{non} to the l^{th} term of $\mathbf{I_S}$. This is done for all the nonlinear terms and all the branch relationships. As a result, a vector of nonlinear terms $\mathbf{I_{non}}$ adds to the RHS of equation 2.10, and the MNA equations can be written as [2]:

$$\mathbf{A}(\mathbf{p})\,\mathbf{x} = \mathbf{I_S} + \mathbf{I_{non}} \tag{2.11}$$

[2] If the controlled variable is the voltage across some element, an additive term v^{non} appears in the branch relationship for the element. If this branch relationship is the r^{th} row of $\mathbf{A}(\mathbf{p})$, then v^{non} appears in the r^{th}

It is to be noted that Equation 2.11 cannot be solved as it stands, because $\mathbf{I_{non}}$ is not a known vector, but is a function of the branch relationship controlling variables. Each controlling variable is a simple *linear* function of one or more unknown variables[3]. For example, if a diode is connected across nodes l and m, the controlling variable is $v_d = v_l - v_m$; or, a controlling variable x may be a delayed version of an unknown variable y, i.e., $x(t) = y(t - \tau)$. Thus $\mathbf{I_{non}}$ is a function of the unknown vector, and so Equation 2.11 cannot be solved immediately.

To solve the equation, each unknown variable is expressed as its Volterra series. To illustrate the technique, the input is assumed to be $e^{j\omega t}$. Then the unknown vector $\mathbf{x} = [v_1, \ldots, v_k, i_{k+1}, \ldots, i_n]^{\mathrm{T}}$ can be written as the sum of three vectors of (as yet unknown) Volterra transforms:

$$\mathbf{x} = \mathbf{H_1} e^{j\omega t} + \mathbf{H_2} e^{j2\omega t} + \mathbf{H_3} e^{j3\omega t} \tag{2.12}$$

where:

$$\mathbf{H_i} = \begin{bmatrix} H_i^{v_1}(\omega, \cdots, \omega) \\ \vdots \\ H_i^{v_k}(\omega, \cdots, \omega) \\ H_i^{i_{k+1}}(\omega, \cdots, \omega) \\ \vdots \\ H_i^{i_n}(\omega, \cdots, \omega) \end{bmatrix} \quad \text{for } i = 1, 2, 3$$

Then, using the fact that $\mathbf{A(p)}$ is a linear operator, the LHS of Equation 2.11 becomes:

$$\mathbf{A(p)}\, \mathbf{H_1} e^{j\omega t} + \mathbf{A(p)}\, \mathbf{H_2}\, e^{j2\omega t} + \mathbf{A(p)}\, \mathbf{H_3}\, e^{j3\omega t} \tag{2.13}$$

On the RHS, the known vector $\mathbf{I_S}$ has entries of only the form $e^{j\omega t}$ and $-e^{j\omega t}$ since the single input is $e^{j\omega t}$. Therefore:

$$\mathbf{I_S} = \mathbf{I'_S}\, e^{j\omega t} \tag{2.14}$$

where $\mathbf{I'_S}$ is a vector containing only $1'$s, $-1'$s and $0'$s.

---

position of $\mathbf{I_{non}}$. If the controlled variable is a charge, and the nonlinear additive term is q_l^{non} in the l^{th} equation, then $\frac{\partial q_l^{non}}{\partial t}$ appears in the l^{th} position of $\mathbf{I_{non}}$. Similarly for fluxes.

[3]Henceforth, "unknown variable" stands for any variable in the unknown vector $[v_1, \ldots, v_k, i_{k+1}, \ldots, i_n]^{\mathrm{T}}$, and "controlling variable" stands for any variable that occurs on the RHS of a nonlinear branch relationship such as Equation 2.8 or Equation 2.9

Each controlling variable can also be expressed in a Volterra series in the input $e^{j\omega t}$. Because each controlling variable is a *linear* function of unknown variables, the Volterra transform of the controlling variable is a linear function of transforms of the same order of the unknown variables, i.e.:

$$H_n^x(\cdot,\ldots,\cdot) = k_1 H_n^{y_{i_1}}(\cdot,\ldots,\cdot) + \cdots + k_n H_n^{y_{i_j}}(\cdot,\ldots,\cdot) \tag{2.15}$$

where x is a controlling variable, the y's are unknown variables, and the k's are constants. Then, in the Volterra expansion for any controlling variable x (Appendix A),

$$x = H_1^x(\omega)\, e^{j\omega t} + H_2^x(\omega,\omega)\, e^{j2\omega t} + H_3^x(\omega,\omega,\omega)\, e^{j3\omega t} \tag{2.16}$$

each H_n^x *depends only on the n_{th} order transforms of the unknown variables.*

Consider the nonlinear vector $\mathbf{I_{non}}$ with all controlling variables expressed in Volterra series form. Each element in $\mathbf{I_{non}}$ is a polynomial in (some of the) controlling variables. Moreover, *the polynomial contains no constant and linear terms*; only square and cube terms of the controlling variables are present, and the coefficients of these terms are the Taylor coefficients at the operating point.

With any controlling variable x expressed as in Equation 2.16 , the expression for x^2 becomes:

$$\begin{aligned} x^2 &= [H_1^x(\omega)]^2 e^{j2\omega t} + [H_2^x(\omega,\omega)]^2 e^{j4\omega t} + [H_3^x(\omega,\omega,\omega)]^2 e^{j6\omega t} \\ &\quad + 2H_1^x(\omega)H_2^x(\omega,\omega)e^{j3\omega t} + 2H_2^x(\omega,\omega)H_3^x(\omega,\omega,\omega)e^{j5\omega t} \\ &\quad + 2H_1^x(\omega)H_3^x(\omega,\omega,\omega)e^{j4\omega t} \end{aligned}$$

and the expression for x^3 becomes:

$$x^3 = [H_1^x(\omega)]^3 e^{j3\omega t} + \text{higher order terms in } e^{j\omega t}$$

The nonlinear vector $\mathbf{I_{non}}$ can then be expressed as:

$$\mathbf{I_{non}} = \mathbf{N_2} e^{j2\omega t} + \mathbf{N_3} e^{j3\omega t} + \text{higher order terms in } e^{j\omega t} \tag{2.17}$$

where $\mathbf{N_2}$ and $\mathbf{N_3}$ are functions of the Volterra transforms of the controlling variables and the Taylor coefficients. Equation 2.17 has two important features:

- There are *no constant and no $e^{j\omega t}$ terms.*

- The elements of $\mathbf{N_i}$, for $i = 2, 3$, are *functions of Volterra transforms of order at most* $i - 1$.

Combining equations 2.14 and 2.17 and truncating, the RHS of Equation 2.11 becomes:

$$\mathbf{I'_S} e^{j\omega t} + \mathbf{N_2} e^{j2\omega t} + \mathbf{N_3} e^{j3\omega t} \tag{2.18}$$

Equating 2.13 and 2.18, and noting that the functions $\{e^{jn\omega t}\}$, $n = 1,2,3$ are linearly independent, the following system of equations is obtained:

$$\begin{aligned} \mathbf{A}(j\omega)\mathbf{H_1} &= \mathbf{I'_S} && (2.19) \\ \mathbf{A}(j\omega)\mathbf{H_2} &= \mathbf{N_2} && (2.20) \\ \mathbf{A}(j\omega)\mathbf{H_3} &= \mathbf{N_3} && (2.21) \end{aligned}$$

where $\mathbf{A(p)}$ has been replaced by $\mathbf{A}(j\omega)$ because $p(e^{j\omega t}) = j\omega\, e^{j\omega t}$.

Note that Equation 2.19 is simply the linearized equation for the circuit; therefore $\mathbf{H_1}$ consists of simply the s.s. linear a.c. values of the circuit variables. Once $\mathbf{H_1}$ is calculated using Equation 2.19, $\mathbf{N_2}$ can be calculated because it depends only on the Volterra transforms of order 1 which are available from $\mathbf{H_1}$. Once $\mathbf{N_2}$ is known, another small-signal analysis of the linearized circuit, this time with the source vector equal to $\mathbf{N_2}$, will yield the second order transforms $\mathbf{H_2}$. Now $\mathbf{H_1}$ and $\mathbf{H_2}$ are known, so $\mathbf{N_3}$ can be calculated, since it depends only of transforms of order upto 2. With $\mathbf{N_3}$ known, another small-signal linear analysis gives $\mathbf{H_3}$.

2.3 Summary

From the above, the procedure for computing the Volterra transforms is seen to consist of iterations of the following:

1. A.C. analyses of a *linear* circuit with different source excitations, to obtain Volterra transforms of different orders.

2. Computation of the "distortion sources" $\mathbf{N_2}$, $\mathbf{N_3}$ from previously calculated Volterra transforms, to be used in (1).

The following summarises the procedure for calculating the Volterra kernels in the case of a circuit driven by a single input source of one frequency:

1. Linearise the circuit. Set the a.c. value of the input source to 1 (regardless of the actual value of the source). Solve the linearized circuit with this input (equivalent to an a.c. analysis); the resulting values of the circuit variables are equal to $H_1^x(\omega)$.

2. Use the formulae in Appendix B and the values of $H_1^x(\omega)$ to calculate the values of the distortion sources at 2ω. Using these distortion sources as excitation to the linearized circuit (and setting the value of the original input to zero), solve the linear circuit to obtain the values of $H_2^x(\omega,\omega)$.

3. Repeat step 2 for 3ω, using both $H_1^x(\omega)$ and $H_2^x(\omega,\omega)$ in the distortion sources for 3ω.

Once the Volterra transforms are computed, the formulae in Appendix A can be used to calculate the a.c. values of the different frequency components of each circuit variable and so the strengths of its harmonics and cross-frequencies can be determined.

The above procedure illustrates how to obtain the kernels $H_1^x(\omega)$, $H_2^x(\omega,\omega)$, and $H_3^x(\omega,\omega,\omega)$; for kerriels like $H_2^x(\omega_1,\omega_2)$ and $H_3^x(\omega_1,\omega_1,\omega_2)$, the input to the circuit must be taken to be the sum of two exponentials $e^{j\omega_1 t}+e^{j\omega_2 t}$ or $e^{j\omega_1 t}+e^{-j\omega_2 t}$. By following a procedure analogous to the one above, these other kernels can be obtained.

In order to analyse the case where there are several inputs to the circuit, it is necessary to use vector Volterra series. The result of the analysis corroborates what one might intuitively expect: that in the initial step when the linear circuit is driven by the source, one should use all the inputs to drive the circuit simultaneously. In this case, however, it is convenient to keep the values of the driving signal as they actually are rather than normalise them to 1; this saves additional computation after the determination of the kernels.

Because of the iterative structure of the procedure, using only linear analyses of the circuit and evaluations of the distortion sources, the Volterra series technique is ideally suited for implementation in a simulator that already supports linear small-signal analysis.

Chapter 3

Distortion Generators in Devices

It is seen in Chapter 2 that the devices relevant to distortion analysis are the ones that have nonlinearities, i.e., their Taylor expansions should have superlinear terms. SPICE3 supports the following SPICE3 device models: BJT, BSIM, CAP, CCCS, CCVS, CSW, DIO, IND, ISRC, JFET, MES, MOS1, MOS2, MOS3, RES, SW, TRA, URC, VCCS, VCVS, and VSRC[Qua89a].

Of these, the devices of interest for distortion analysis are the diode, BJT and FET (JFET, MOS1, MOS2, MOS3, BSIM, MESFET) models. Though a switch is a nonlinear device, it is linear at every point except the switching threshold, where it has an abrupt nonlinearity that does not have a Taylor expansion. Therefore, switches cannot be analysed by the Volterra series technique if the operating point value of the controlling variable is close enough to the switching threshold for the switch to flip in a small-signal analysis. Under normal conditions, this is not true; a switch is linear, and needs no special consideration. Also, uniform R-C transmission lines (URCs) are broken up into resistors, capacitors and (optionally) diodes for simulation and do not need to be considered separately.

In the following sections, the distortion-generating relationships in the device models are presented, giving the identities of controlling variables for each nonlinear equation. The expressions for the relationships are not given.

3.1 The DIOde model

The diode model (Figure 3.1) has the following nonlinearities:

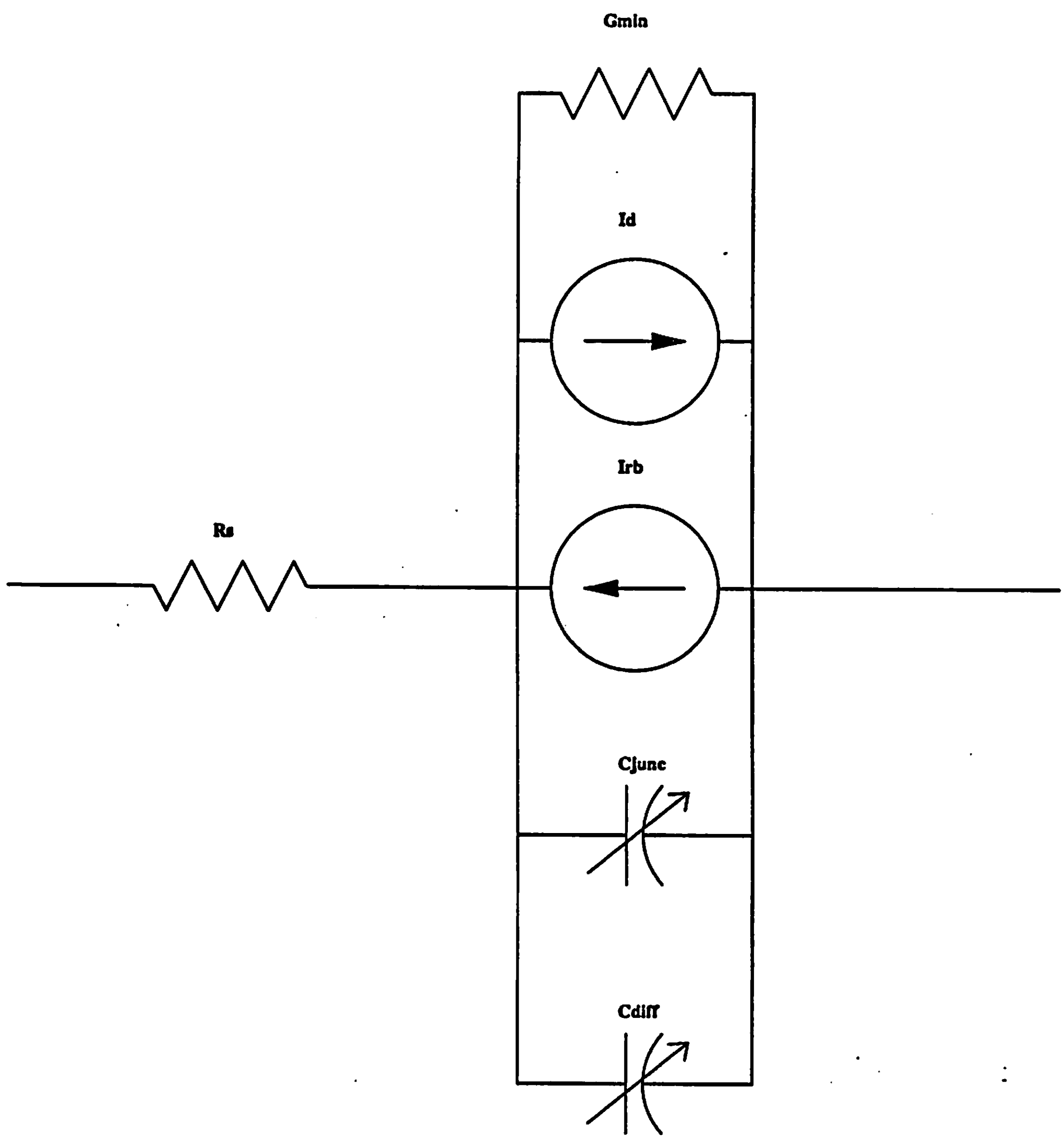

Figure 3.1: DIODE model

- The forward region memoryless exponential nonlinearity:

$$I_D = f(V_D)$$

- The reverse breakdown nonlinearity, also memoryless and exponential:

$$I_{RB} = f(V_D)$$

- The nonlinear voltage-controlled stored charge:

$$Q_D = Q_{CJ} + Q_{CD} = f(V_D)$$

There is only one controlling variable V_D for each equation.

3.2 The BJT model

BJTs (Figure 3.2) can be either NPN or PNP. In order to use a single set of equations to describe both kinds of BJTs, the variables in the following equations are denoted hatted ($\hat{}$). If the device is NPN, then $\hat{x} = x$, where x is any variable in Figure 3.2. If the device is PNP, then $\hat{x} = -x$.[1]

The BJT nonlinearities are:

- The g_m generator:

$$\hat{I}_C = f(\hat{V}_{BE}^{del}, \hat{V}_{BC}, \hat{Q}_B)$$

where

$$\hat{Q}_B = f(\hat{V}_{BE}, \hat{V}_{BC})$$

Therefore,

$$\hat{I}_C = f(\hat{V}_{BE}^{del}, \hat{V}_{BC}, \hat{V}_{BE})$$

$\hat{V}_{BE}^{del}$ is $\hat{V}_{BE}$ delayed by a time τ_d; this accounts for the linear phase shift in the g_m generator. The Volterra transforms for $\hat{V}_{BE}^{del}$ can be easily shown to be

$$H_n^{\hat{V}_{BE}^{del}}(\omega_1, \ldots, \omega_n) = e^{-j\tau_d(\omega_1 + \cdots + \omega_n)} H_n^{\hat{V}_{BE}}(\omega_1, \ldots, \omega_n)$$

[1]This mapping of variables is significant because it changes the signs of the coefficients of the square terms in the Taylor series for PNP devices.

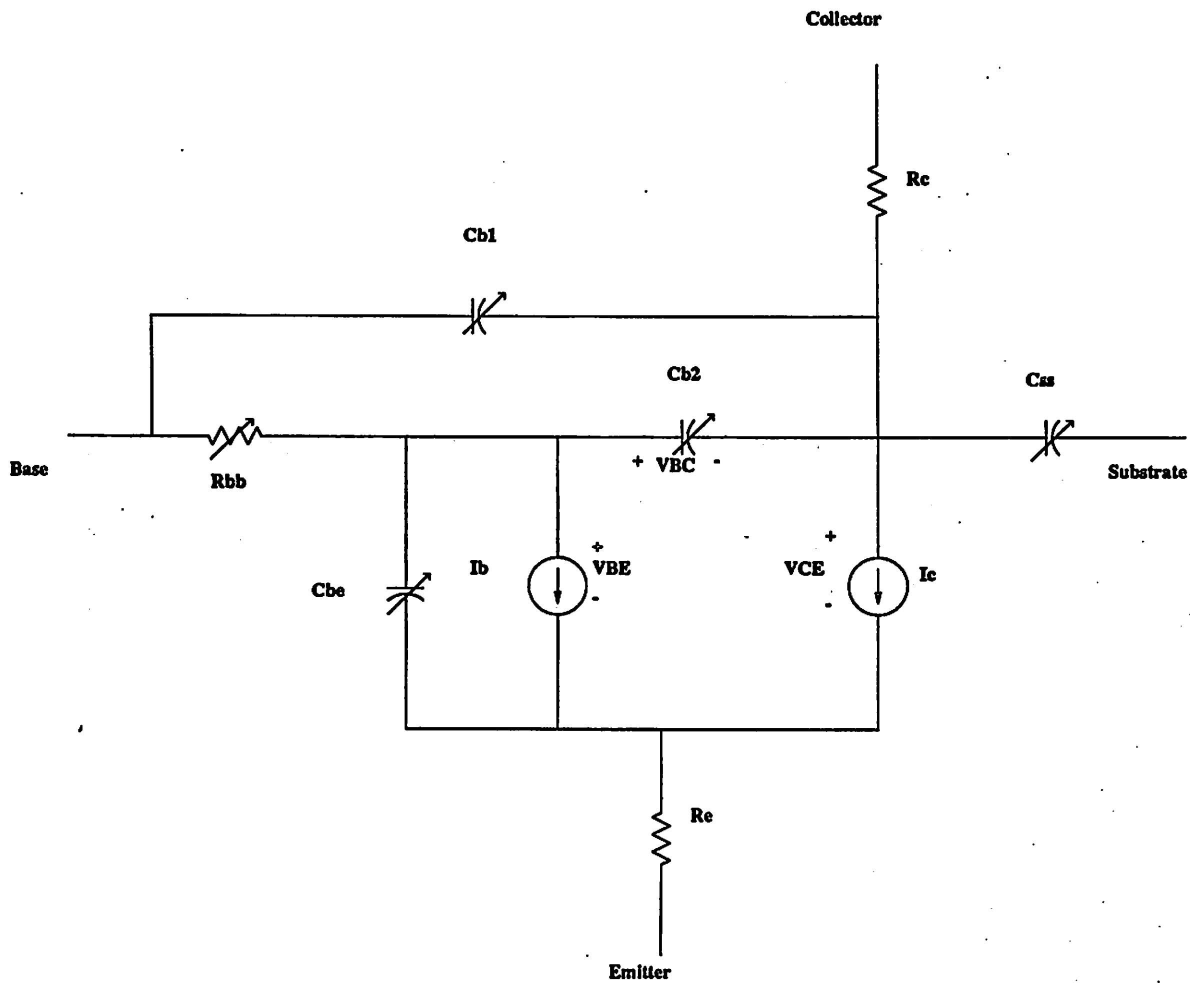

Figure 3.2: BJT model

- The base current:

$$\hat{I}_B = f(\hat{V}_{BE}, \hat{V}_{BC})$$

- The base resistance:

$$\begin{aligned}\hat{I}_{BB} &= \frac{\hat{V}_{BB}}{\hat{R}_{BB}} \\ &= f(\hat{V}_{BB}, \hat{V}_{BE}, \hat{V}_{BC}) \text{ if } J_{RB} \text{ given} \\ &= f(\hat{V}_{BB}, \hat{I}_{BB}) \text{ otherwise}\end{aligned}$$

In the case where $\hat{I}_{BB}$ is an implicit function, a series inversion is used to express $\hat{I}_{BB}$ as a Taylor series in $\hat{V}_{BB}$.

- The base emitter stored charge:

$$\hat{Q}_{BE} = f(\hat{V}_{BE}, \hat{V}_{BC}, \hat{Q}_B) = f(\hat{V}_{BE}, \hat{V}_{BC})$$

- The base collector stored charges:

$$\hat{Q}_{B1}\,,\,\hat{Q}_{B2} = const \cdot \hat{Q}_{BC}$$

$$\hat{Q}_{BC} = f(\hat{V}_{BC})$$

- The collector substrate stored charge:

$$\hat{Q}_{SS} = f(\hat{V}_{CS})$$

3.3 The JFET and MESFET models

In writing the expressions for FETs, it is convenient to write a single set of equations in primed ($'$) variables. In order to obtain the actual circuit variables in Figure 3.3 from the primed variable and vice-versa, the hatted variables are first obtained using $x = \hat{x}$ for N-channel devices and $x = -\hat{x}$ for P-channel devices as in Section 3.2. The relationship between the primed variables and the hatted variables is as follows:

1. If the device is in forward mode, i.e., $\hat{V}_{DS} \geq 0$, then $\hat{x}_{ab} = x'_{ab}$, where x can be any of V, I and Q, and a, b can be any of D, G, S, B.

2. If the device is in reverse mode, i.e., $\hat{V}_{DS} < 0$, then the source and drain interchange, and the relationship becomes:

$$\begin{aligned}
\hat{x}_{DS} &= -x'_{DS} \\
\hat{x}_{GD} &= x'_{GS} \\
\hat{x}_{GS} &= x'_{GD} \\
\hat{x}_{GB} &= x'_{GB} \\
\hat{x}_{DB} &= x'_{SB} \\
\hat{x}_{SB} &= x'_{DB}
\end{aligned}$$

These equations are used to calculate the Taylor coefficients of the actual circuit variables in terms of those of the primed variables.

The JFET and MESFET are very similar devices; the only functional difference in their equations is that the stored charges in a MESFET are a function of two controlling variables whereas a JFET has only one. The nonlinearities are:

- The drain source current function:

$$I'_{DS} = f(V'_{GS}, V'_{DS})$$

- The gate drain diode current and charge:

$$I'_{GD} = f(V'_{GD}) \text{ for JFETs and MESFETs}$$

$$Q'_{GD} = f(V'_{GD}) \text{ for JFETs}$$

$$Q'_{GD} = f(V'_{GD}, V'_{GS}) \text{ for MESFETs}$$

- The gate source diode current and charge:

$$I'_{GS} = f(V'_{GS}) \text{ for JFETs and MESFETs}$$

$$Q'_{GS} = f(V'_{GS}) \text{ for JFETs}$$

$$Q'_{GS} = f(V'_{GS}, V'_{GD}) \text{ for MESFETs}$$

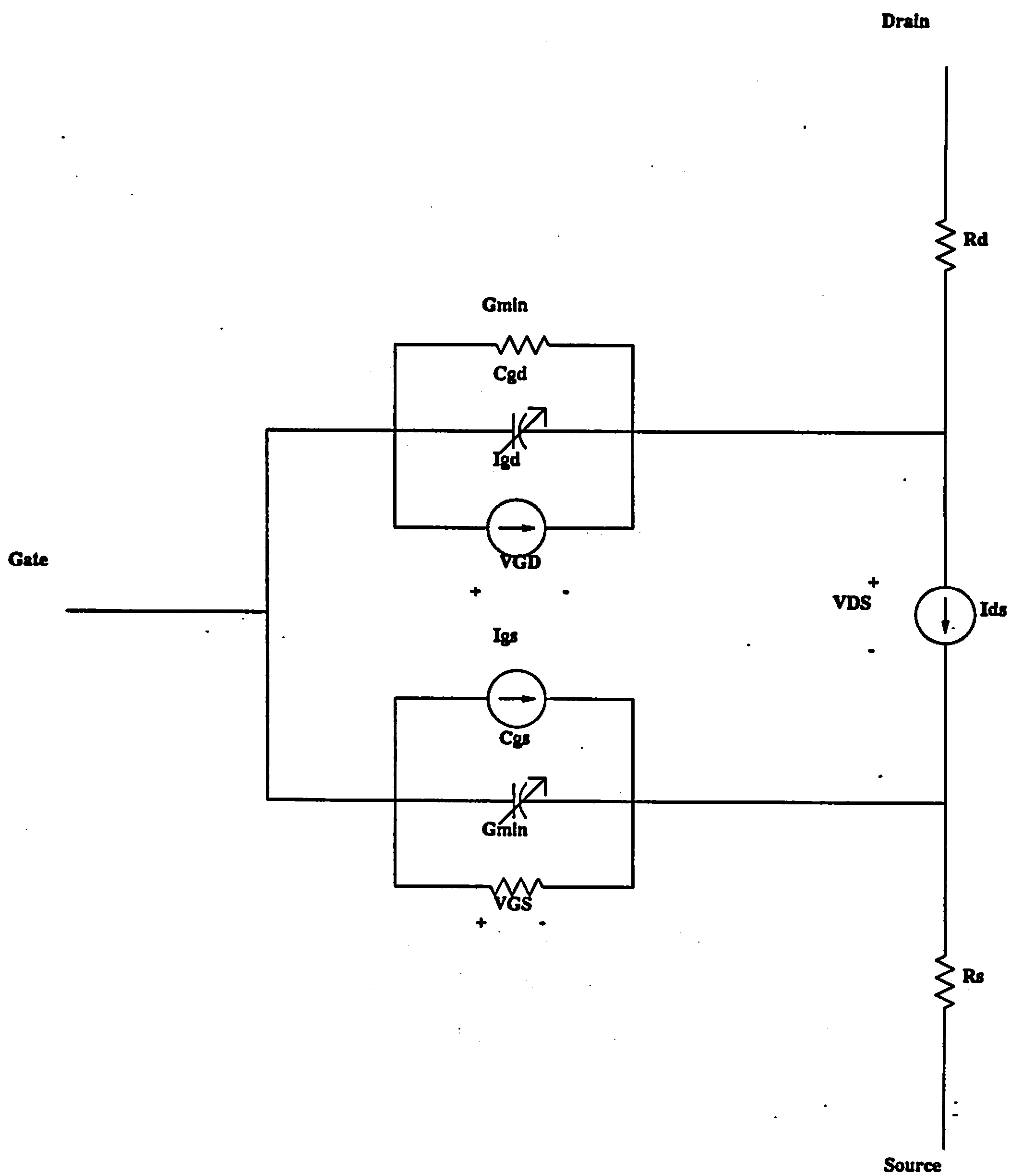

Figure 3.3: JFET model

3.4 The MOS models

The equations for the MOS device models (MOS1, MOS2, MOS3, BSIM, Figure 3.4) all have the same structure in their controlling variables, although the complexity of the nonlinear relationship varies widely from model to model. The hatted and primed terminology used for JFETs and MESFETs is also applicable here. The nonlinearities are:

- The drain source current:

$$I'_{DS} = f(V'_{GS}, V'_{DS}, V'_{BS})$$

- The body drain diode current and charge:

$$I'_{BD} = f(V'_{BD})$$

$$Q'_{BD} = f(V'_{BD})$$

- The body source diode current and charge:

$$I'_{BS} = f(V'_{BS})$$

$$Q'_{BS} = f(V'_{BS})$$

- The gate drain and gate source charges:

$$Q'_{GD}, Q'_{GS} = f(V'_{GS}, V'_{DS}, V'_{BS})$$

- The gate base charge Q'_{GB} is linear in V'_{GB}, and is controlled by other variables in a switched (abruptly nonlinear) manner. Therefore it can be treated as linear element except at the thresholds, and need not be considered.

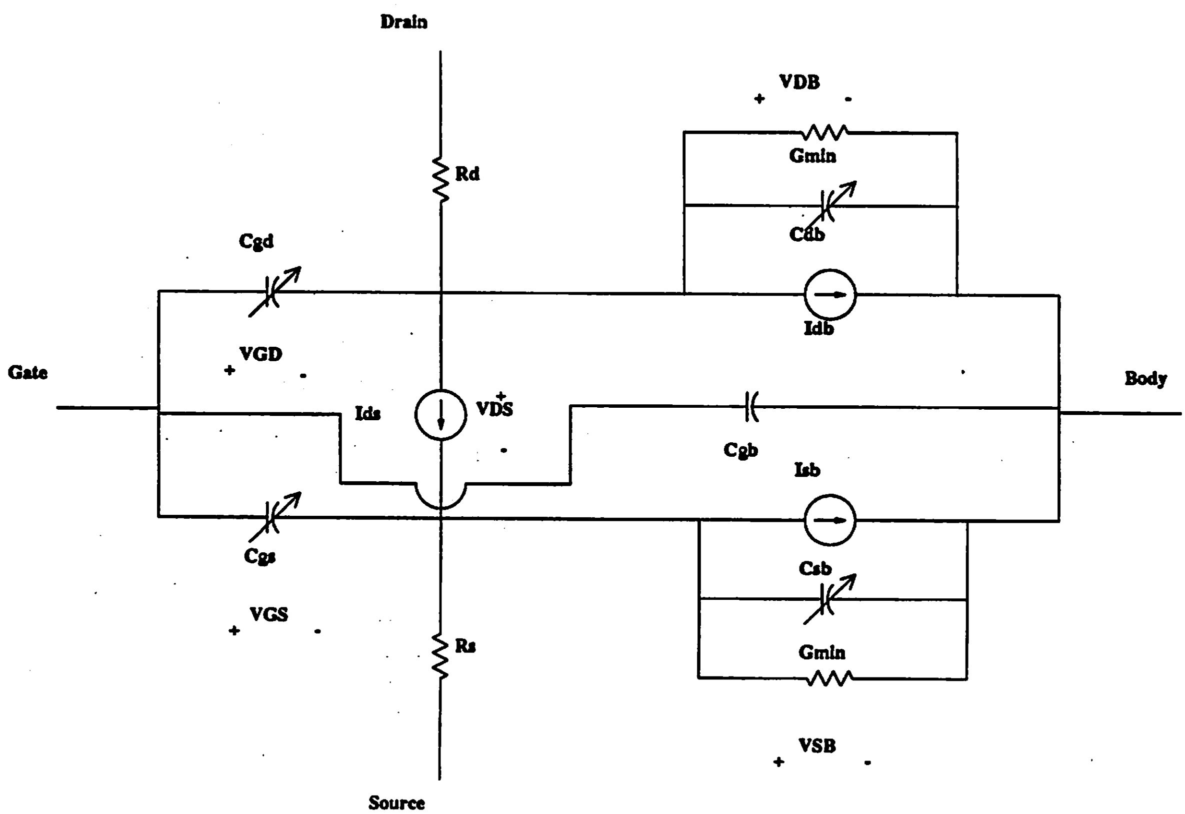

Figure 3.4: MOS model

Chapter 4

Results

The SPICE3 implementation of distortion analysis was used to analyze a number of test circuits. The circuits were chosen to include instances of the common device types - DIOdes, BJTs, JFETs and MOSFETs and MESFETs. For each device, the new analysis was tested on circuits of increasing complexity containing the device. Where available, theoretical values of the distortion magnitudes were compared. Unfortunately, theoretical results were available only for very simple circuits and device models.

Each circuit was run through a transient analysis using the SPICE3 *.tran* command, and this was followed by a Fourier analysis. The magnitudes of the output harmonics were compared with SPICE3 *.disto* results. It is very difficult, unfortunately, to obtain reliable results of intermodulation components from a transient-fourier analysis. Consequently, intermodulation was not analysed with *.tran*. Another problem was that it was virtually impossible to produce a distortion vs. frequency plot using *.tran* on account of the large amount of CPU time taken.

SPECTRE [KS86,KSS88,Kun87] produces accurate plots of harmonic and intermodulation distortion vs frequency. Plots of SPECTRE and SPICE3 distortion were compared and are given in Appendix D. No comparison was made with SPICE2 distortion which is known to be inaccurate.

Table 4.1 is a summary of results for harmonic analyses. Results for intermodulation analyses are presented in Appendix D. It can be seen that there is a good match between SPICE3 *.disto* and SPECTRE results. Transient-Fourier analysis very often makes wrong predictions. In such cases, it is usually possible to arrive at reasonable results by changing some analysis or circuit parameters - *.tran* timesteps, fourier gridsize and interpo-

Circuit		SPICE3 distortion				Transient + Fourier	Spectre
		match with (%)			Run Time	Run Time	Run Time
		Tran-Four	Spectre	Theory			
Diode (D.1)	2ω	0.05	0.04	0.03	0.04	48.14	0.03
	3ω	0.25	0.07	-			
BJT diffpair (D.2)	$2\omega^b$	-	-	-	0.08	132.71	0.08
	3ω	0.104	0.10	0.03			
MOS diffpair (D.3)	$2\omega^b$	-	-[c]	-	0.08	5.73	-[c]
	3ω	0.002	-[c]	0.02			
JFET singlestg (D.4)	2ω	0.51	0.55[d]	-	0.05	68.16	0.45[d]
	3ω	0.846	0.60[d]	-			
JFET diffpair (D.5)	2ω	0.61	0.01	-	0.07	4.6	0.08
	3ω	0.03	0.04	-			
MES[f] singlestg (D.6)	2ω	0.004	0.004	-	0.05	4.53	0.05
	3ω	0.007	0.004	-			
MOSAMP2 (D.9)	2ω	4.25	-[c]	-	1.24	74.92	-[c]
	3ω	-[a]	-[c]	-			
UA741 (D.7)	2ω	-[a]	0.14	-	0.71	59.89	11.85
	3ω	-[a]	0.07	-			
UA733 (D.8)	2ω	-	0.28	-	0.42	32.78	1.03
	3ω	17.8	0.03	-			
Mixer (D.10)	2ω	-	0.019	-	0.07	-	0.13
	3ω	-	0.008	-			
DB Mixer (D.11)	$2\omega^b$	-	-	-	0.16	-	0.62
	3ω	-	0.9	-			

[a] large mismatch
[b] roundoff errors only
[c] Spectre does not support MOS devices
[d] Spectre converges to a somewhat erroneous value; details in Appendix D.4
[e] suspected error in Spectre value
[f] SPICE and SPECTRE MES charge storage models differ; details in Appendix D.6

All times are in secs. on a VAX 8650 running Ultrix 3.0

Table 4.1: Comparison of SPICE3 *.disto* with other methods

lating polynomial degree and input signal magnitude are examples.

The performance of *.disto* on the MOS models could not be properly verified because SPECTRE does not support them. Very simple MOS circuits using level 1 MOS models were found to compare well with theoretical and *.tran* results. There remains a possibility of undetected coding errors in the MOS2, MOS3 and BSIM models, particularly as the distortion code for these models is especially long and convoluted.

Chapter 5

Conclusion and Observations

Distortion analysis has been implemented in SPICE3. Runs on trial circuits indicate that the analysis is accurate. In evaluating the performance of the implementation, comparisons were made with transient-fourier and SPECTRE analyses. Some features of distortion analysis are seen to be:

Accuracy: Distortion analysis is accurate for small input and output values. Transient analysis is inaccurate for small signals because the errors introduced by RELTOL may be larger than the magnitudes of the harmonics produced. Subsequent Fourier analysis also adds to the error, especially in weak-third order harmonics very close to strong first-order signals. SPECTRE results match very closely with SPICE3 distortion results in most cases; when the output magnitudes approach extremely low values (of the order of fV), SPECTRE displays noise that arises from Newton-Raphson convergence at these low values. *.disto* which does not use numerical convergence produces better results in such cases (see Appendix D). SPECTRE does not implement linear g_m delay and the nonlinear base resistance of the BJT whereas SPICE3 does; it is observed that the nonlinear base resistance contributes quite significantly to distortion.

Speed: Distortion analysis is much faster than transient analysis (typically 50–100 times faster for a single cycle transient analysis, not counting the time taken for the Fourier analysis). Often, in the two-frequency input case, several cycles have to be simulated because the intermodulation frequencies are much smaller than the fundamental frequencies. In such cases, distortion analysis may be 1000 or more times faster than transient analysis. SPECTRE is about as fast as *.disto* for small circuits; as circuit size increases, the time taken by *.disto* increases linearly and is typically 10 times less than SPECTRE time for medium circuits, for a single frequency point. A large fraction of the time taken by *.disto* consists of doing operating point setup calculations which do not have to be repeated for different frequencies; when plots are produced, *.disto* is about 3–10 times faster for harmonic analysis, and 40–100 times faster for intermodulation analysis, than SPECTRE. Such a comparison may not be meaningful, though, as SPECTRE is capable of more diverse application than *.disto*.

Convenience: Transient analysis requires several parameters (TSTEP, TSTOP, TMAX) to be adjusted by trial and error before a stable result is reached. The accuracy of a Fourier analysis is critically dependent on the base frequency, and also depends on the number of grid points and the degree of the interpolating polynomial. Arriving at a dependable result using the transient+fourier technique usually involves trying several different values of the parameters. In contrast, there are no parameters to vary in distortion analysis. In SPECTRE, it may be necessary to vary parameters to accelerate convergence, though this is usually not required.

The disadvantages of distortion analysis are:

Large input levels: The truncation of the Volterra and Taylor series causes distortion analysis to become inaccurate for large signal levels. Transient analysis is therefore much more accurate for large signals. The errors due to RELTOL also decrease in significance because the distortion components become disproportionately stronger compared to the input signal when it increases. SPECTRE remains accurate at all normally encountered levels of distortion.

Upper bounds for signal levels: No rigorous estimate of the error caused by series truncation is available for general circuits. Guidelines for limiting signal levels can be arrived at from an analysis of typical small building blocks like single and differential amplifier stages.

Length of code: The distortion code is long, mainly because of the Taylor coefficient and distortion source evaluation code. This is because all the intricacies of each model are faithfully reflected in the distortion code. An idea of which distortion generators can be neglected may help reduce the amount of code significantly, especially in the MOS2, MOS3, and BSIM models.

Bibliography

[CN73] S. H. Chisholm and L. W. Nagel. Efficient computer simulation of distortion in electronic circuits. *IEEE Transaction on Circuit Theory*, CT-20:742–745, November 1973.

[KS86] Kenneth S. Kundert and Alberto Sangiovanni-Vincentelli. Simulation of nonlinear circuits in the frequency domain. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, CAD-5(4):521–535, October 1986.

[KSS88] Kenneth S. Kundert, Gregory B. Sorkin, and Alberto Sangiovanni-Vincentelli. Applying harmonic balance to almost-periodic circuits. *IEEE Transactions on Microwave Theory and Techniques*, MTT-36(2):366–378, February 1988.

[Kun87] Kenneth S. Kundert. *Spectre User's Guide: A Frequency Domain Simulator for Nonlinear Circuits*. University of California, Berkeley, EECS Industrial Liaison Program, University of California, Berkeley California, 94720, April 1987.

[Kuo73] Y. L. Kuo. Distortion analysis of bipolar transistor circuits. *IEEE Transactions on Circuit Theory*, CT-20(6), November 1973.

[Nag73] Lawrence W. Nagel. *SPICE2*. PhD thesis, EECS department, University of California, Berkeley, 1973.

[Nar67] S. Narayanan. Transistor Distortion Analysis using Volterra Series representation. *Bell System Technical Journal*, May-June 1967.

[Ped87] Donald O. Pederson. *EECS 142 Reader*. Fall 1987. Course notes for EECS 142 offered at UC Berkeley.

[Qua87] Thomas L. Quarles. *SPICE 3B.1 User's Guide.* University of California, Berkeley, EECS Industrial Liaison Program, University of California, Berkeley California, 94720, April 1987.

[Qua89a] Thomas L. Quarles. *SPICE 3C.1 User's Guide.* University of California, Berkeley, EECS Industrial Liaison Program, University of California, Berkeley California, 94720, April 1989.

[Qua89b] Thomas L. Quarles. *SPICE3.* PhD thesis, EECS department, University of California, Berkeley, 1989.

[Vol59] V. Volterra. *Theory of Functionals and of Integral and Integro-Differential Equations.* Dover, 1959.

[WS80] Donald D. Weiner and John F. Spina. *Sinusoidal Analysis and Modeling of Weakly Nonlinear Circuits.* Van Nostrand Reinhold Company, 1980.

Appendix A

Volterra expansions for useful inputs

This appendix lists the volterra series for some exponential and sinusoidal inputs. x denotes the variable for which the series is given, and the superscripts ^{x} indicate that the Volterra transforms are for the variable x.

A.1 Input $= e^{j\omega t}$

$$x(t) = H_1^x(\omega)e^{j\omega t} + H_2^x(\omega,\omega)e^{j2\omega t} + H_3^x(\omega,\omega,\omega)e^{j3\omega t} + \cdots$$

A.2 Input $= e^{j\omega_1 t} + e^{j\omega_2 t}$

$$\begin{aligned} x(t) \;=\;& H_1^x(\omega_1)e^{j\omega_1 t} + H_1^x(\omega_2)e^{j\omega_2 t} + 2H_2^x(\omega_1,\omega_2)e^{j(\omega_1+\omega_2)t} \\ &+ H_2^x(\omega_1,\omega_1)e^{j2\omega_1 t} + H_2^x(\omega_2,\omega_2)e^{j2\omega_2 t} + H_3^x(\omega_1,\omega_1,\omega_1)e^{j3\omega_1 t} \\ &+ H_3^x(\omega_1,\omega_1,\omega_2)e^{j(2\omega_1+\omega_2)t} + \cdots \end{aligned}$$

A.3 Input $= A\cos(\omega t + \phi_1)$

The input is rewritten (with the cos term in exponential notation) as:

$$\frac{1}{2}\left(E_1 e^{j\omega t} + E_1^* e^{-j\omega t}\right)$$

E_1 is the complex amplitude of the sinusoidal wave.

The Volterra series for this input is very similar to the above with ω_1 replaced by ω and ω_2 replaced by $-\omega$, and is not given here. The complex amplitudes of the important terms are given instead.

- <u>$e^{j2\omega t}$ term:</u> $\frac{1}{2}E_1E_1H_2^x(\omega,\omega)$

- $\underline{e^{j3\omega t} \text{ term:}}$ $\frac{1}{2}\frac{1}{2}E_1E_1E_1H_3^{\pi}(\omega,\omega,\omega)$

A.4 Input $= A\cos(\omega_1 t + \phi_1) + B\cos(\omega_2 t + \phi_2)$

The input is rewritten as:

$$\frac{1}{2}\left(E_1e^{j\omega_1 t} + E_1^*e^{-j\omega_1 t}\right) + \frac{1}{2}\left(E_2e^{j\omega_2 t} + E_2^*e^{-j\omega_2 t}\right)$$

The entire series for this input is long. The complex amplitudes of the important terms are listed here. For the complete series, refer to [WS80, page 89].

- $\underline{e^{j(\omega_1+\omega_2)t} \text{ term:}}$ $E_1E_2H_2^{\pi}(\omega_1,\omega_2)$
- $\underline{e^{j(\omega_1-\omega_2)t} \text{ term:}}$ $E_1E_2^*H_2^{\pi}(\omega_1,-\omega_2)$
- $\underline{e^{j(2\omega_1-\omega_2)t} \text{ term:}}$ $\frac{1}{2}\frac{1}{2}3E_1E_1E_2^*H_3^{\pi}(\omega_1,\omega_1,-\omega_2)$

Appendix B

Distortion Source Formulae

This appendix lists the distortion source formulae for functions of three variables. Each branch relationship's controlled variable is expanded in a Taylor series in (at most) three controlling variables. All variables refer to the small-signal deviations. In the following, i is the controlled variable and x, y, and z are the controlling variables. The Taylor series is as follows:

$$\begin{aligned} i \;=\; & c_x x + c_y y + c_z z + c_{xx} x^2 + c_{yy} y^2 + c_{zz} z^2 \\ & + c_{xy} xy + c_{yz} yz + c_{xz} xz + c_{xxx} x^3 + c_{yyy} y^3 + c_{zzz} z^3 \\ & + c_{xxy} x^2 y + c_{xxz} x^2 z + c_{xyy} xy^2 + c_{yyz} y^2 z + c_{xzz} xz^2 + c_{yzz} yz^2 \\ & + c_{xyz} xyz \end{aligned}$$

The c's are the Taylor series coefficients. The distortion source for each harmonic and intermodulation frequency will be expressed in terms of these coefficients. In the following formulae, the superscripts of the Volterra transforms identify the controlling variable with which they are associated.

B.1 Distortion source at 2ω

For convenience in representation, define:

$$f(a, b, c) = a \cdot b \cdot c$$

The distortion source for i at 2ω is:

$$\begin{aligned} & f(c_{xy}, H_1^x(\omega), H_1^x(\omega)) + f(c_{yy}, H_1^y(\omega), H_1^y(\omega)) \\ & \quad + f(c_{zz}, H_1^z(\omega), H_1^x(\omega)) + f(c_{xy}, H_1^x(\omega), H_1^y(\omega)) \\ & \quad + f(c_{yz}, H_1^y(\omega), H_1^z(\omega)) + f(c_{xz}, H_1^x(\omega), H_1^z(\omega)) \end{aligned}$$

B.2 Distortion source at 3ω

Define:

$$f(a,b,c,d,e) = a(be + cd)$$

and:

$$g(a,b,c,d) = abcd$$

Then the distortion source for i at 3ω is:

$$\begin{aligned}
&f(c_{xx}, H_1^x(\omega), H_1^x(\omega), H_2^x(\omega,\omega), H_2^x(\omega,\omega)) + \\
&f(c_{yy}, H_1^y(\omega), H_1^y(\omega), H_2^y(\omega,\omega), H_2^y(\omega,\omega)) + \\
&f(c_{zz}, H_1^z(\omega), H_1^z(\omega), H_2^z(\omega,\omega), H_2^z(\omega,\omega)) + \\
&f(c_{xy}, H_1^x(\omega), H_1^y(\omega), H_2^x(\omega,\omega), H_2^y(\omega,\omega)) + \\
&f(c_{yz}, H_1^y(\omega), H_1^z(\omega), H_2^y(\omega,\omega), H_2^z(\omega,\omega)) + \\
&f(c_{xz}, H_1^x(\omega), H_1^z(\omega), H_2^x(\omega,\omega), H_2^z(\omega,\omega)) + \\
&g(c_{xxx}, H_1^x(\omega), H_1^x(\omega), H_1^x(\omega)) + \\
&g(c_{yyy}, H_1^y(\omega), H_1^y(\omega), H_1^y(\omega)) + \\
&g(c_{zzz}, H_1^z(\omega), H_1^z(\omega), H_1^z(\omega)) + \\
&g(c_{xxy}, H_1^x(\omega), H_1^x(\omega), H_1^y(\omega)) + \\
&g(c_{xxz}, H_1^x(\omega), H_1^x(\omega), H_1^z(\omega)) + \\
&g(c_{xyy}, H_1^x(\omega), H_1^y(\omega), H_1^y(\omega)) + \\
&g(c_{yyz}, H_1^y(\omega), H_1^y(\omega), H_1^z(\omega)) + \\
&g(c_{xzz}, H_1^x(\omega), H_1^z(\omega), H_1^z(\omega)) + \\
&g(c_{yzz}, H_1^y(\omega), H_1^z(\omega), H_1^z(\omega)) + \\
&g(c_{xyz}, H_1^x(\omega), H_1^y(\omega), H_1^z(\omega))
\end{aligned}$$

B.3 Distortion source at $\omega_1 + \omega_2$

Define:

$$f(a,b,c,d,e) = \frac{a}{2}(be + cd)$$

Then the distortion source for i at $\omega_1 + \omega_2$ is:

$$\begin{aligned}
&f(c_{xx}, H_1^x(\omega_1), H_1^x(\omega_1), H_1^x(\omega_2), H_1^x(\omega_2)) + \\
&f(c_{yy}, H_1^y(\omega_1), H_1^y(\omega_1), H_1^y(\omega_2), H_1^y(\omega_2)) + \\
&f(c_{zz}, H_1^z(\omega_1), H_1^z(\omega_1), H_1^z(\omega_2), H_1^z(\omega_2)) + \\
&f(c_{xy}, H_1^x(\omega_1), H_1^y(\omega_1), H_1^x(\omega_2), H_1^y(\omega_2)) + \\
&f(c_{yz}, H_1^y(\omega_1), H_1^z(\omega_1), H_1^y(\omega_2), H_1^z(\omega_2)) + \\
&f(c_{xz}, H_1^x(\omega_1), H_1^z(\omega_1), H_1^x(\omega_2), H_1^z(\omega_2))
\end{aligned}$$

B.4 Distortion source at $2\omega_1 + \omega_2$

Define:

$$f(a,b,c,d,e,f,g,h,l) = \frac{a}{3}\left[2\,(al + ch) + dg + ef\right]$$

and:

$$g(a,b,c,d,e,f,g) = \frac{a}{3}(bcg + bfd + ecd)$$

Then the distortion source for i at $2\omega_1 + \omega_2$ is:

$$\begin{gathered}
f(c_{xx}, H_1^x(\omega_1), H_1^x(\omega_1), H_1^x(\omega_2), H_1^x(\omega_2), H_2^x(\omega_1,\omega_1), H_2^x(\omega_1,\omega_1), \\
H_2^x(\omega_1,\omega_2), H_2^x(\omega_1,\omega_2))+ \\
f(c_{yy}, H_1^y(\omega_1), H_1^y(\omega_1), H_1^y(\omega_2), H_1^y(\omega_2), H_2^y(\omega_1,\omega_1), H_2^y(\omega_1,\omega_1), \\
H_2^y(\omega_1,\omega_2), H_2^y(\omega_1,\omega_2))+ \\
f(c_{zz}, H_1^z(\omega_1), H_1^z(\omega_1), H_1^z(\omega_2), H_1^z(\omega_2), H_2^z(\omega_1,\omega_1), H_2^z(\omega_1,\omega_1), \\
H_2^z(\omega_1,\omega_2), H_2^z(\omega_1,\omega_2))+ \\
f(c_{xy}, H_1^x(\omega_1), H_1^y(\omega_1), H_1^x(\omega_2), H_1^y(\omega_2), H_2^x(\omega_1,\omega_1), H_2^y(\omega_1,\omega_1), \\
H_2^x(\omega_1,\omega_2), H_2^y(\omega_1,\omega_2))+ \\
f(c_{yz}, H_1^y(\omega_1), H_1^z(\omega_1), H_1^y(\omega_2), H_1^z(\omega_2), H_2^y(\omega_1,\omega_1), H_2^z(\omega_1,\omega_1), \\
H_2^y(\omega_1,\omega_2), H_2^z(\omega_1,\omega_2))+ \\
f(c_{xz}, H_1^x(\omega_1), H_1^z(\omega_1), H_1^x(\omega_2), H_1^z(\omega_2), H_2^x(\omega_1,\omega_1), H_2^z(\omega_1,\omega_1), \\
H_2^x(\omega_1,\omega_2), H_2^z(\omega_1,\omega_2))+ \\
g(c_{xxx}, H_1^x(\omega_1), H_1^x(\omega_1), H_1^x(\omega_1), H_1^x(\omega_2), H_1^x(\omega_2), H_1^x(\omega_2))+ \\
g(c_{yyy}, H_1^y(\omega_1), H_1^y(\omega_1), H_1^y(\omega_1), H_1^y(\omega_2), H_1^y(\omega_2), H_1^y(\omega_2))+ \\
g(c_{zzz}, H_1^z(\omega_1), H_1^z(\omega_1), H_1^z(\omega_1), H_1^z(\omega_2), H_1^z(\omega_2), H_1^z(\omega_2))+ \\
g(c_{xxy}, H_1^x(\omega_1), H_1^x(\omega_1), H_1^y(\omega_1), H_1^x(\omega_2), H_1^x(\omega_2), H_1^y(\omega_2))+ \\
g(c_{xxz}, H_1^x(\omega_1), H_1^x(\omega_1), H_1^z(\omega_1), H_1^x(\omega_2), H_1^x(\omega_2), H_1^z(\omega_2))+ \\
g(c_{xyy}, H_1^x(\omega_1), H_1^y(\omega_1), H_1^y(\omega_1), H_1^x(\omega_2), H_1^y(\omega_2), H_1^y(\omega_2))+ \\
g(c_{yyz}, H_1^y(\omega_1), H_1^y(\omega_1), H_1^z(\omega_1), H_1^y(\omega_2), H_1^y(\omega_2), H_1^z(\omega_2))+ \\
g(c_{xzz}, H_1^x(\omega_1), H_1^z(\omega_1), H_1^z(\omega_1), H_1^x(\omega_2), H_1^z(\omega_2), H_1^z(\omega_2))+ \\
g(c_{yzz}, H_1^y(\omega_1), H_1^z(\omega_1), H_1^z(\omega_1), H_1^y(\omega_2), H_1^z(\omega_2), H_1^z(\omega_2))+ \\
g(c_{xyz}, H_1^x(\omega_1), H_1^y(\omega_1), H_1^z(\omega_1), H_1^x(\omega_2), H_1^y(\omega_2), H_1^z(\omega_2))
\end{gathered}$$

Appendix C

How to use SPICE3 distortion analysis

This describes how to use SPICE3C.ir.2, an experimental version of SPICE3 that supports small signal distortion analysis based on Volterra series.

SPICE3C.ir.2 is accessible as **/cad/new/vice** on ic. In addition to the usual SPICE3 commands, SPICE3C.ir.2 supports a new analysis type called *.disto*.

The syntax of the *.disto* command is as follows:

vxxxxxxx n+ n- distof1 < f1mag < f1phase >> distof2 < f2mag < f2phase >>
ixxxxxxx n+ n- distof1 < f1mag < f1phase >> distof2 < f2mag < f2phase >>
.disto { *lin dec oct* } *numpts fstart fstop < f2overf1 >*

Each independent source in the circuit is a potential input for distortion analysis. If the optional keyword *f2overf1* is not specified, all inputs to the circuit are assumed to be of a single frequency. *.disto* then does a harmonic analysis - i.e., it calculates the values of the 2^{nd} and 3^{rd} harmonics of this frequency at every point in the circuit. The magnitude and phase of each input source at the fundamental frequency f_1 are specified by the arguments of the *distof1* keyword in the input file line for the source. The default values of the magnitude and phase are 1.0 and 0.0 respectively; the phase should be specified in degrees. If a source does not have the *distof1* keyword, it does not contribute to the input excitation for a harmonic analysis. The arguments of the *.disto* command, excepting *f2overf1* if specified, are identical to the *.ac* command arguments, and are used to sweep the frequency f_1 to obtain a plot of distortion vs input frequency. It is to be noted that the plots produced are of the values at the harmonic frequencies, but plotted against the fundamental frequency f_1.

If the optional keyword *f2overf1* is specified, it should be a real number between (and not equal to) 0.0 and 1.0; in this case, *.disto* does a spectral (intermodulation) analysis - i.e., it assumes that there are two different input frequencies f_1 and f_2 present in the circuit, and calculates the values of the circuit variables at the cross product frequencies $f_1 + f_2$, $f_1 - f_2$ and $2f_1 - f_2$. f_1 is swept according to the *.disto* arguments *numpts*, *fstart* and *fstop* exactly as in the harmonic analysis case. f_2 is *kept constant* at the value $f2overf1 \times fstart$. This is done for compatibility with SPECTRE. The magnitude and phase of each independent source at the frequencies f_1 and f_2 are specified by the arguments of the *distof1* and *distof2* keywords respectively.

The *.disto* analysis currently supports the diode, BJT, JFET, MOS1, MOS2, MOS3, BSIM and MES models. Switches cannot been implemented because they are a strong nonlinearity; however, if a circuit contains switches which will not flip for the small input values used in distortion analysis, a *.disto* analysis will be accurate.

The current version of the *.disto* analysis produces the distortion values for *every* circuit variable, unlike SPICE2 in which a single load resistor has to be specified. SPICE2, however, has a feature by which the contribution of each nonlinearity to the output distortion can be computed. This is not currently supported in SPICE3, but will be built into future versions. The distortion code will be released in the version SPICE3C.2. Please send bugs and suggestions to *jaijeet@ic.Berkeley.EDU*.

Appendix D

Results

The following lists the results of running *.disto* on test circuits. For each circuit, plots comparing *.disto* and SPECTRE values are given, along with listings of the input files.

SPECTRE [KS86,KSS88,Kun87] is a frequency domain circuit analysis program that calculates the fourier components of the steady-state periodic waveforms of a circuit. It can handle harmonics or frequency mixes of upto 15^{th} order, and solves a large system of coupled nonlinear equations in the unknown fourier components to arrive at accurate results. The reason that *.disto* does better than SPECTRE in terms of speed is that the Volterra series technique is essentially a one-shot relaxation step to solve the same equations that SPECTRE solves, but with the following important differences:

1. Frequency components of upto only third order are considered.
2. The fourier components of each succeeding harmonic or intermodulation order are assumed to be much smaller than those of the preceding order and are ignored in comparison.
3. The equations are *ordered* in such a way that at each step in the relaxation process, each equation (with simplifications as in 2) can be solved exactly in terms of known constants or previously calculated unknowns.

Because of the assumptions made, *.disto* is not accurate for large distortion levels where the assumptions break down. Since SPECTRE does not make these assumptions, it remains accurate at these levels. On the other hand, since *.disto* does not use techniques that require convergence or the solution of nonlinear equations, it is faster and sometimes more accurate that SPECTRE at small distortion levels.

The data in Table 4.1 was compiled by running the circuits at the single frequency point 1kHz. In order to be able to theoretically predict distortion values, some of the circuits and device models were simplified; this was done only for generating results to be used in Table 4.1. The simplifications are mentioned for the appropriate circuits. Note that all plots in this appendix were produced from the given input listings.

The transient analysis data in Table 4.1 were produced with the following *.tran parameters*: $TSTEP = 1\mu s$, $TMAX = 1\mu s$, $TSTOP = 1ms$, $TSTART = 0$. The input amplitudes were different for the different circuits; they are given in the circuit listings. The fourier analysis parameter values used were: *fourgridsize* = 1000, and *polydegree* = 3.

D.1 Diode-Resistor circuit

The circuit is a simple diode-resistor-voltage-source series circuit. The data in Table 4.1 was obtained by running the given input files with the diode model replaced by the default model, in both SPICE3 and SPECTRE. The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for the analyses running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	0.43	1.5	3
Spectral	0.75	67.65	90

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 computes only 6[1]. A more reasonable speedup factor in the spectral case is therefore: 90/2 = 45

[1] *.disto* calculates distortion components at the frequencies $f_1 + f_2$, $f_1 - f_2$ and $2f_1 - f2$, in addition to the internally used values at f_1, f_2 and $2f_1$; SPECTRE computes components at all mixes of upto third order, which number 12

diode.spice3 Fri Mar 10 21:49:57 1989 1

```
        Diode-Resistor circuit

r1 2 0 1k
d1 1 2 diode
vcc 1 3 5v ac 0.001 sin(5 0.01 1000) distof1 0.01 distof2 0.01
vcc2 3 0 0v
*.model diode D
.model diode D is=1.0e-14 tt=0.1n cjo=2p
*.disto dec 1 1000.0 1000.0
*.disto dec 20 1.0e3 1.0e8
*.disto dec 20 1.0e3 1.0e8 0.9
*the output is v(2)
.end
```

diode.spectre Fri Jan 27 19:38:59 1989 1

```
;       Diode-Resistor circuit
global 0


;circuit
r1 2 0 resistor r=1k
d1 1 2 dio
vcc 1 3 vsource vdc=5 mag=0.01 mag1=0.01 mag1f=0.01 mag2f=0.01 ;ac sin(5 0.01 1000)
vcc2 3 0 vsource vdc=0 ;sin(0 0.01 900)

;models
;model dio diode
model dio diode is=1.0e-14 tt=0.1n cjo=2p

;analyses

;the output is v(2)
;acan ac start=1.0e4 stop=1.0e8 dec=20
;boom harmonic maxharm=3 fund=1.0e3
boom harmonic maxharm=3 start=1.0e4 stop=1.0e8 dec=20 ;saman=0 thresh=1.0e-10
aagh spectral fund2=0.9e4 order=3 maxharm1=3 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e8 dec=20 saman=0
```

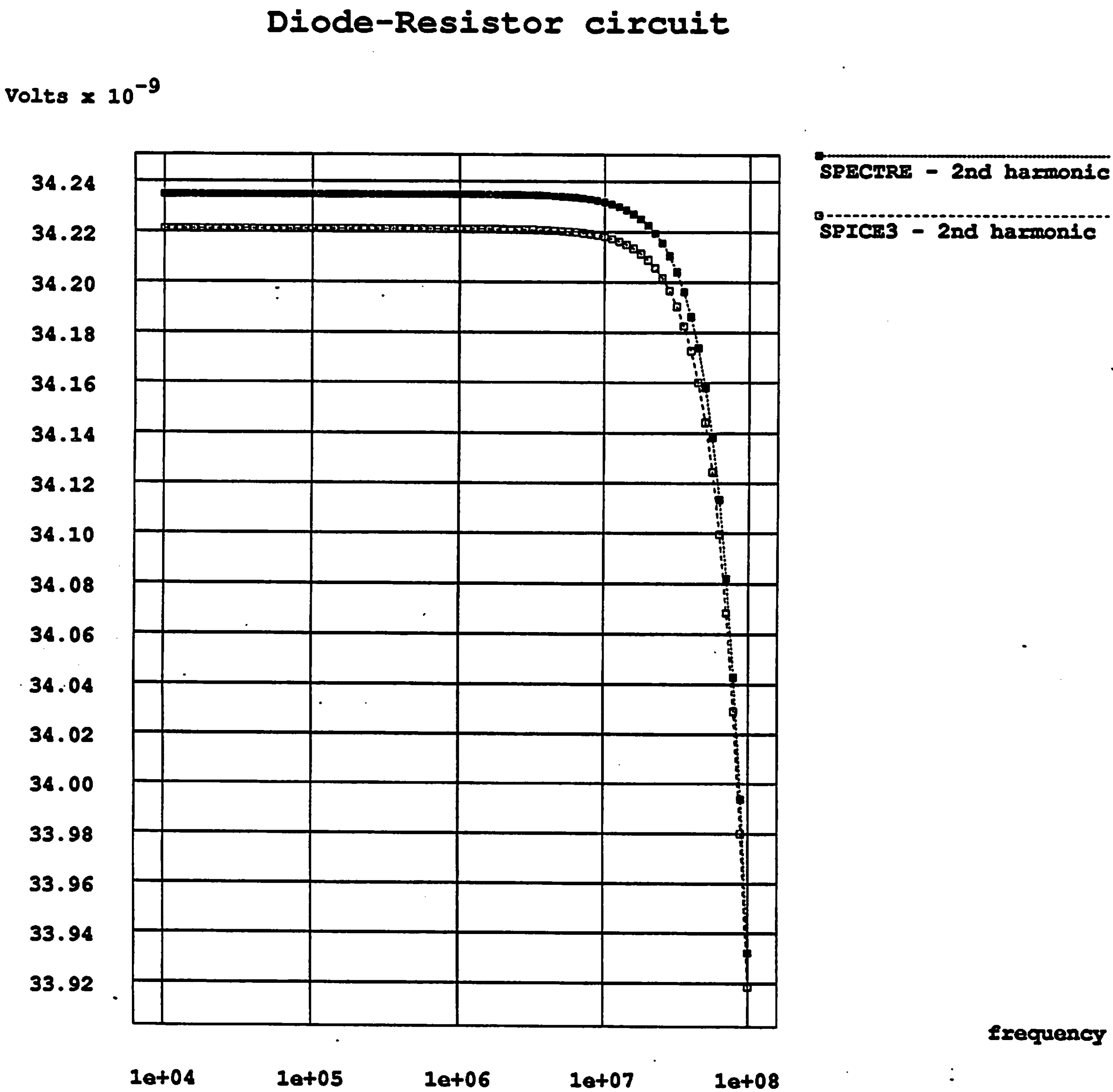

Figure D.1: Diode-Resistor circuit - 2nd harmonic

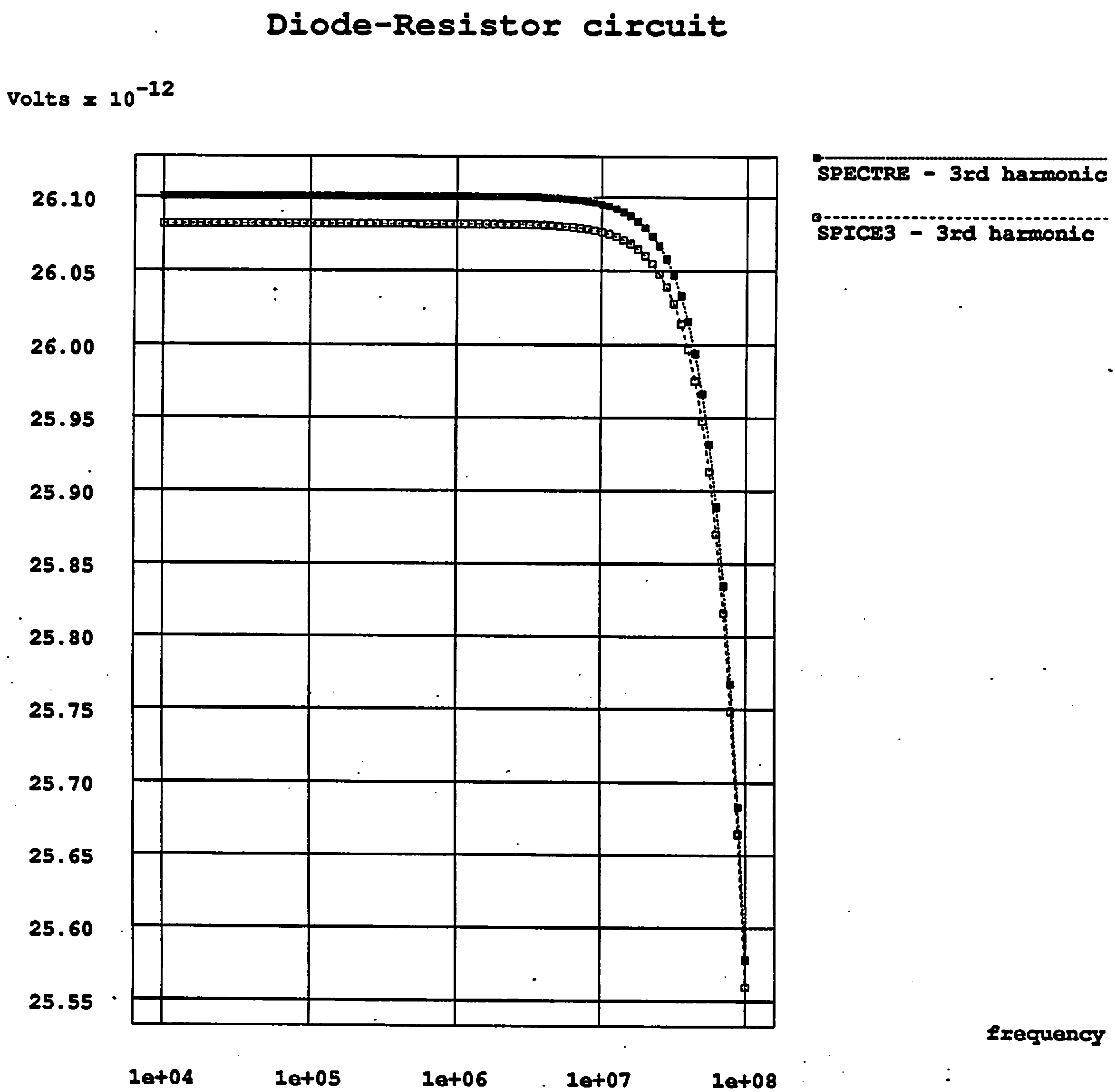

Figure D.2: Diode-Resistor circuit - 3rd harmonic

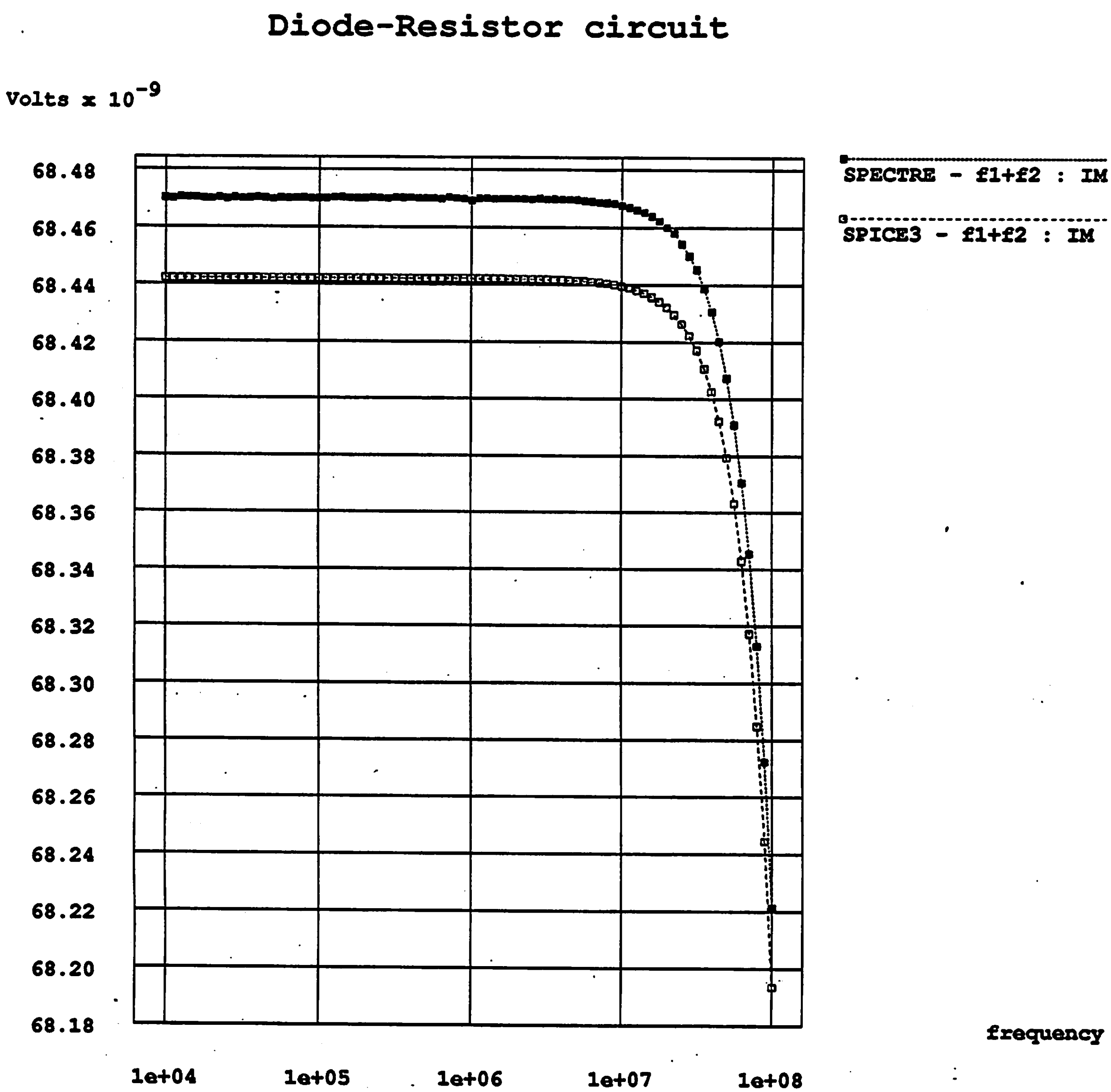

Figure D.3: Diode-Resistor circuit - IM : $f_1 + f_2$

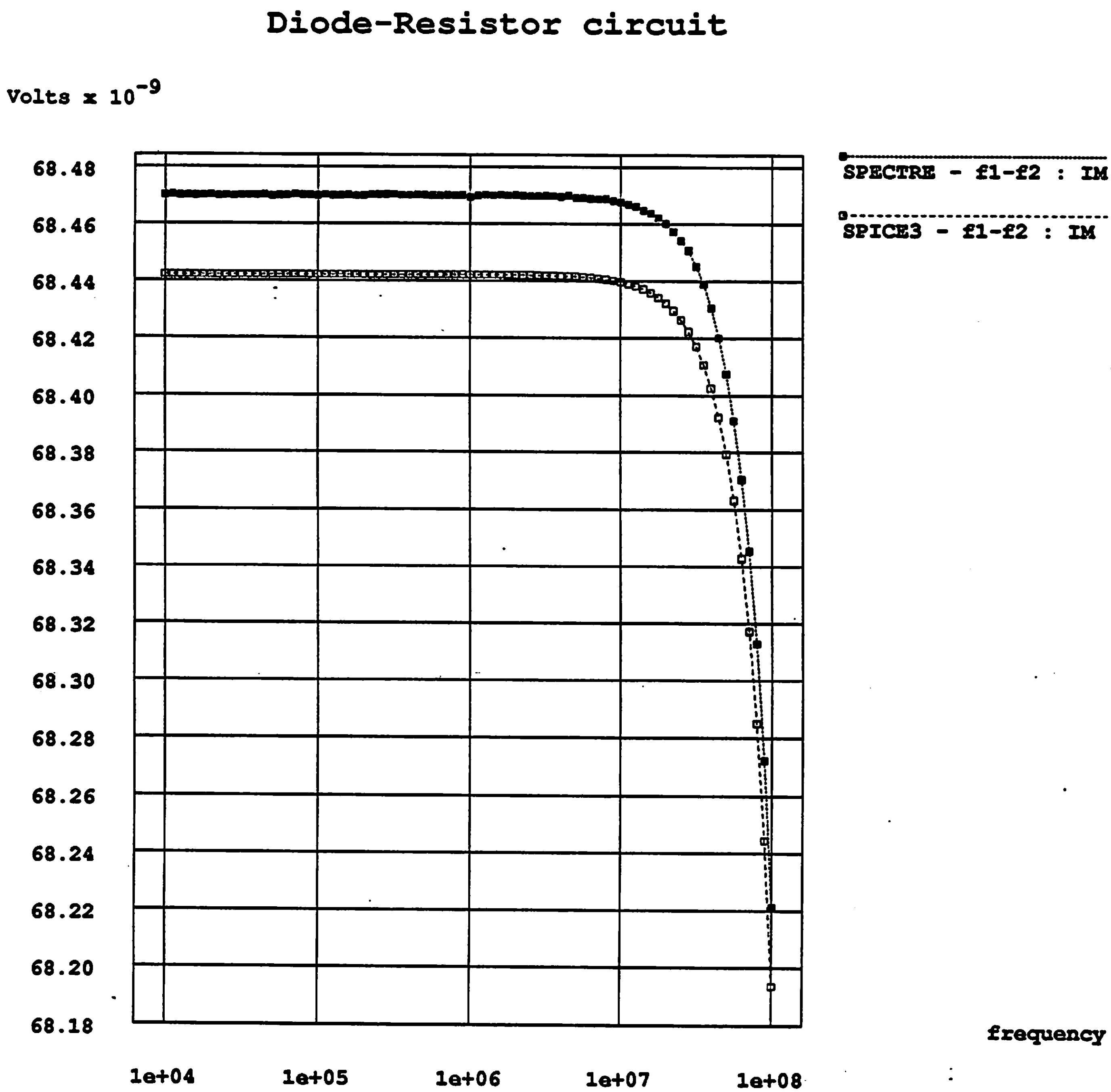

Figure D.4: Diode-Resistor circuit - IM : $f_1 - f_2$

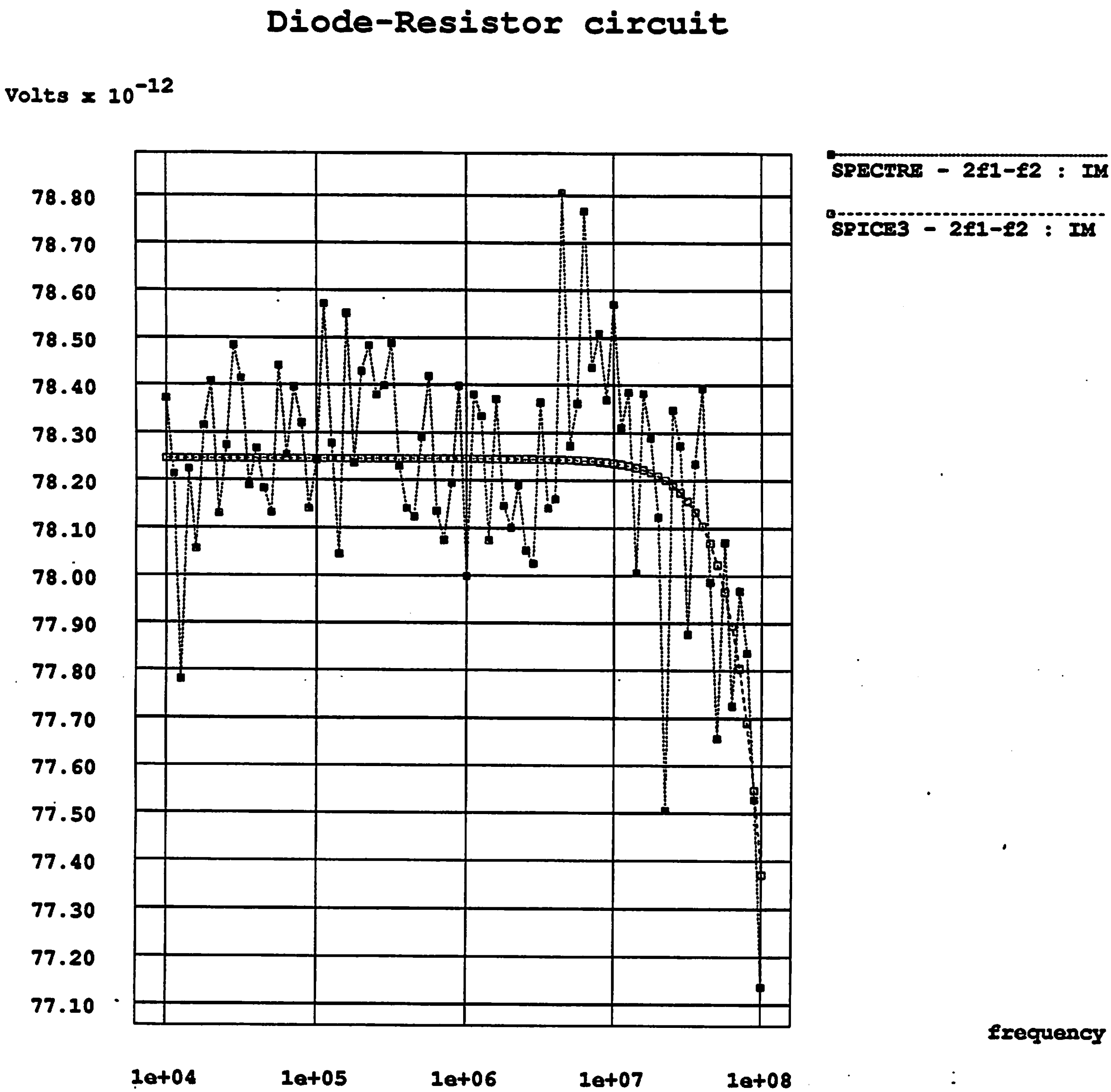

Figure D.5: Diode-Resistor circuit - IM : $2f_1 - f_2$

D.2 BJT differential pair circuit

The circuit is a simple BJT differential pair circuit, without emitter degeneration.

The data in Table 4.1 was obtained by running the given input files with the BJT model replaced by a simpler model with only I_S and β_F specified, in both SPICE3 and SPECTRE. Details are given in the input listing. In such a case (with no second order effects and no charge storage elements), the theoretical value of the second harmonic at the output is zero. The results of simulation are just the roundoff/truncation errors in numerical calculation, and were ignored for the purposes of Table 4.1.

If charge storage effects are modelled, a strong second harmonic component does arise at middle frequencies, rather surprisingly. This can be seen in the plots.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for the analyses running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	1.89	9.27	5
Spectral	3.78	305.33	81

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: $80/2 = 40$

bjtdiffpair.spice3 **Fri Mar 10 21:48:21 1989** **1**

```
BJT differential pair circuit

v1 1 0  0v distof1 0.0001 distof2 0.0001
q1 3 1 4 2 mod1
q2 6 0 4 2 mod1
rc1 5 3 10k
rc2 5 6 10k
vcc 5 0 10v
iee 4 2 1m
vee 2 0 -10v

*.model mod1 npn is=1.0e-16 bf=100
.model mod1 npn is=1e-16 bf=100 rb=100 cjs=2.0e-12 tf=0.3e-9 tr=6e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50
*.disto dec 1 1000 1000
*.disto dec 20 1.0e3 1.0e8
*.disto dec 20 1.0e3 1.0e8 0.9
*the output is v(6)
.end
```

bjtdiffpair.spectre **Fri Jan 27 19:37:59 1989** **1**

```
; BJT differential pair circuit

; Circuit
global 0

v1 1 0 vsource vdc=0 mag=0.0001 phase=0.0 mag1=0.0001 mag1f=0.0001 mag2f=0.0001
q1 3 1 4 2 mod1 region=1
q2 6 0 4 2 mod1 region=1
rc1 5 3 resistor r=10k
rc2 5 6 resistor r=10k
vcc 5 0 vsource vdc=10
iee 4 2 isource idc=1m
vee 2 0 vsource vdc=-10

;models

;model mod1 bjt npn=yes is=1e-16 bf=100
model mod1 bjt npn=yes is=1e-16 bf=100 rb=100 cjs=2.0e-12 tf=0.3e-9 tr=6e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50

; Analyses

;output is v(6)
;boom harmonic maxharm=3 fund=1.0e3
boom harmonic maxharm=3 start=1.0e4 stop=1.0e8 dec=20
aagh spectral fund2=0.9e4 order=3 maxharm1=3 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e8 dec=20
```

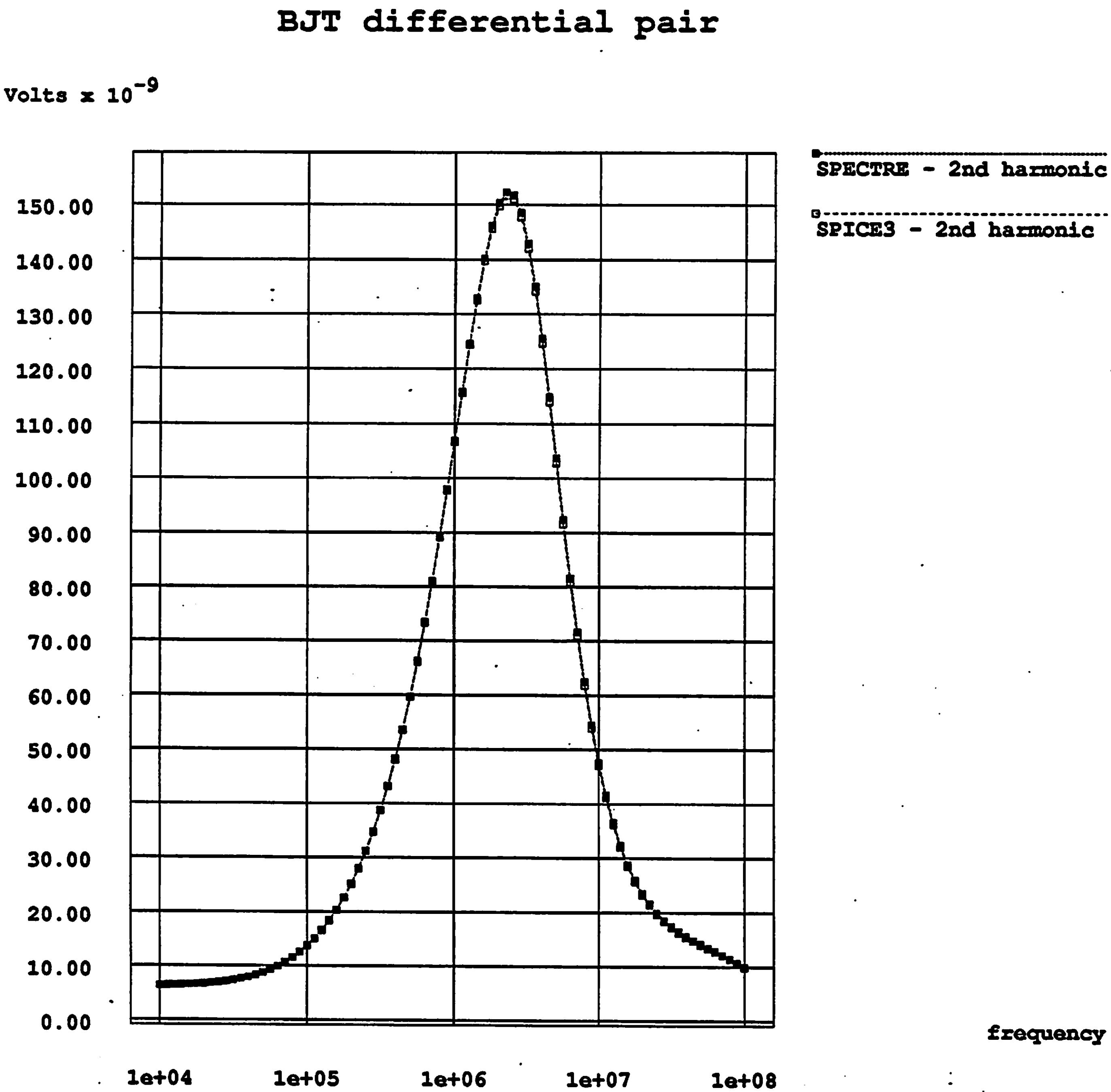

Figure D.6: BJT differential pair circuit - 2nd harmonic

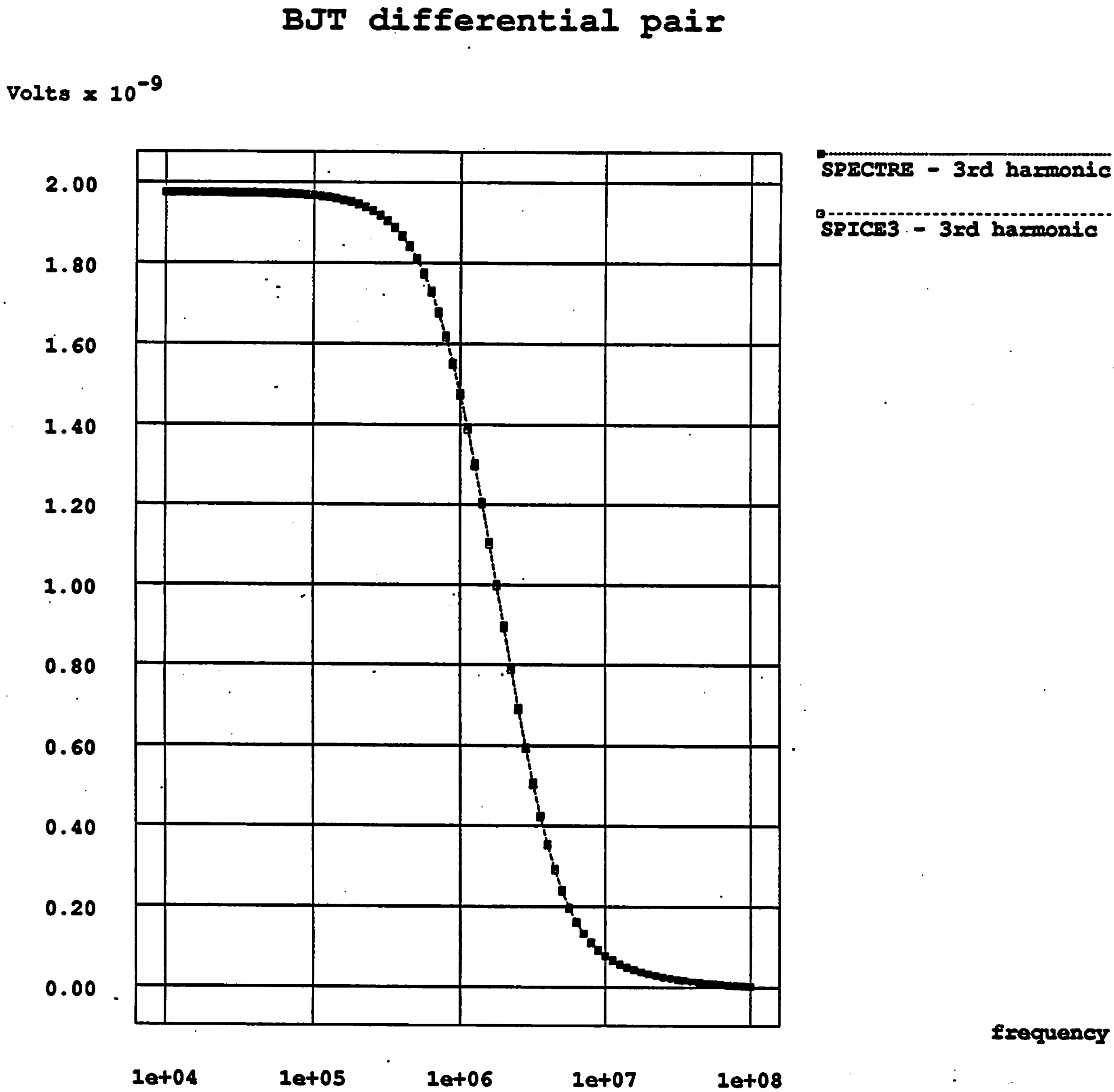

Figure D.7: BJT differential pair circuit - 3rd harmonic

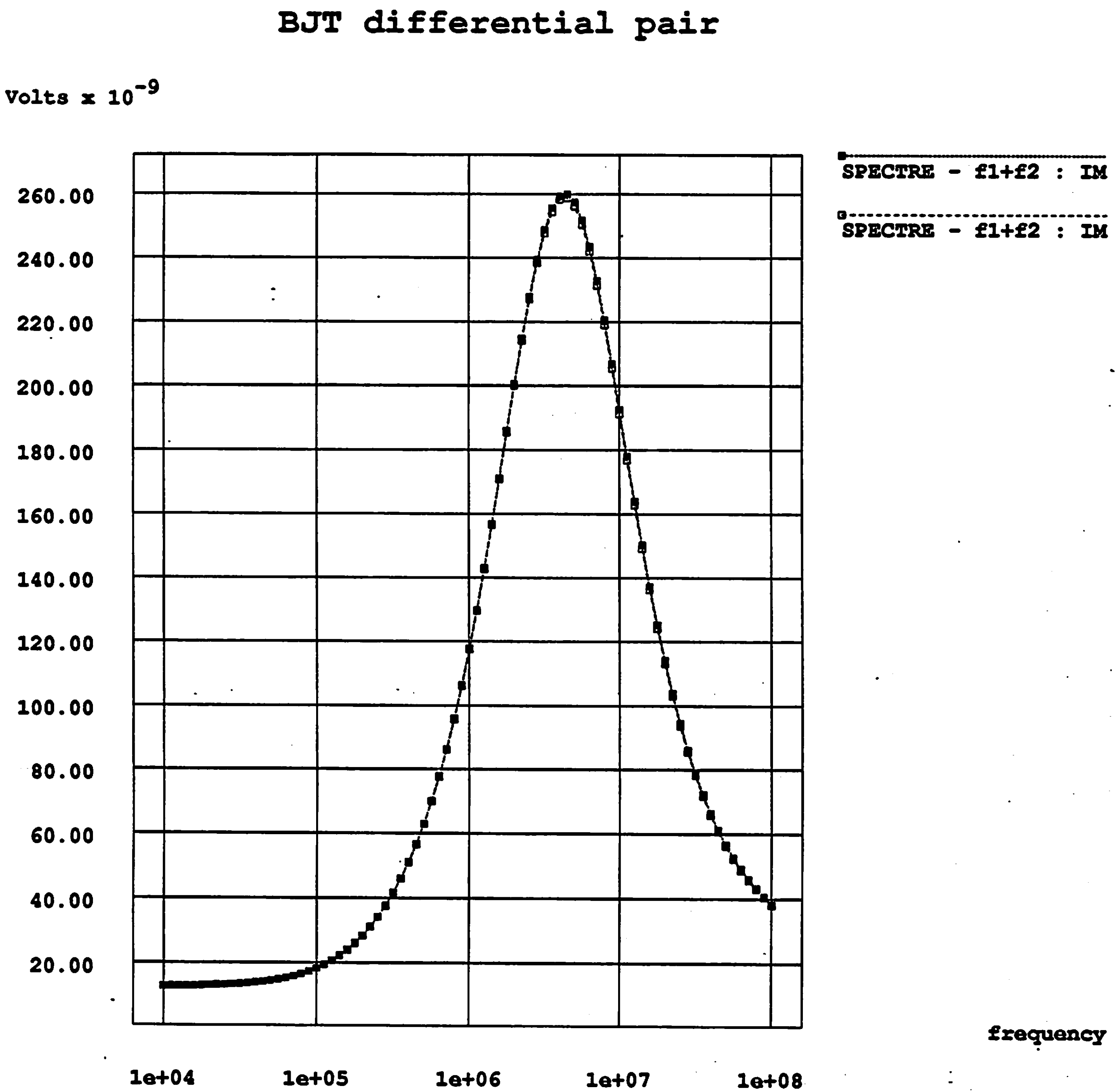

Figure D.8: BJT differential pair circuit - IM : $f_1 + f_2$

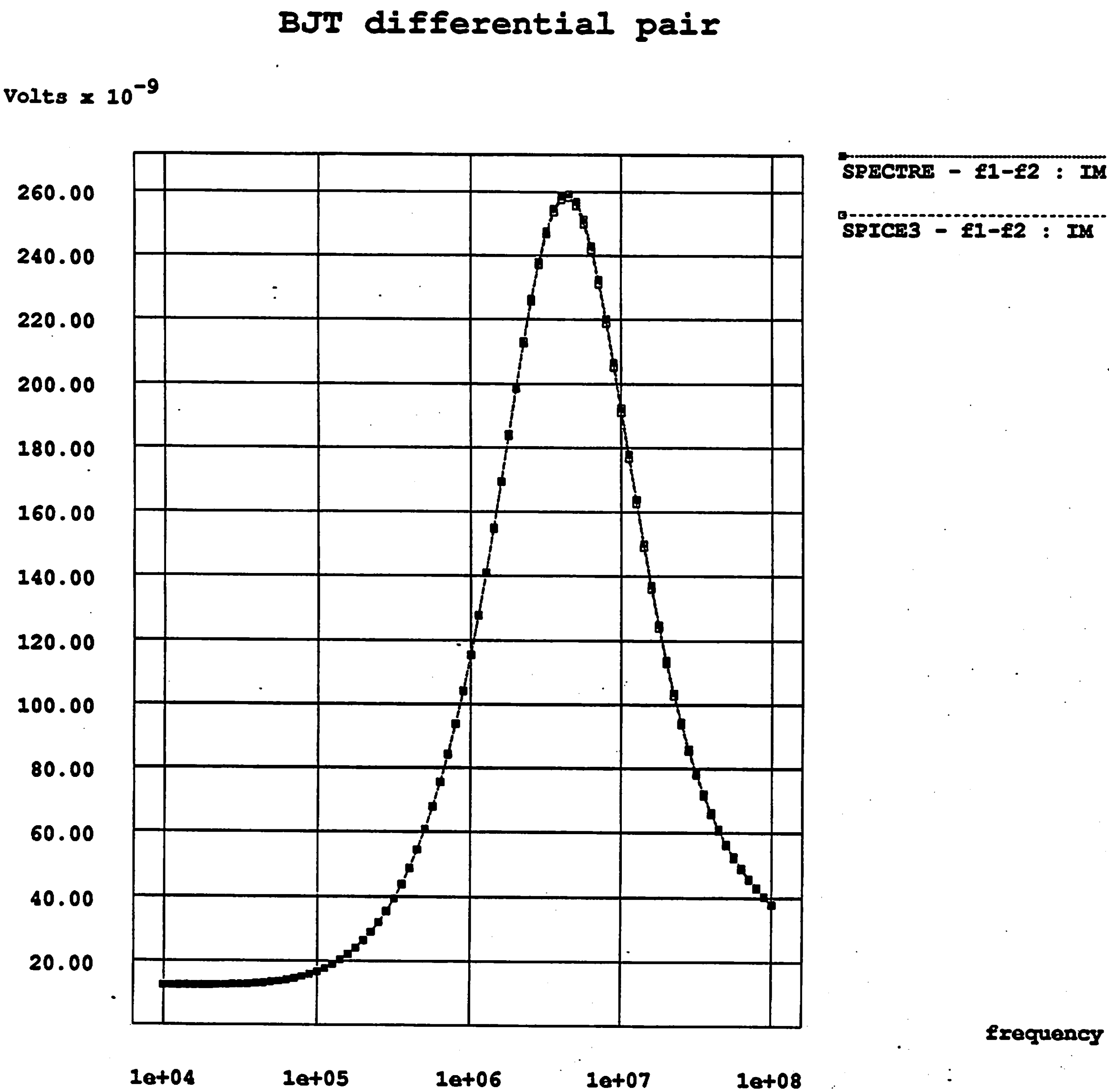

Figure D.9: BJT differential pair circuit - IM : $f_1 - f_2$

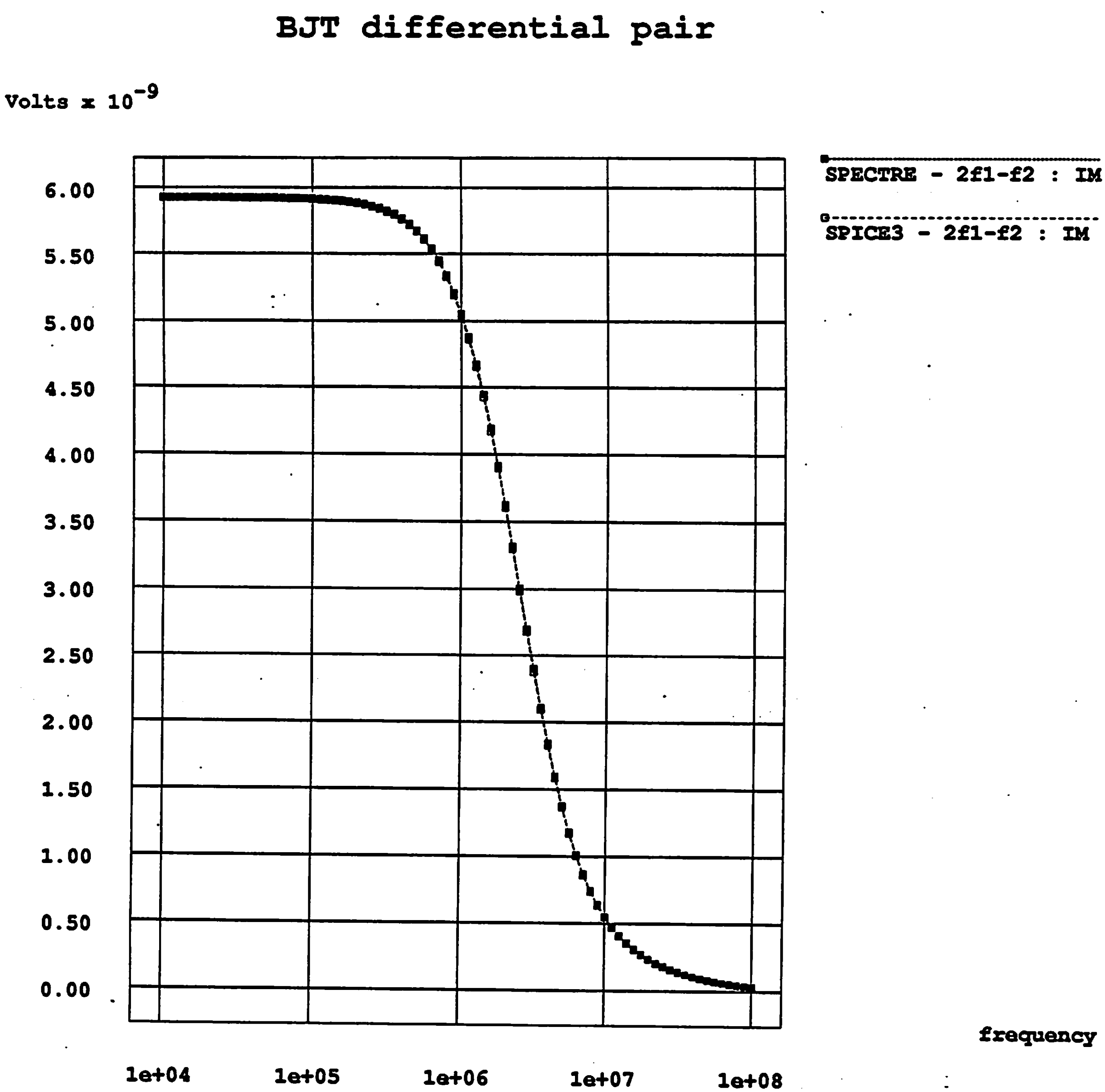

Figure D.10: BJT differential pair circuit - IM : $2f_1 - f_2$

D.3 MOSFET differential pair circuit

The circuit is a simple MOS differential pair circuit, without emitter degeneration.

The data in Table 4.1 was obtained by running the given input files with the MOS model replaced by the default level 1 model. Details are given in the input listing. In such a case (with no second order effects and no charge storage elements), the theoretical value of the second harmonic at the output is zero. The results of simulation are just the roundoff/truncation errors in numerical calculation, and were ignored for the purposes of Table 4.1.

If charge storage effects are modelled, a strong second harmonic component does arise at middle frequencies, rather surprisingly. This can be seen in the plots.

Since SPECTRE does not support MOS devices, no SPECTRE results are presented. The following plots illustrate the difference between the outputs for the level = 1,2,3 MOS models.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU time for *.disto* (with the MOS2 model) running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	1.7	-	-
Spectral	3.5	-	-

MOSFET differential pair circuit

```
v1 1 7 0 sin(0 0.01 1000) ac 0.01 distof1 0.01 distof2 0.01
v2 7 0 0
*sin(0 0.04 900)
m1 3 1 4 2 mod1 l=10u w=500u
m2 6 0 4 2 mod1 l=10u w=500u
*.model mod1 NMOS level=1
.model mod1 NMOS level=1 cbd=20pf cbs=20pf cgso=4.0e-11 cgdo=4.0e-11 cgbo=2.0e-11 cj=2.0e-4 cjsw=1.0e-9 gamma=0.37 kp=3.1e-5
*+ delta=1.0
rc1 5 3 10k
rc2 5 6 10k
vcc 5 0 10
iee 4 2 1m
vee 2 0 -10
*.tran 10us 2msec
*.disto dec 1 1000 1000
*.disto dec 20 1.0e4 1.0e8
*.disto dec 20 1.0e4 1.0e8 0.9
*the output is v(6)
.end
```

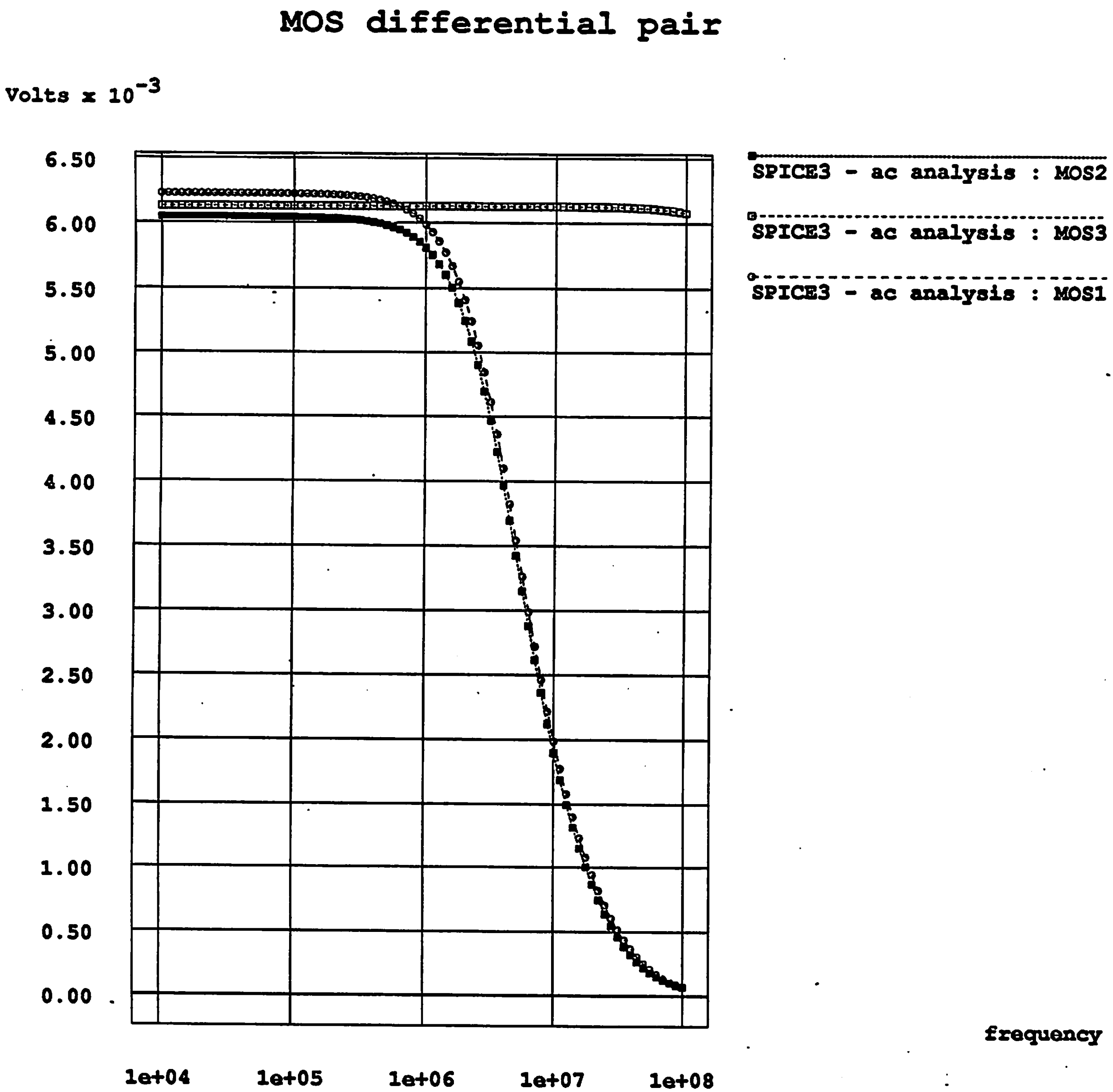

Figure D.11: MOS differential pair circuit - ac analysis

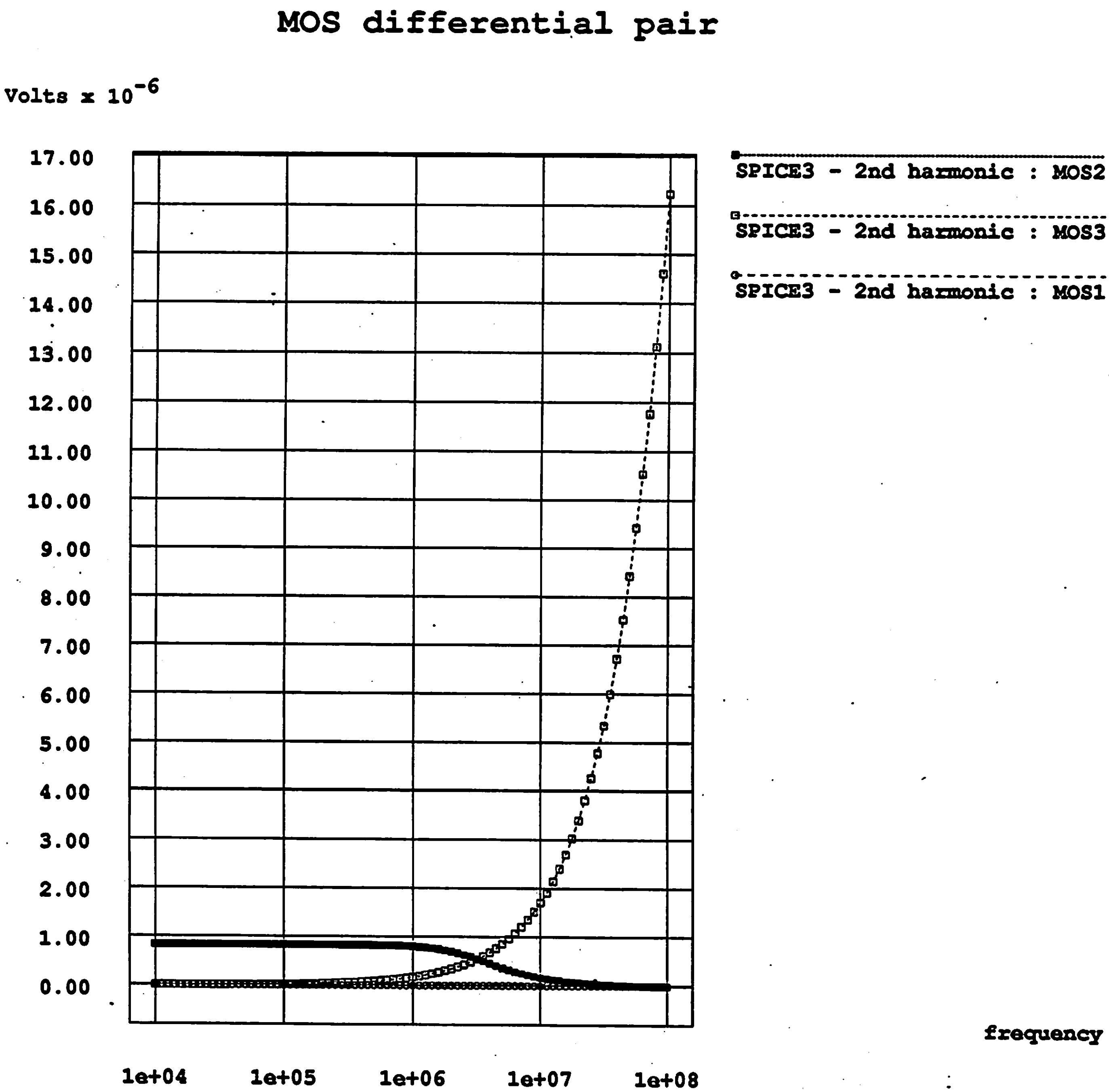

Figure D.12: MOS differential pair circuit - 2nd harmonic

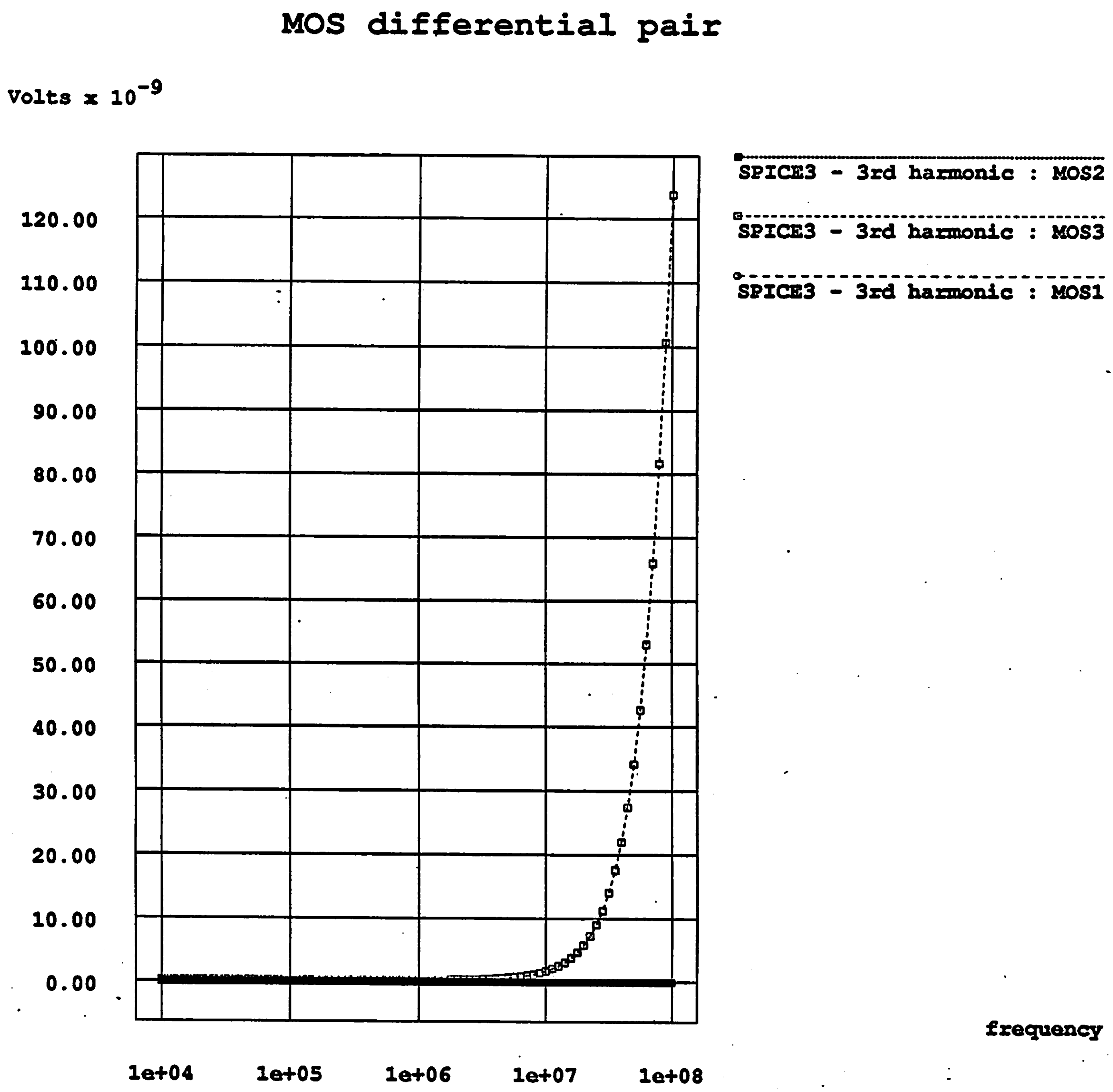

Figure D.13: MOS differential pair circuit - 3rd harmonic

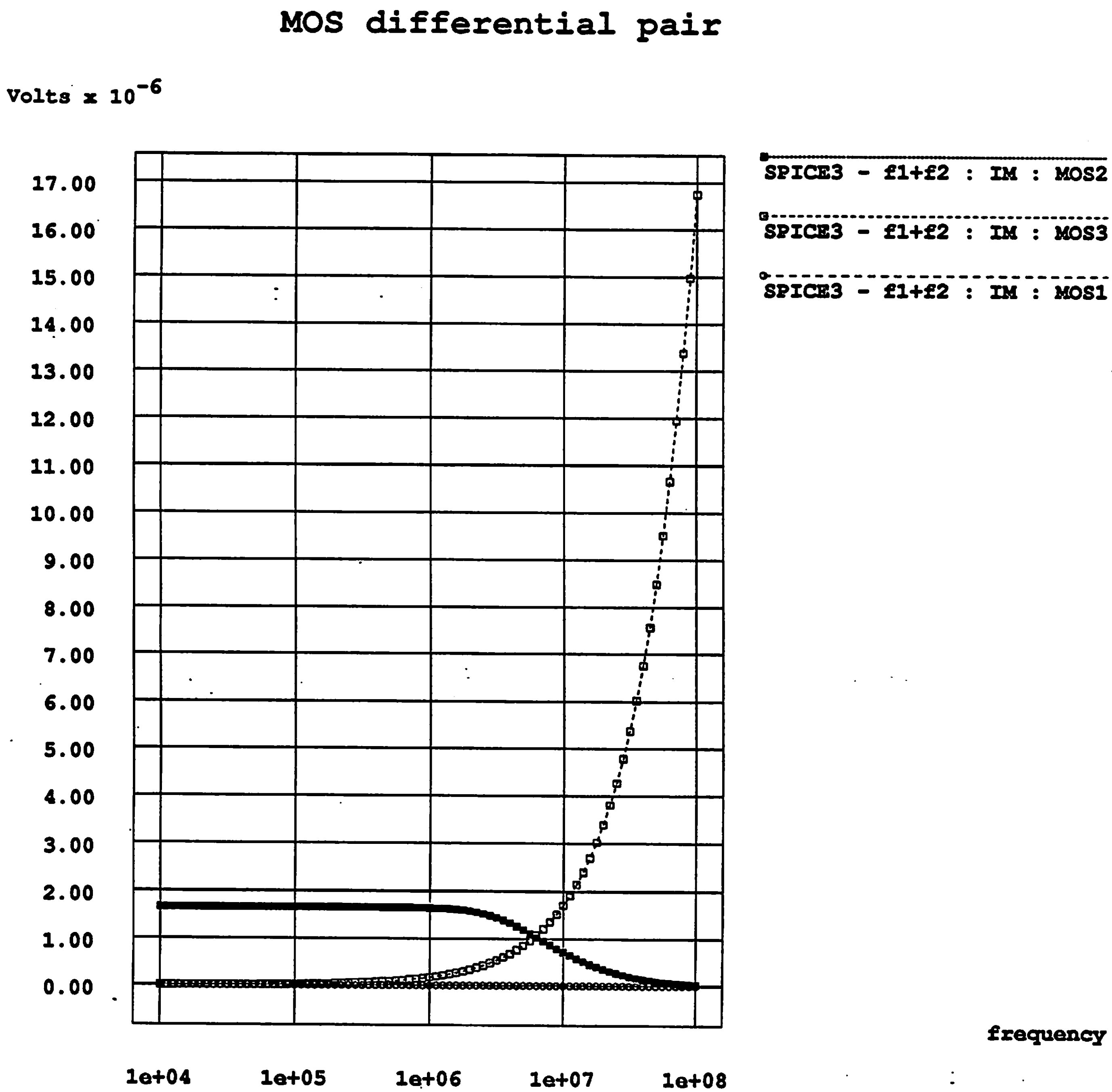

Figure D.14: MOS differential pair circuit - IM : $f_1 + f_2$

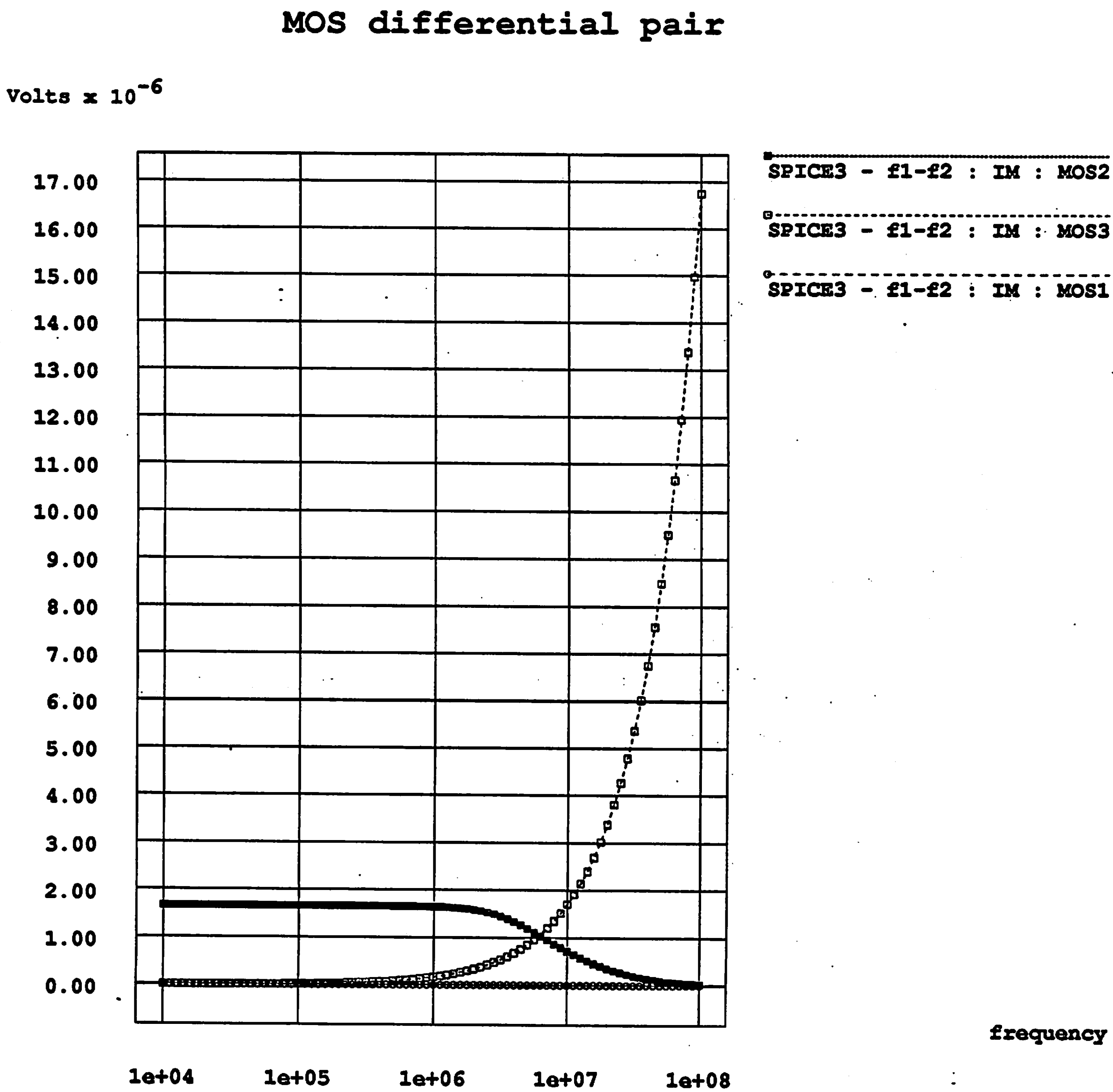

Figure D.15: MOS differential pair circuit - IM : $f_1 - f_2$

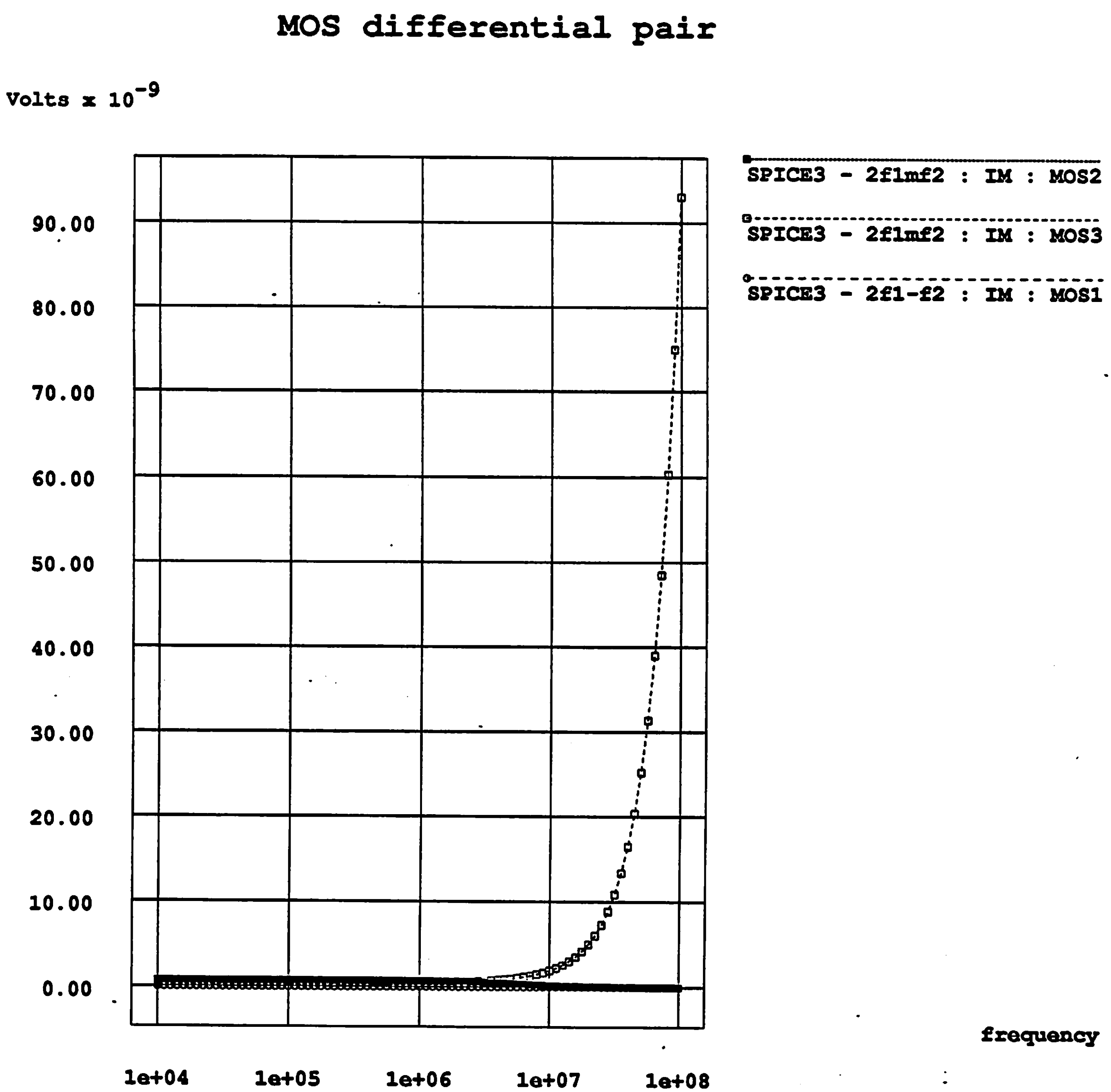

Figure D.16: MOS differential pair circuit - IM : $2f_1 - f_2$

D.4 JFET single stage circuit

The circuit is a simple JFET-resistor single-stage circuit.

The data in Table 4.1 was obtained by running the given input files with the same JFET model at the single frequency point of 1Khz.

For the first frequency point in a plot starting at 1kHz or a lower frequency, SPECTRE seems to give an erroneous result; also, it takes a disproportionately long time. For subsequent frequency points in a plot, it presumably uses old values as a starting point, and arrives at more consistent results. This behaviour was not observed for the following plots, which start at 10kHz, but is reflected in Table 4.1.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for the analyses running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	0.68	6.0	9
Spectral	1.67	123.15	74

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: 74/2 = 37

jfetsinglestage.spice3 **Fri Mar 10 21:52:37 1989** **1**

```
JFET single-stage circuit

j1 3 1 2 jf1
vcc 4 0 5v
*re 2 0 1k
vdum 2 0 0v
rl 4 3 5k
vin 1 0 0.0v ac 0.01 sin(0.0 0.01 1000) distof1 0.01 distof2 0.01
*.model jf1 NJF
*lambda=1.0e-4 cgs=5pf cgd=1pf pb=0.6
.model jf1 njf vto=-1 beta=1000u lambda=0.005 cgs=1pf cgd=1pf pb=0.7
*.disto dec 1 1000 1000
*.disto dec 20 1.0e4 1.0e8
*.disto dec 20 1.0e4 1.0e8 0.9
*the output is v(3)
.end
```

jfetsinglestage.spectre **Fri Jan 27 21:35:01 1989** **1**

```
; JFET single-stage circuit
global 0

;circuit
j1 3 1 2 jf1
vcc 4 0 vsource vdc=5
;re 2 0 resistor r=1k
vdum 2 0 vsource vdc=0
rl 4 3 resistor r=5k
vin 1 0 vsource vdc=0.0 mag=0.01 mag1=0.01 mag1f=0.01 mag2f=0.01
model jf1 jfet nfet=yes vto=-1 beta=1000u lambda=0.005 cgs=1p cgd=1p pb=0.7

;analyses
; the output is v(3)
;boom harmonic maxharm=3 fund=1.0e3
boom harmonic maxharm=3 start=1.0e4 stop=100.0e6 dec=20
aagh spectral fund2=0.9e4 order=3 maxharm1=3 saman=10 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e8 dec=20
```

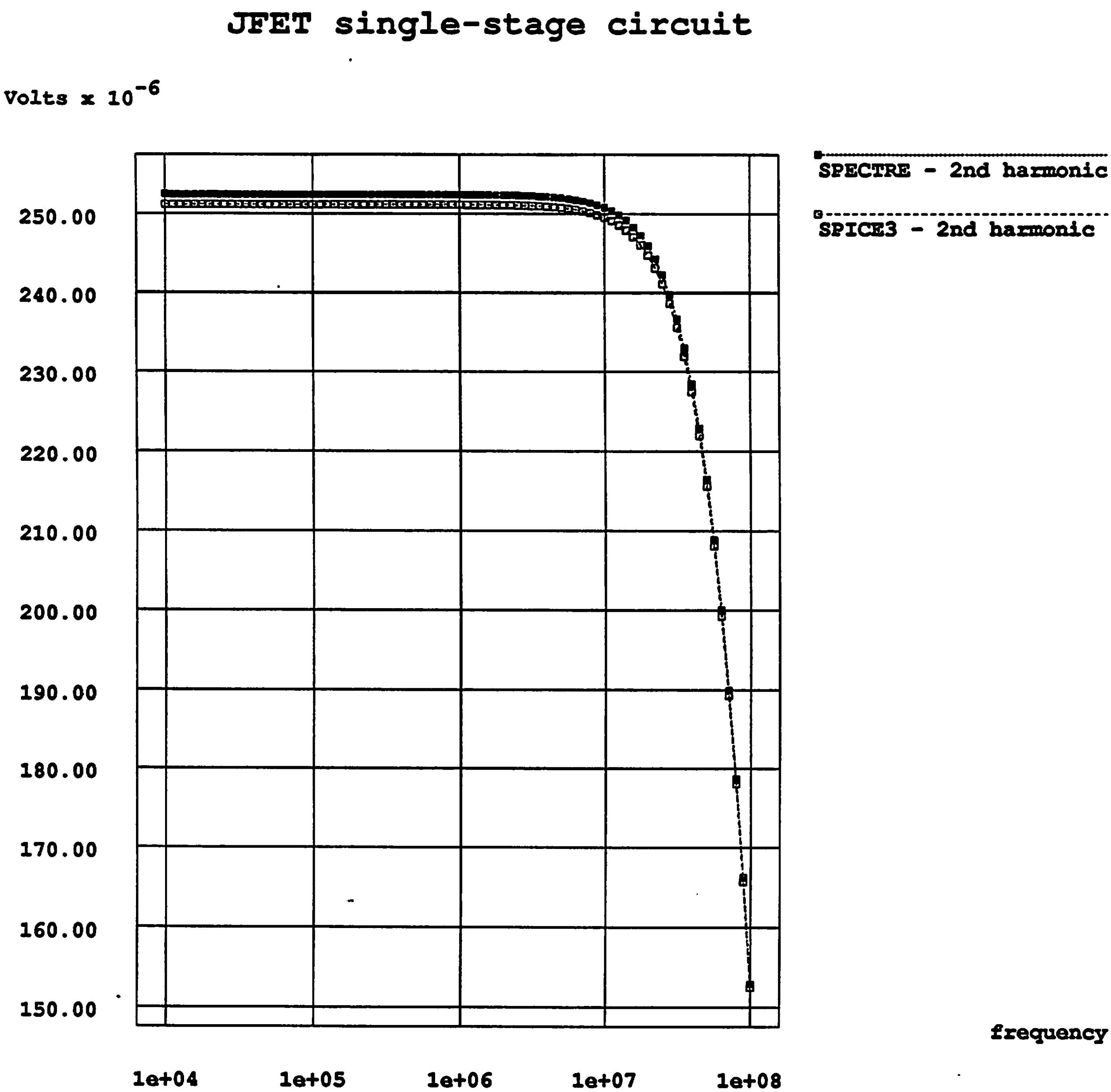

Figure D.17: JFET single-stage circuit - 2nd harmonic

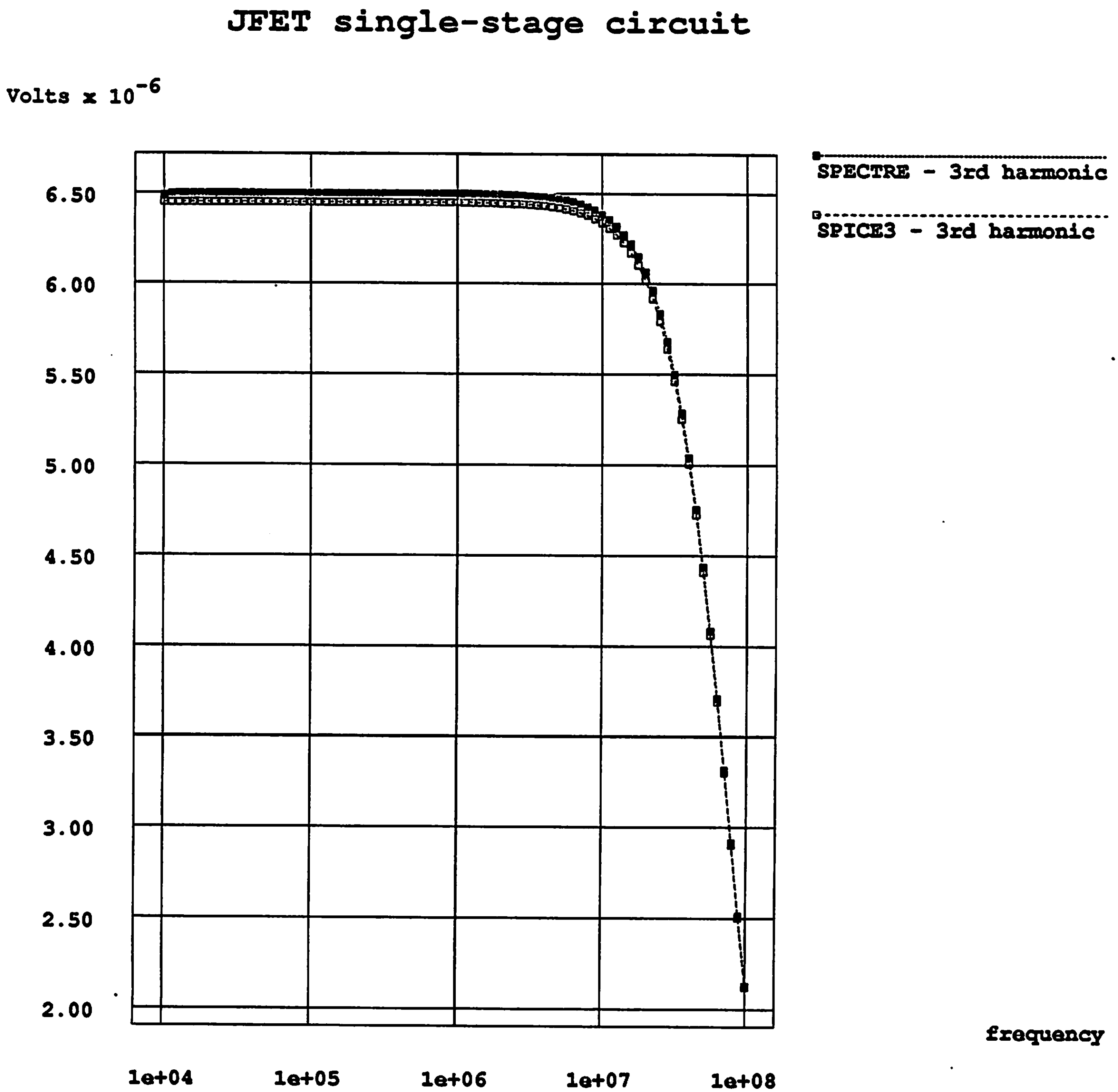

Figure D.18: JFET single-stage circuit - 3rd harmonic

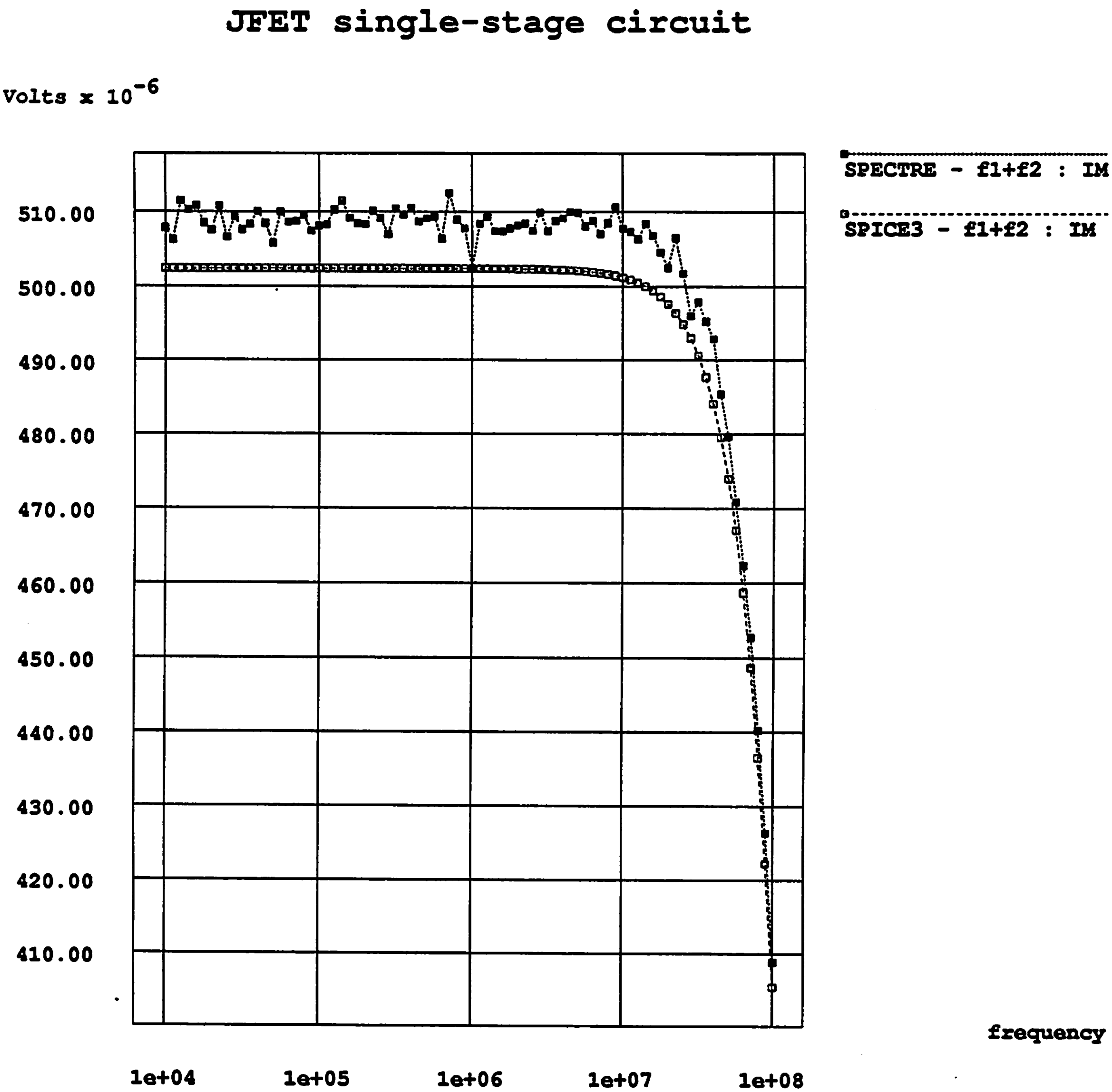

Figure D.19: JFET single-stage circuit - IM : $f_1 + f_2$

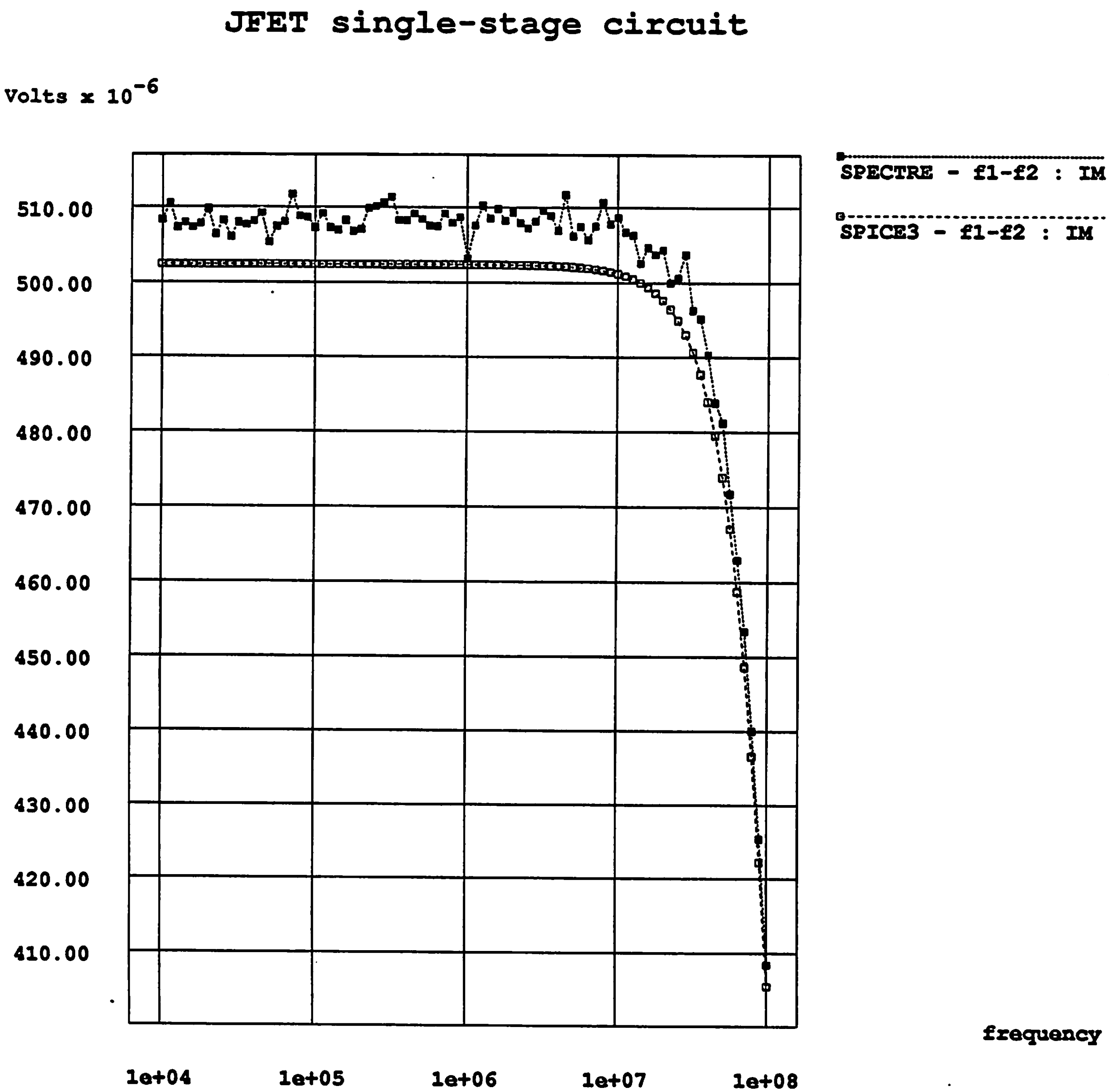

Figure D.20: JFET single-stage circuit - IM : $f_1 - f_2$

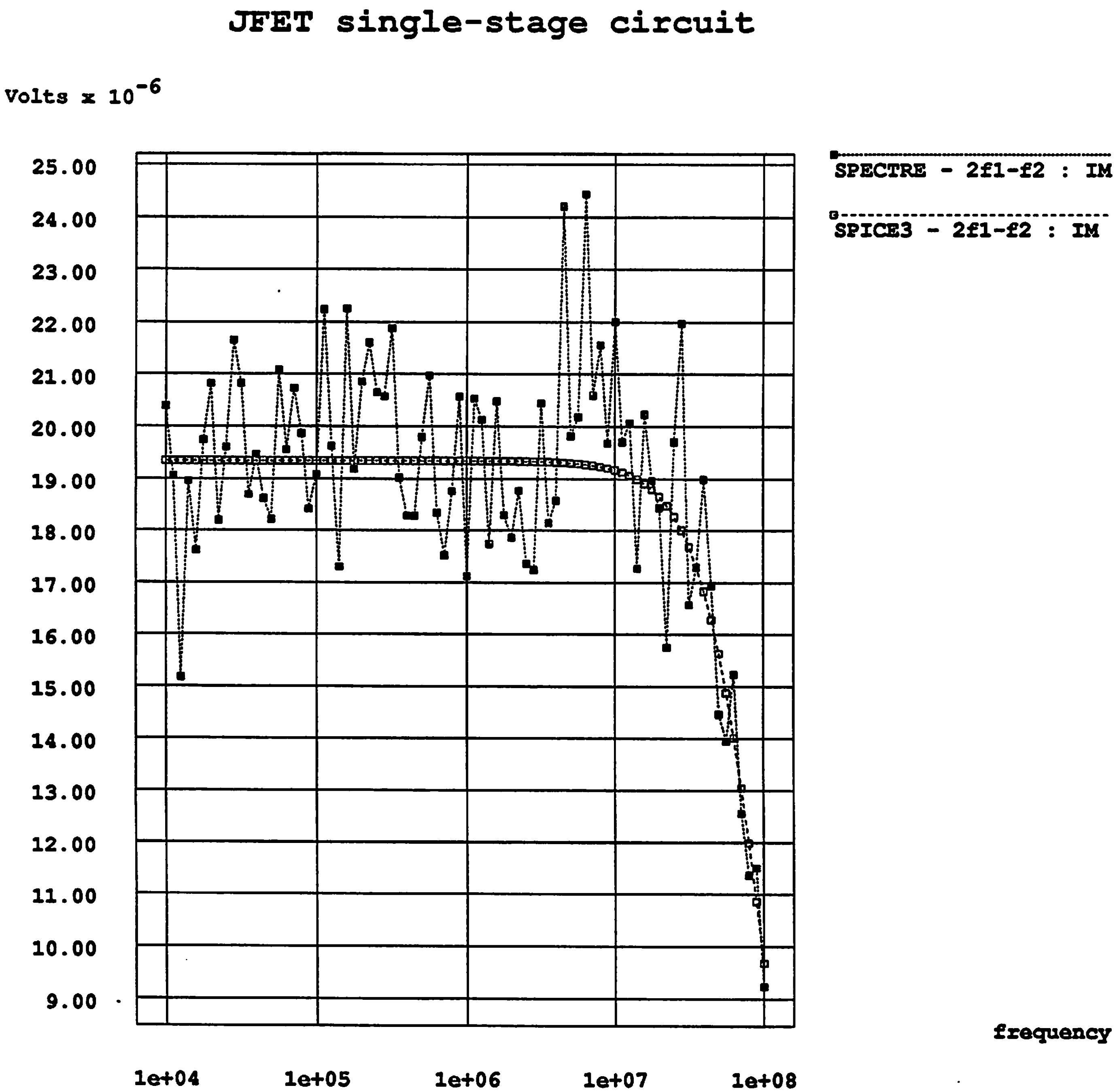

Figure D.21: JFET single-stage circuit - IM : $2f_1 - f_2$

D.5 JFET differential pair circuit

The circuit is a JFET differential pair.

The data in Table 4.1 was obtained by running the given input files with the same model but at 1kHz.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for the analyses running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	0.98	5.28	5
Spectral	1.99	172.47	86

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: 86/2 = 43

```
jfetdiffpair.spice3          Fri Mar 10 21:51:52 1989          1
```

JFET differential pair circuit

```
v1 1 0  0v distof1 0.0001 distof2 0.0001
j1 3 1 4  jf1
j2 6 0 4  jf1
rc1 5 3 10k
rc2 5 6 10k
vcc 5 0 10v
iee 4 2 1m
vee 2 0 -10v

*.model mod1 npn is=1.0e-16 bf=100
.model jf1 njf vto=-1 beta=1000u lambda=0.005 cgs=1pf cgd=1pf pb=0.7
*.disto dec 1 1000 1000
*.disto dec 20 1.0e4 1.0e8
*.disto dec 20 1.0e4 1.0e8 0.9
*the output is v(6)
.end
```

```
jfetdiffpair.spectre          Fri Jan 27 21:33:05 1989          1
```

```
; JFET differential pair circuit

; Circuit
global 0

v1 1 0 vsource vdc=0 mag=0.0001 phase=0.0 mag1=0.0001 mag1f=0.0001 mag2f=0.0001
model jf1 jfet nfet=yes vto=-1 beta=1000u lambda=0.005 cgs=1p cgd=1p pb=0.7
j1 3 1 4  jf1 region=1
j2 6 0 4  jf1 region=1
rc1 5 3 resistor r=10k
rc2 5 6 resistor r=10k
vcc 5 0 vsource vdc=10
iee 4 2 isource idc=1m
vee 2 0 vsource vdc=-10

; Analyses

;the output is v(6)
;boom harmonic maxharm=3 fund=1.0e3
boom harmonic maxharm=3 start=1.0e4 stop=1.0e8 dec=20
aagh spectral fund2=0.9e4 order=3 maxharm1=3 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e8 dec=20
```

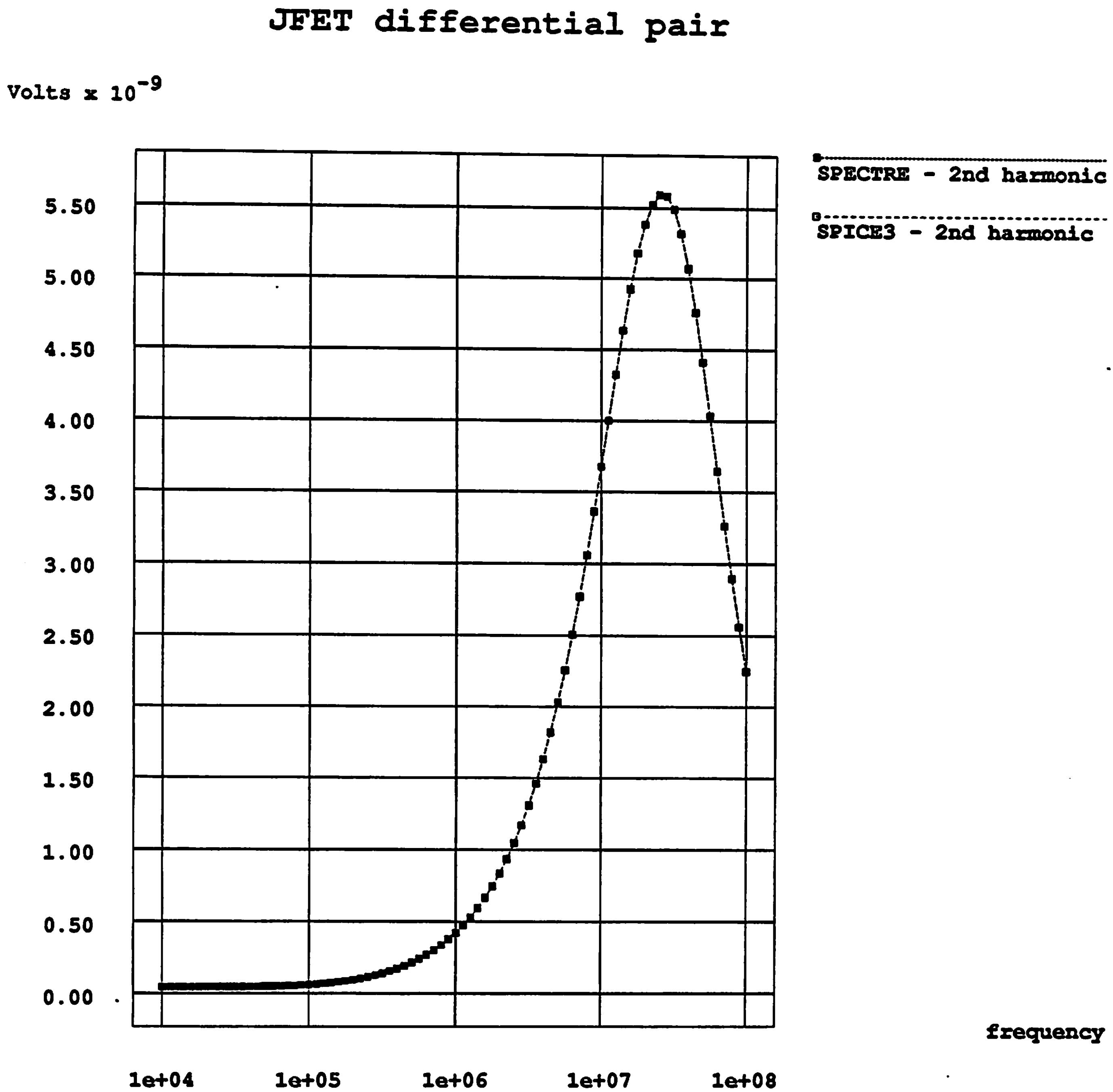

Figure D.22: JFET differential pair circuit - 2nd harmonic

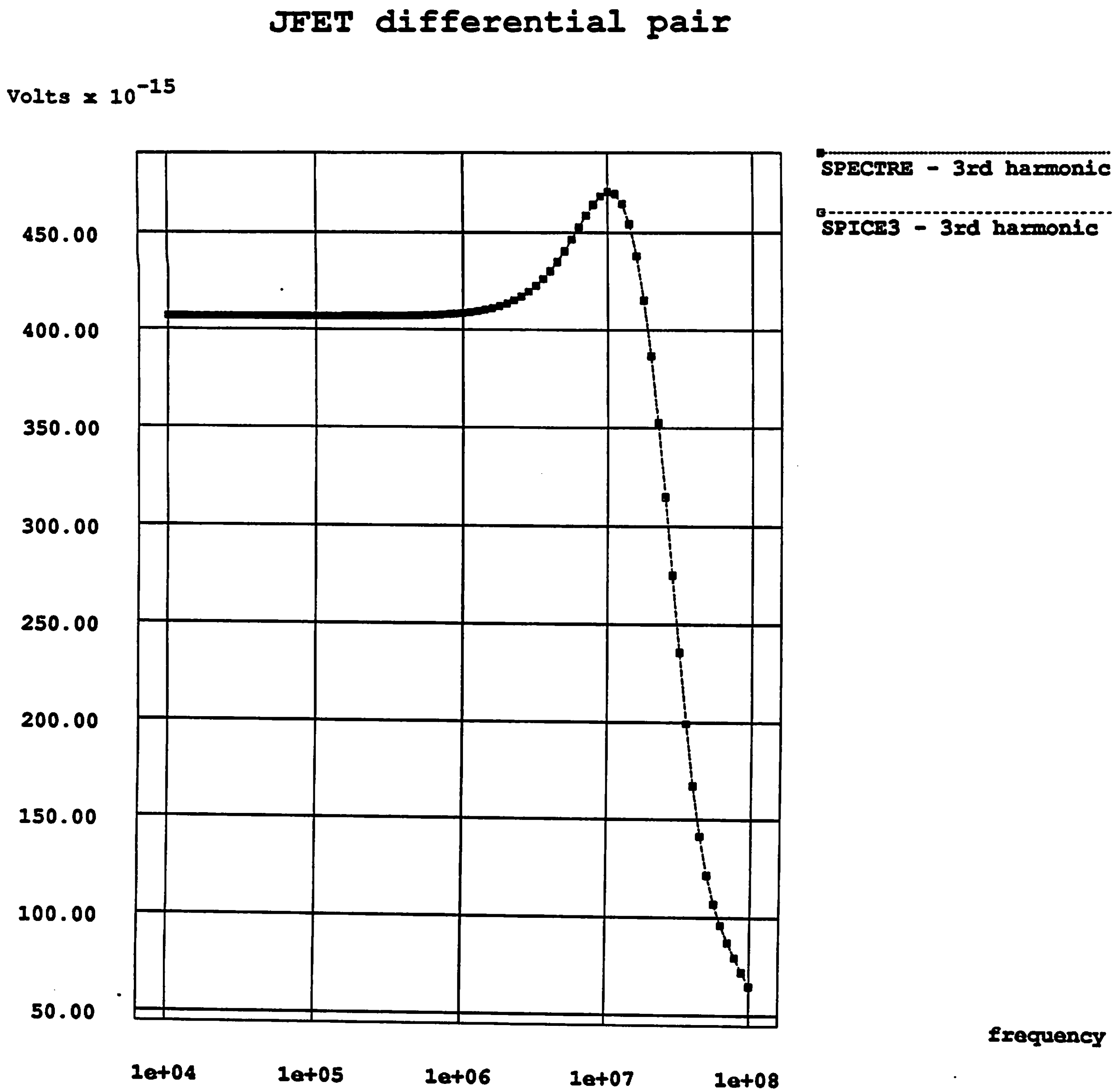

Figure D.23: JFET differential pair circuit - 3rd harmonic

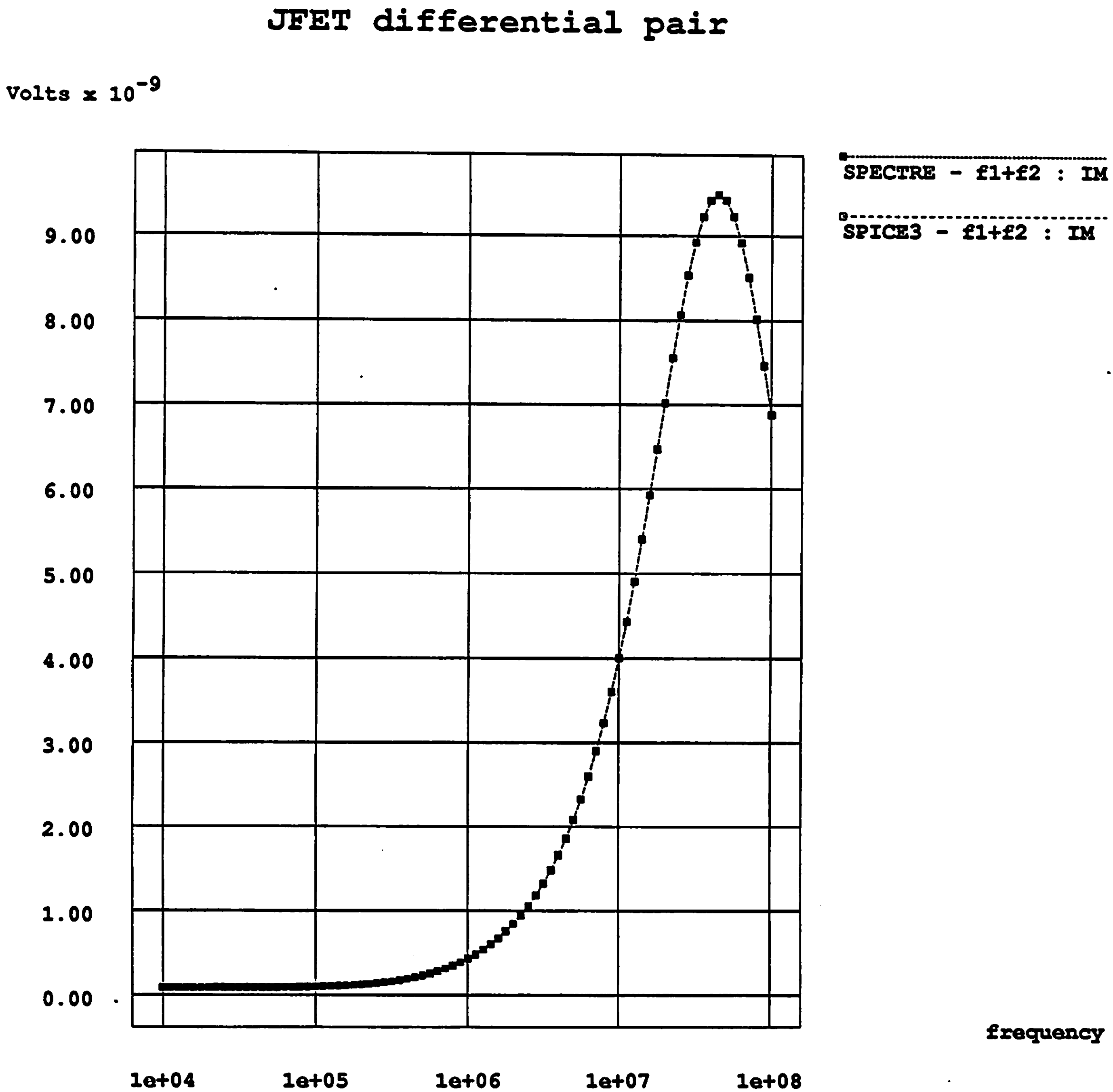

Figure D.24: JFET differential pair circuit - IM : $f_1 + f_2$

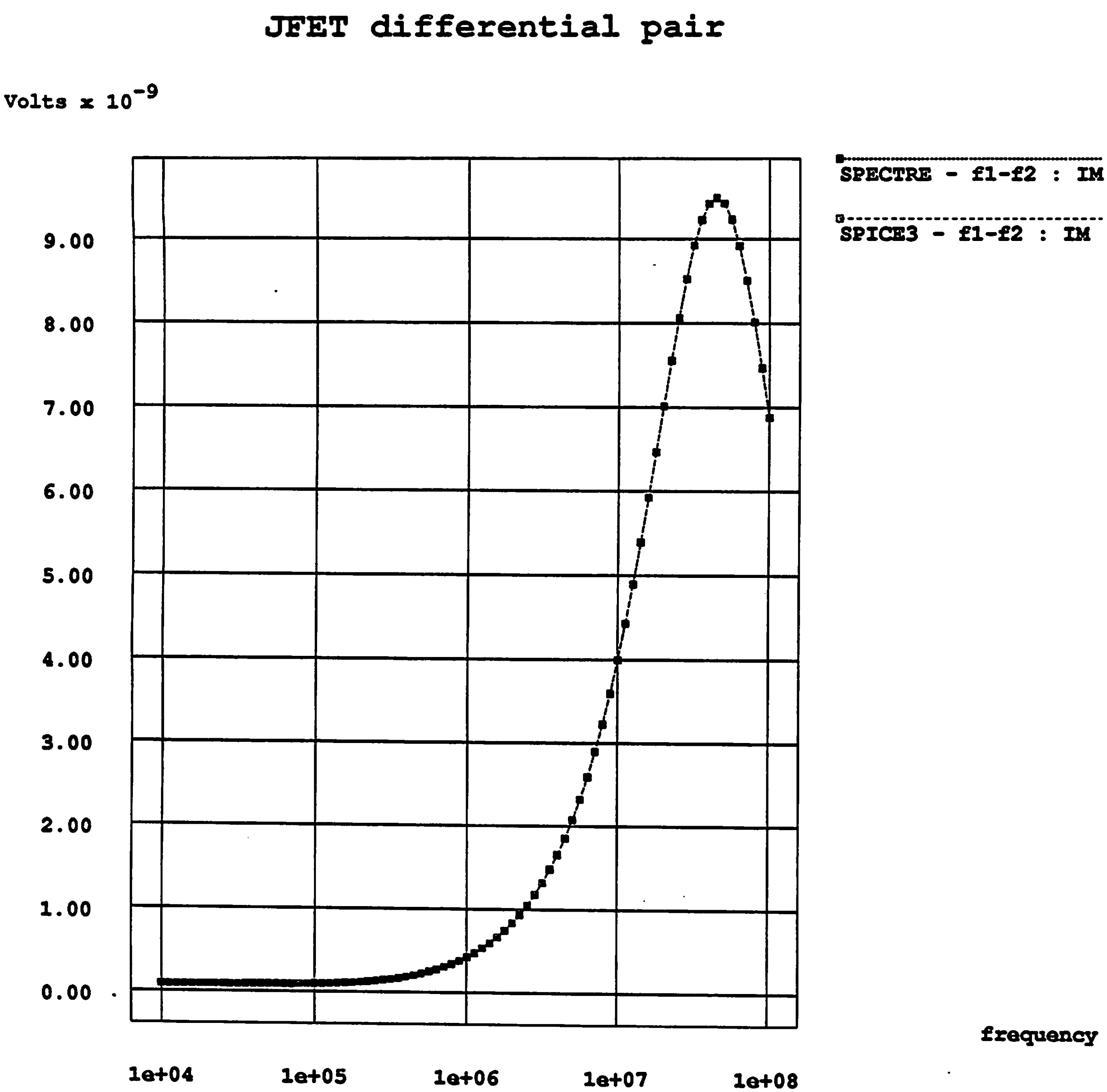

Figure D.25: JFET differential pair circuit - IM : $f_1 - f_2$

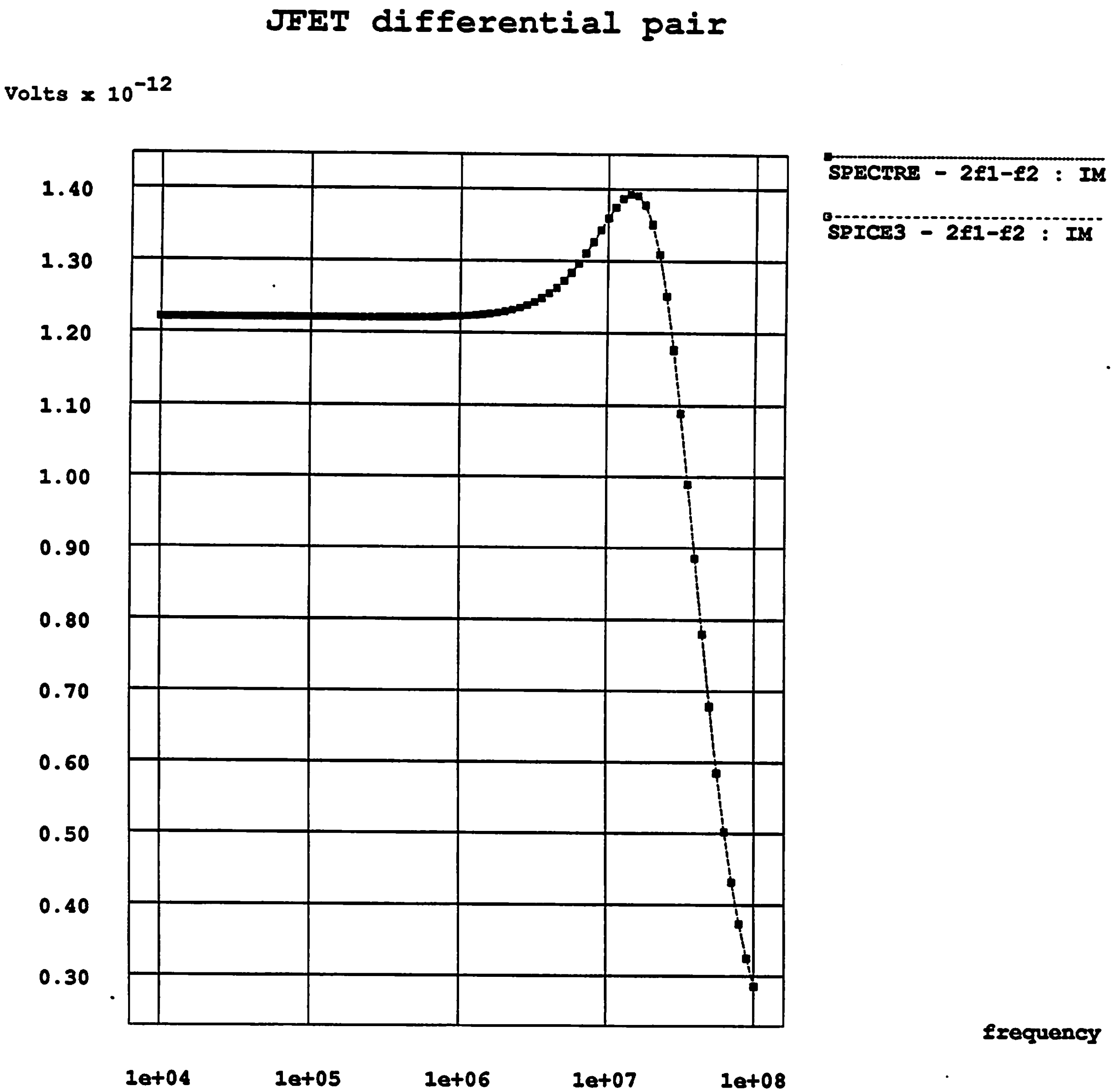

Figure D.26: JFET differential pair circuit - IM : $2f_1 - f_2$

D.6 MESFET single-stage circuit

The circuit is a simple MESFET single-stage circuit. A feature of this circuit is that there is a loop of capacitors formed, so the output does not diminish to zero as the frequency rises.

The data in Table 4.1 was obtained by running the given input files at an input frequency of 1kHz.

The SPICE3 and SPECTRE MESFET models appear to differ in the charge-storage modelling, as can be seen from the plots.

The fundamental frequency is swept from 100kHz to 1000Ghz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for the analyses running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	1.23	6.05	5
Spectral	2.4	165.23	69

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: 69/2 = 35

mesfet.spice3　　Fri Mar 10 21:53:27 1989　　1

```
MESFET single-stage circuit

.model zm1 nmf vto=-2 alpha=2 beta=2.5e-3 b=0.3 lambda=0.017
+ pb=0.6 is=1e-14 fc=0.5
+ cgd=1ff
+ cgs=5ff
* circuit
r1 2 vdd  5k
z1 2  1  0  zm1 1
c1 2  0  30ff
vdd vdd 0 dc 5
vin 1 0 0 ac 0.01 sin(0 0.01 1000) distof1 0.01 distof2 0.01
*.disto dec 1 1000 1000
*.disto dec 20 1.0e3 1.0e8 .
*.disto dec 20 1.0e3 1.0e8 0.9
* the output is v(2)
*.tran 0.1ns 10ns
.end
```

mesfet.spectre　　Fri Jan 27 21:54:28 1989　　1

```
; MESFET single-stage circuit
global 0

;models
model zm1 gaas nfet=yes vto=-2 alpha=2 beta=2.5e-3 b=0.3 lambda=0.017 cgs=5f  cgd=1f pb=0.6 is=1e-14 fc=0.5

;circuit
r1 2 vdd  resistor r=5k
z1 2  1  0  zm1 area=1
c1 2  0  capacitor c=30f
vcc vdd 0 vsource vdc=5
vin 1 0 vsource vdc=0 mag=0.01 mag1=0.01 mag1f=0.01 mag2f=0.01

;analyses

;the output is v(2)
acan ac start=1.0e4 stop=1.0e12 dec=20
;harm harmonic maxharm=3 fund=1.0e3
harm harmonic maxharm=3 start=1.0e4 stop=1.0e12 dec=20
spec spectral fund2=0.9e4 order=3 maxharm1=3 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e12 dec=20
```

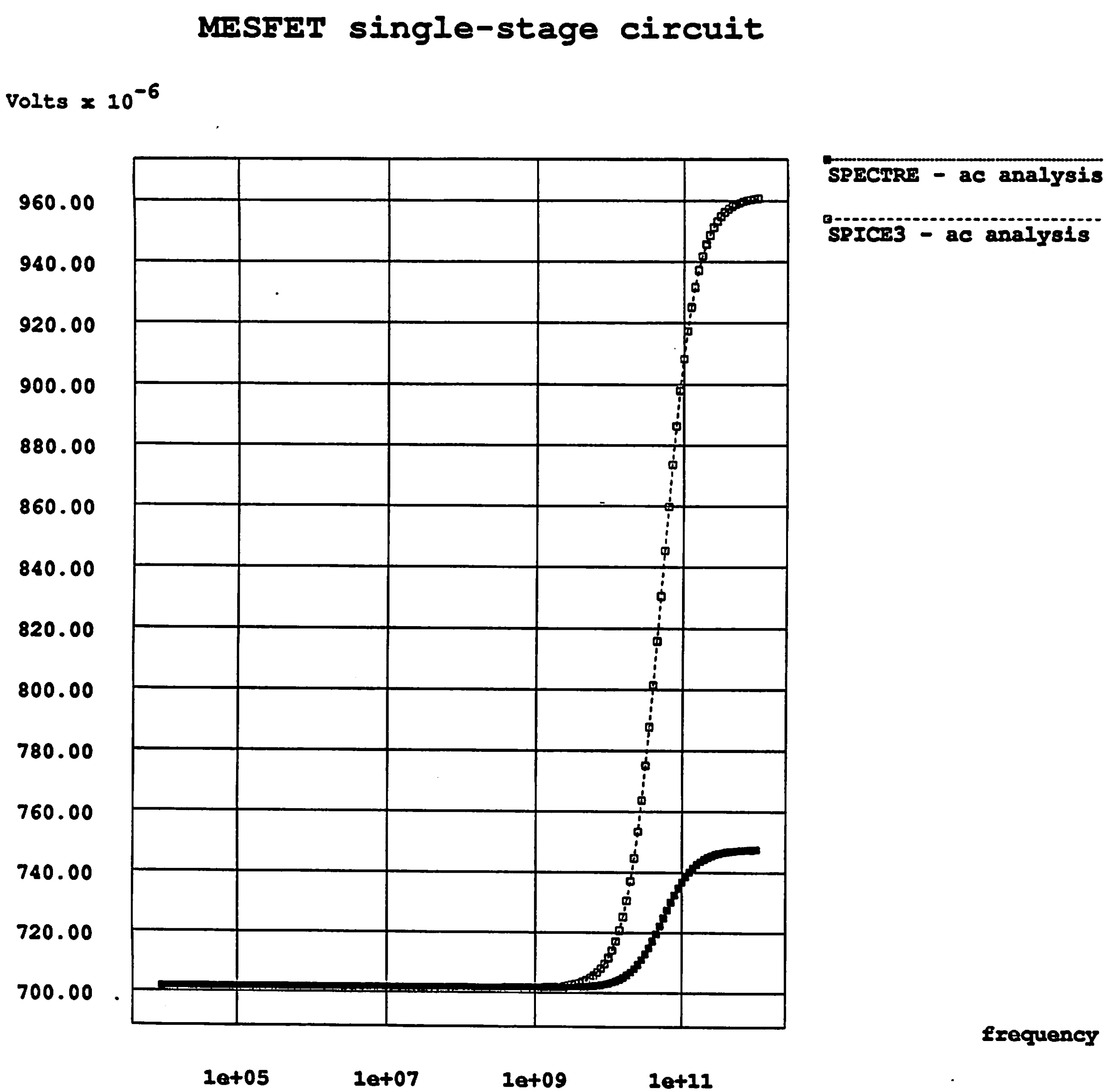

Figure D.27: MESFET single-stage circuit - ac analysis

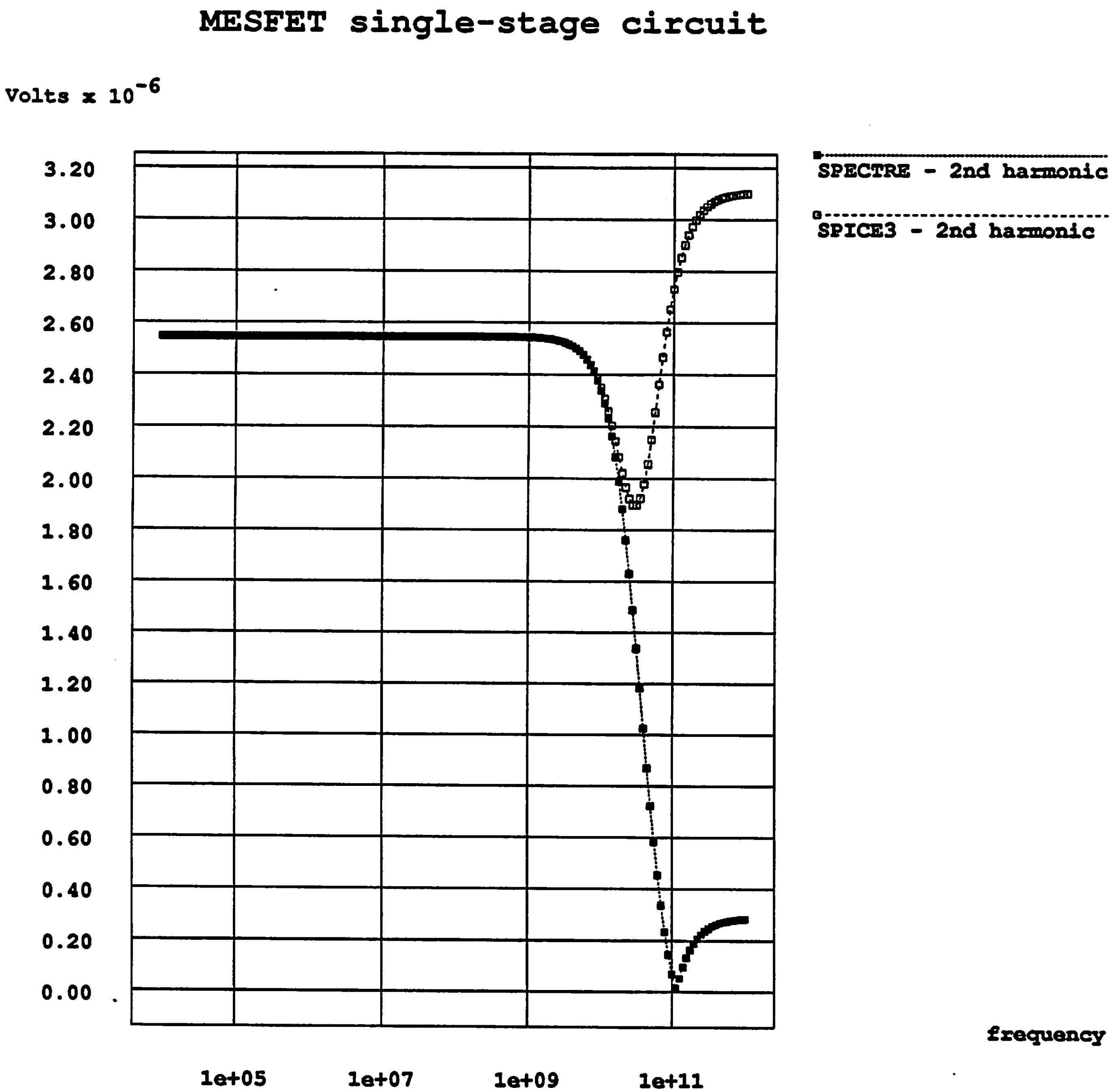

Figure D.28: MESFET single-stage circuit - 2nd harmonic

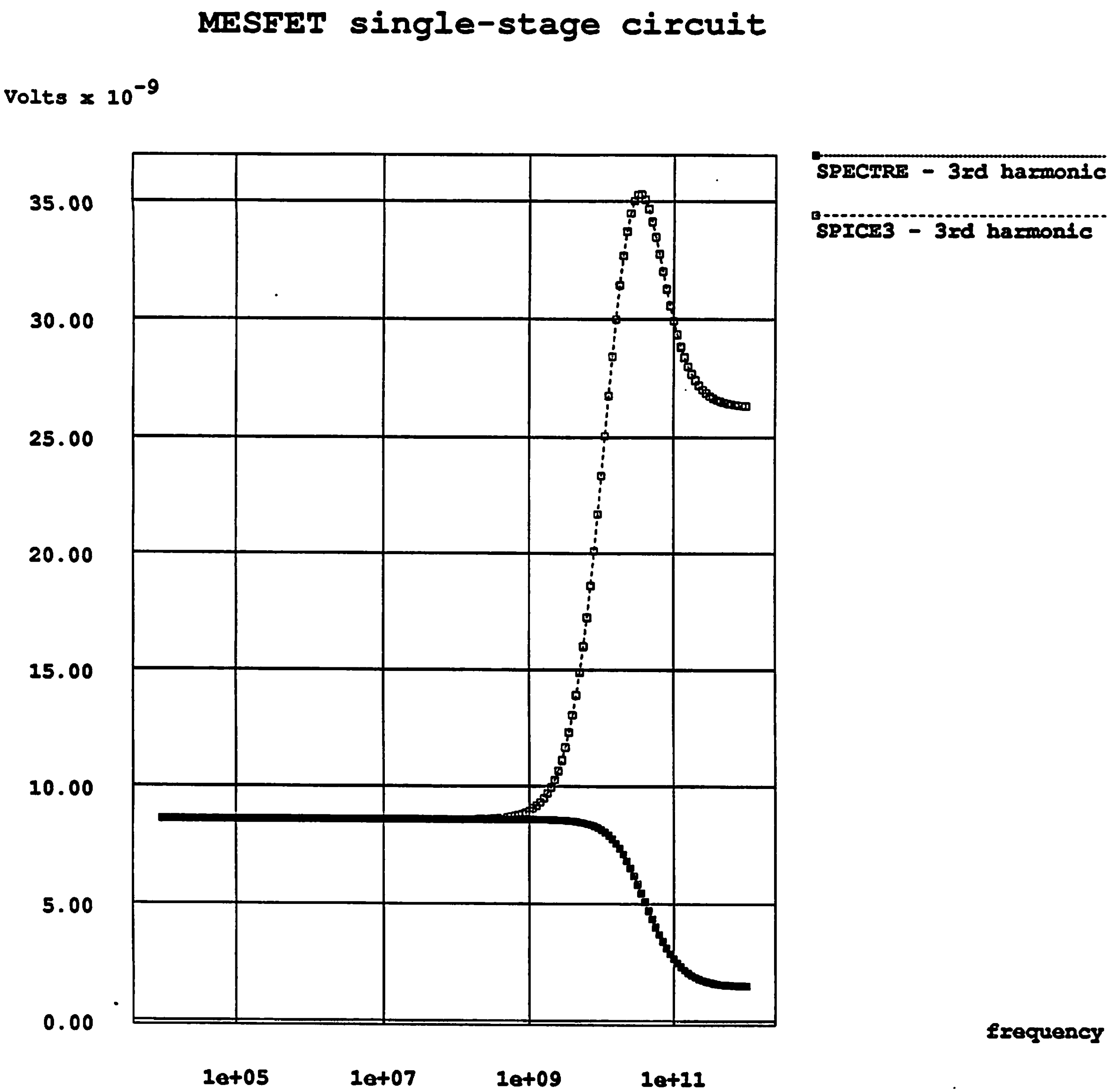

Figure D.29: MESFET single-stage circuit - 3rd harmonic

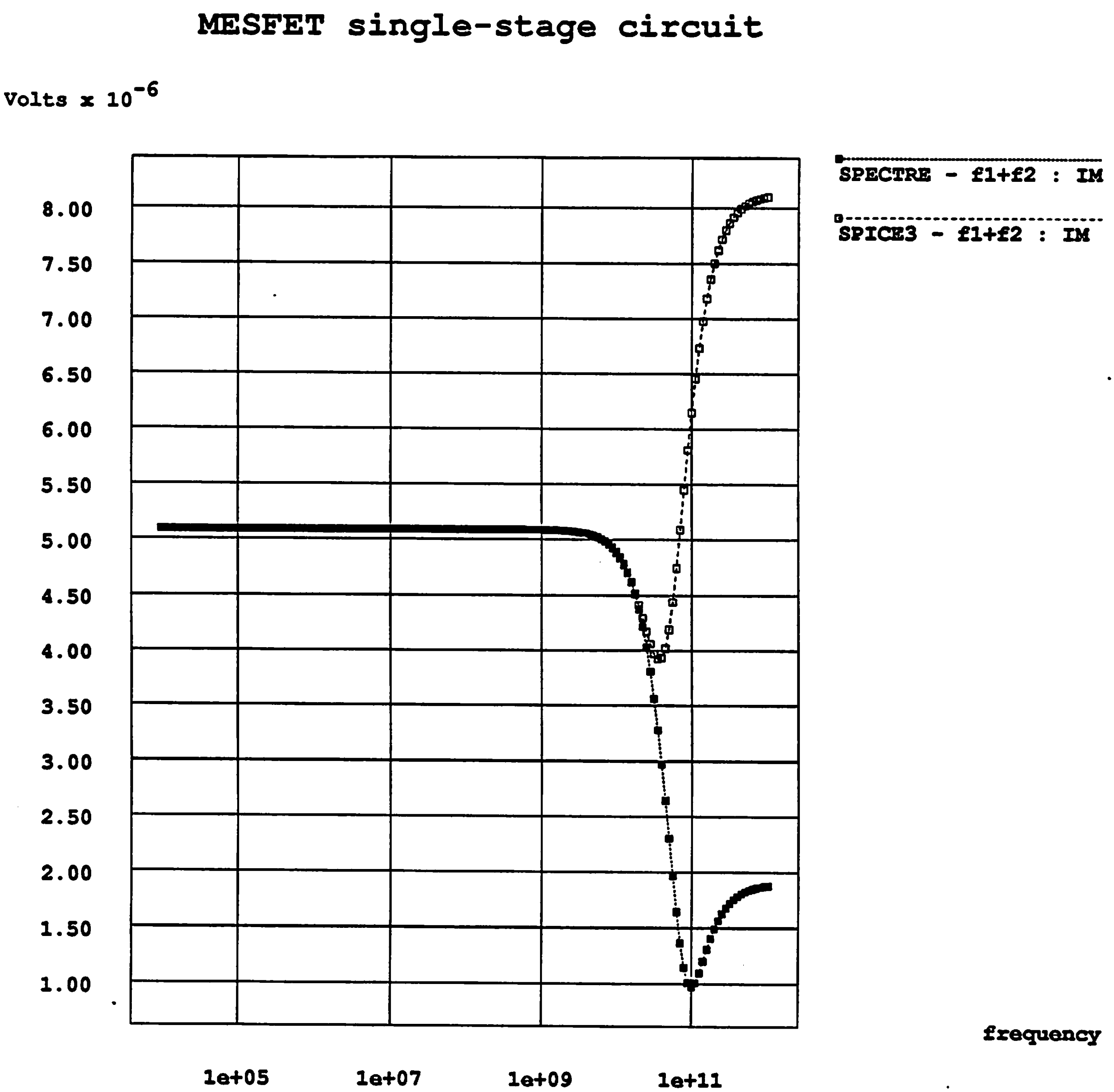

Figure D.30: MESFET single-stage circuit - IM : $f_1 + f_2$

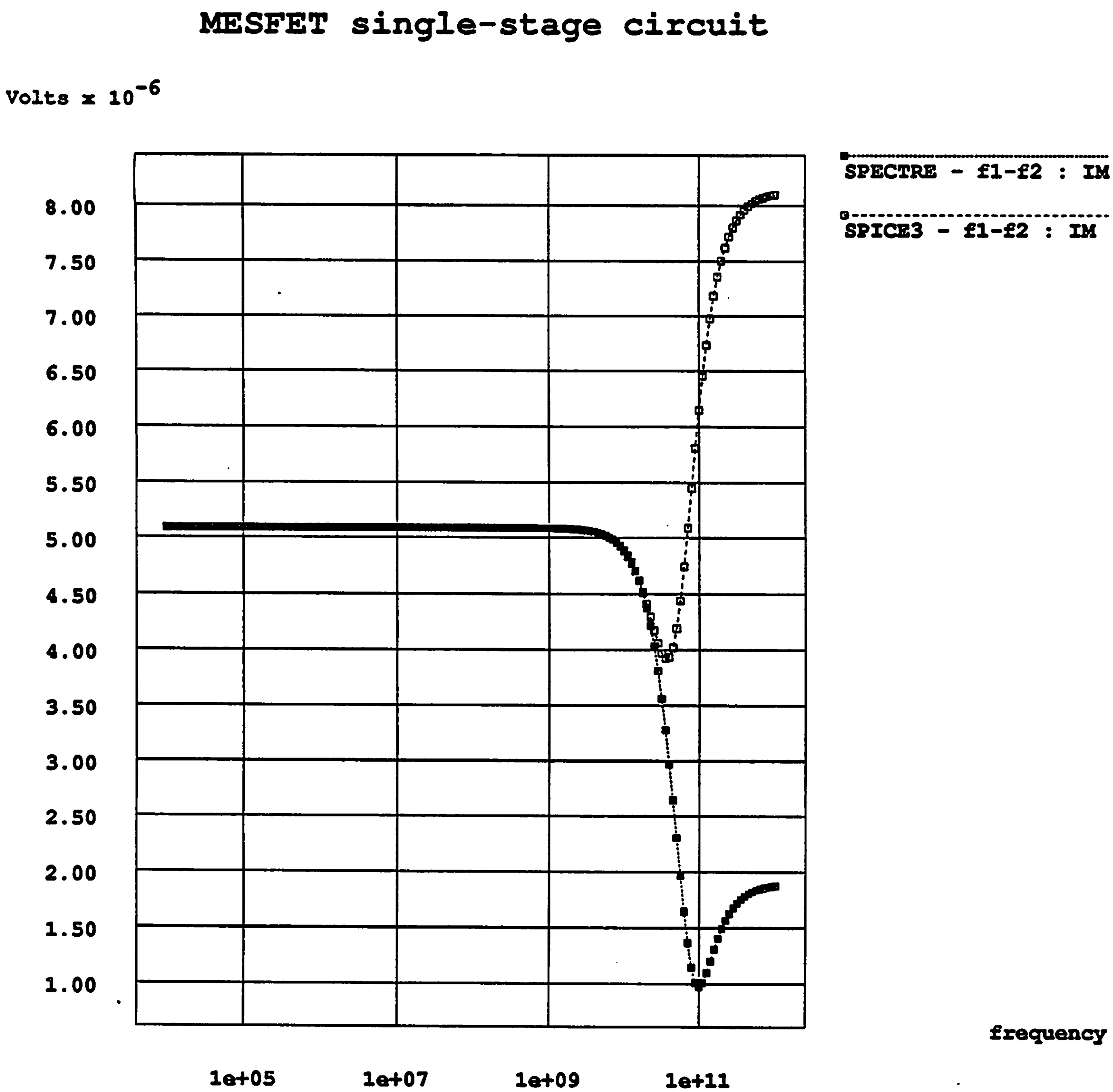

Figure D.31: MESFET single-stage circuit - IM : $f_1 - f_2$

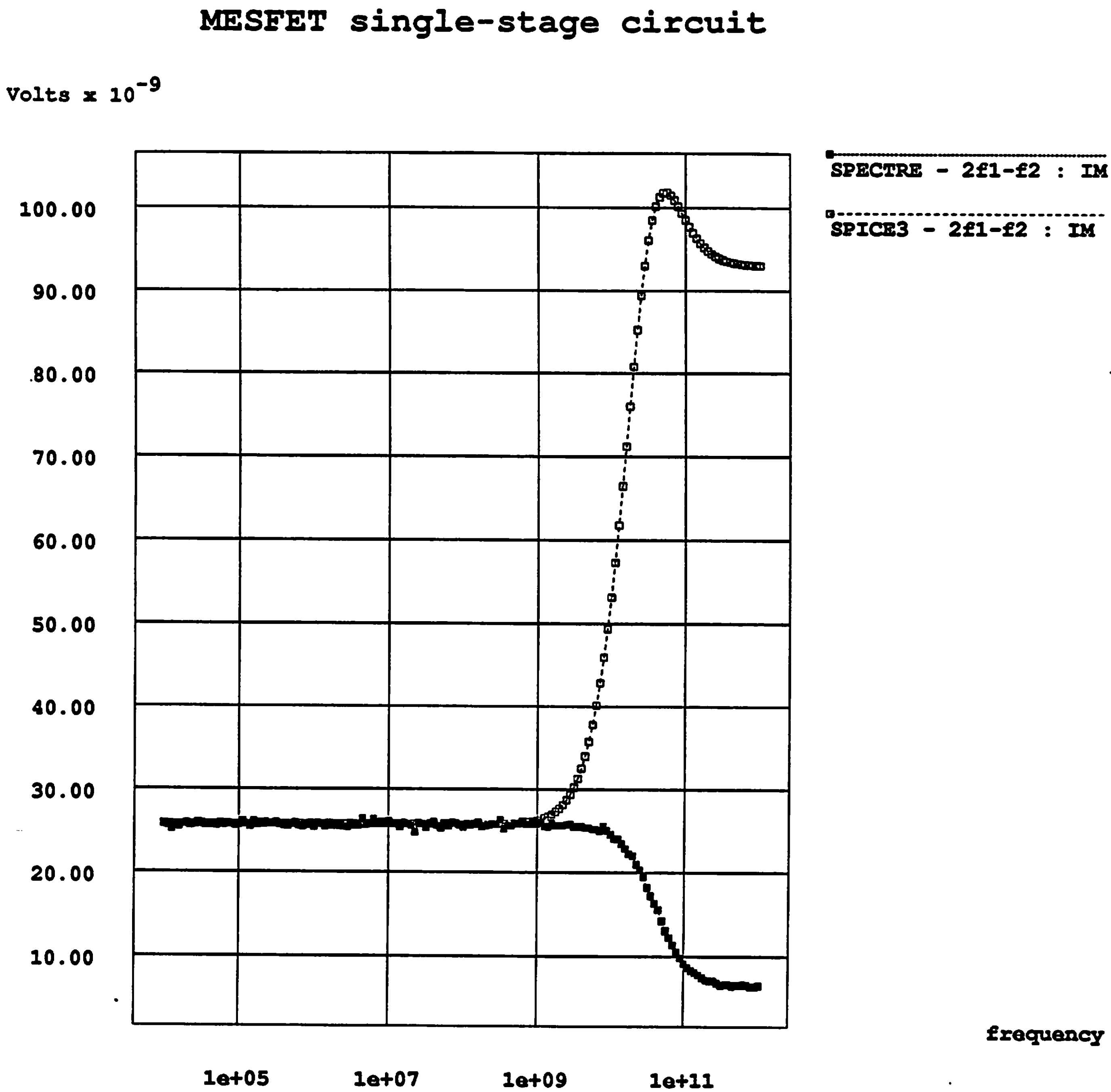

Figure D.32: MESFET single-stage circuit - IM : $2f_1 - f_2$

D.7 ua741 op-amp

The circuit is a 741 op-amp, connected with negative feedback to behave as a 100× amplifier.

The data in Table 4.1 was obtained by running the given input files at 1kHz.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for the analyses running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	12.29	121.38	10
Spectral	32.55	3769.43	120

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: 120/2 = 60

```
UA741 CKT - UA 741 OPERATIONAL AMPLIFIER

*.WIDTH IN=72
*.OPT ACCT
*.DC VIN -0.25 0.25 0.005
*.AC DEC 5 1 100MEG
*.TRAN 2.5US 250US
VCC 27 0 15
VEE 26 0 -15
VIN 30 0 0 sin(0 0.0001 1000) distof1 0.0001 distof2 0.0001
*PULSE(-0.1 0.1 0 5U 5U 80U 200U) AC 1
RS1 2 30 1K
RS2 1 0 1K
RF 24 2 100K
R1 10 26 1K
R2 9 26 50K
R3 11 26 1K
R4 12 26 3K
R5 15 17 39K
R6 21 20 40K
R7 14 26 50K
R8 18 26 50
R9 24 25 25
R10 23 24 50
R11 13 26 50K
COMP 22 8 30PF
vtest alpha 3 0
Q1 alpha 1 4 26 QNL
Q2 3 2 5 26 QNL
Q3 7 6 4 27 QPL
Q4 8 6 5 27 QPL
Q5 7 9 10 26 QNL
Q6 8 9 11 26 QNL
Q7 27 7 9 26 QNL
Q8 6 15 12 26 QNL
Q9 15 15 26 26 QNL
Q10 3 3 27 27 QPL
Q11 6 3 27 27 QPL
Q12 17 17 27 27 QPL
Q14 22 17 27 27 QPL
Q15 22 22 21 26 QNL
Q16 22 21 20 26 QNL
Q17 13 13 26 26 QNL
Q18 27 8 14 26 QNL
Q19 20 14 18 26 QNL
Q20 22 23 24 26 QNL
Q21 13 25 24 27 QPL
Q22 27 22 23 26 QNL
Q23 26 20 25 27 QPL
.MODEL QNL NPN(BF=80 RB=100 CJS=2PF TF=0.3NS TR=6NS CJE=3PF CJC=2PF VAF=50)
.MODEL QPL PNP(BF=10 RB=20 TF=1NS TR=20NS CJE=6PF CJC=4PF VAF=50)
*.disto dec 1 1000 1000
*.disto dec 20 1.0e4 1.0e8
*.disto dec 20 1.0e4 1.0e8 0.9
* the output is v(24)

*.PRINT DC V(8) V(24)
*.PLOT DC V(24) V(8)
*.PRINT AC VM(24) VP(24)
*.PLOT AC VM(24) VP(24)
*.PRINT TRAN V(8) V(24)
*.PLOT TRAN V(24) V(8)
*.width out=70
.end
```

```
;UA741 CKT - UA 741 OPERATIONAL AMPLIFIER
global 0

;models

model QNL bjt npn=yes bf=80 rb=100 cjs=2e-12 tf=0.3e-9 tr=6e-9 cje=3e-12 cjc=2e-12  vaf=50
model QPL bjt pnp=yes bf=10 rb=20 tf=1E-9 tr=20e-9 cje=6e-12 cjc=4e-12 vaf=50

;circuit

VCC 27 0 vsource vdc=15
VEE 26 0 vsource vdc=-15
VIN 30 0 vsource vdc=0 mag=0.0001 mag1=0.0001 mag1f=0.0001 mag2f=0.0001
RS1 2 30 resistor r=1K
RS2 1 0 resistor r=1K
RF 24 2 resistor r=100K
R1 10 26 resistor r=1K
R2 9 26 resistor r=50K
R3 11 26 resistor r=1K
R4 12 26 resistor r=3K
R5 15 17 resistor r=39K
R6 21 20 resistor r=40K
R7 14 26 resistor r=50K
R8 18 26 resistor r=50
R9 24 25 resistor r=25
R10 23 24 resistor r=50
R11 13 26 resistor r=50K
COMP 22 8 capacitor c=30e-12
vtest alpha 3 vsource vdc=0
Q1 alpha 1 4 26 QNL
Q2 3 2 5 26 QNL
Q3 7 6 4 27 QPL
Q4 8 6 5 27 QPL
Q5 7 9 10 26 QNL
Q6 8 9 11 26 QNL
Q7 27 7 9 26 QNL
Q8 6 15 12 26 QNL
Q9 15 15 26 26 QNL
Q10 3 3 27 27 QPL
Q11 6 3 27 27 QPL
Q12 17 17 27 27 QPL
Q14 22 17 27 27 QPL
Q15 22 22 21 26 QNL
Q16 22 21 20 26 QNL
Q17 13 13 26 26 QNL region=1
Q18 27 8 14 26 QNL
Q19 20 14 18 26 QNL
Q20 22 23 24 26 QNL
Q21 13 25 24 27 QPL
Q22 27 22 23 26 QNL
Q23 26 20 25 27 QPL

; Analyses

;the output is v(24)
;boom harmonic maxharm=3 fund=1.0e3
boom harmonic maxharm=3 start=1.0e4 stop=100.0e6 dec=20
aagh spectral fund2=0.9e4 order=3 maxharm1=3 saman=10 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e8 dec=20
```

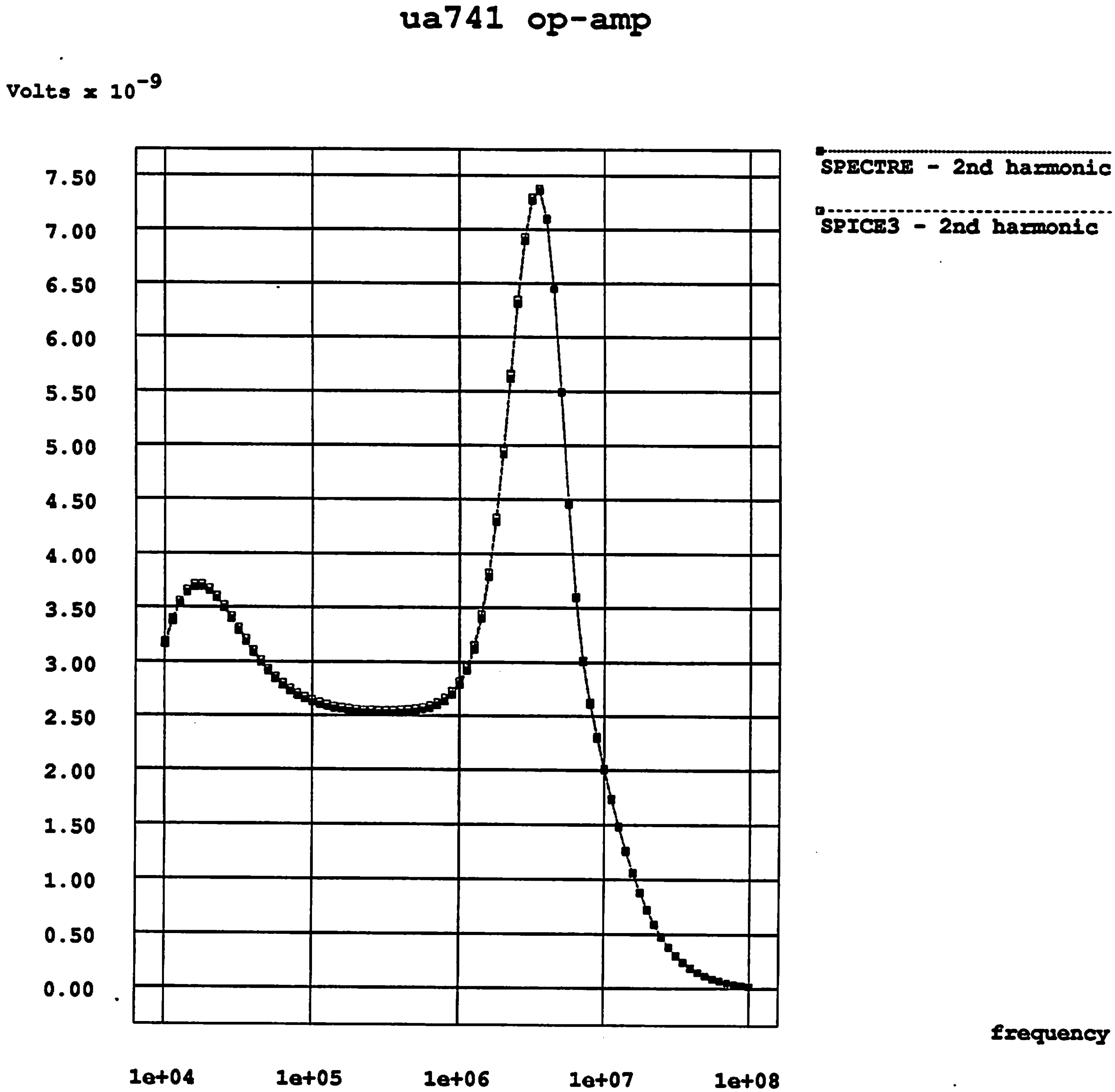

Figure D.33: ua741 op-amp - 2nd harmonic

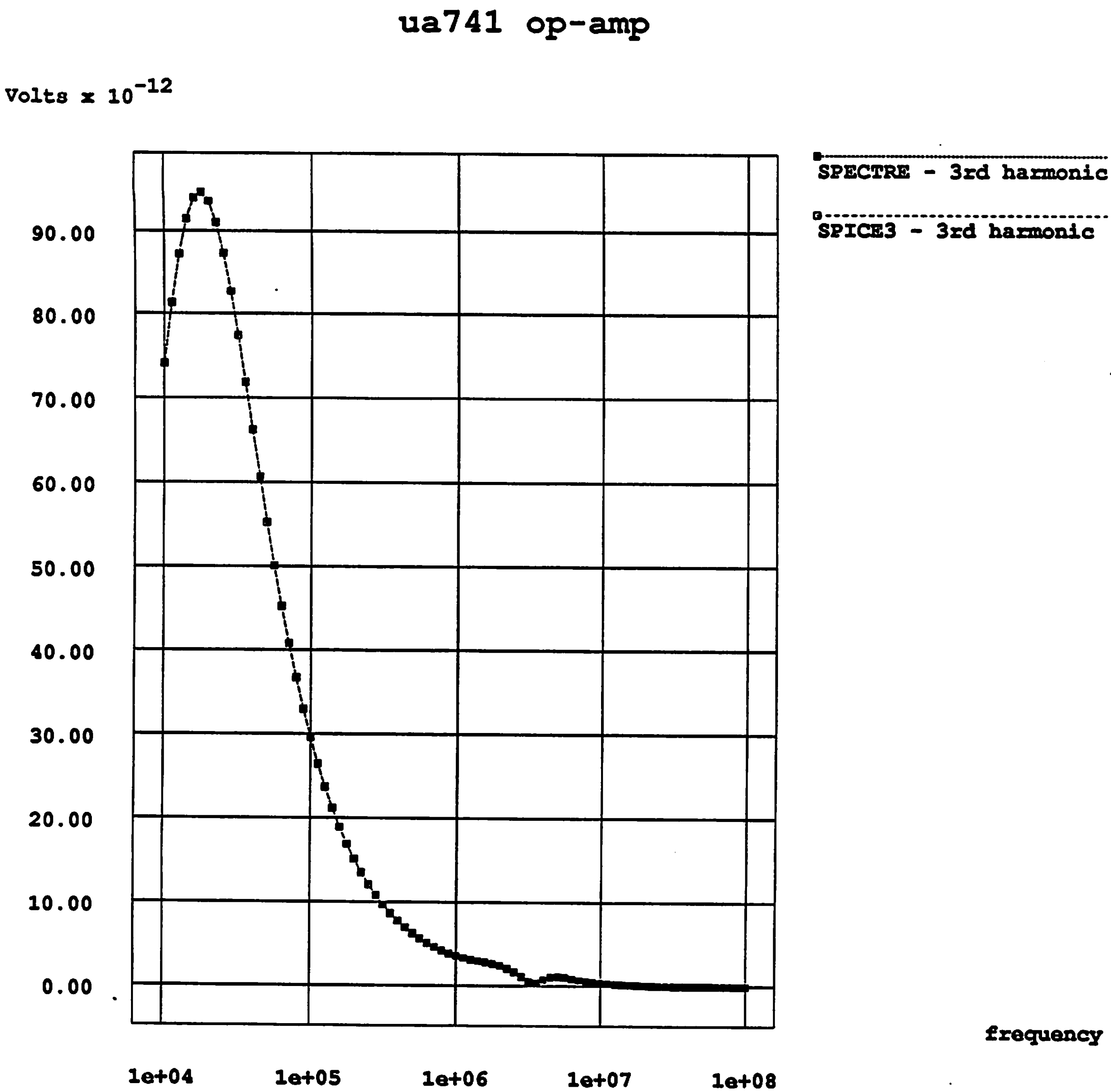

Figure D.34: ua741 op-amp - 3rd harmonic

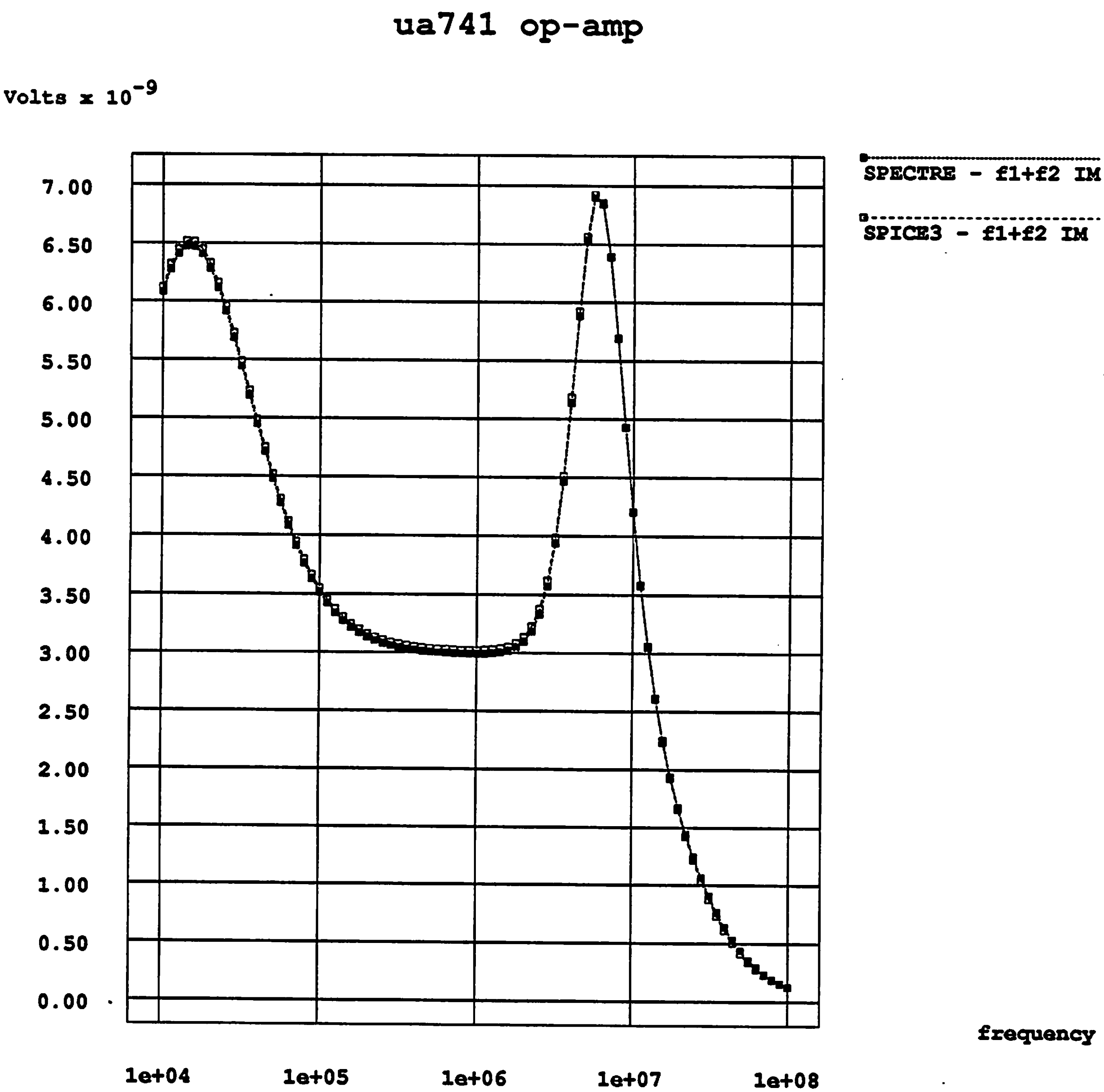

Figure D.35: ua741 op-amp - IM : $f_1 + f_2$

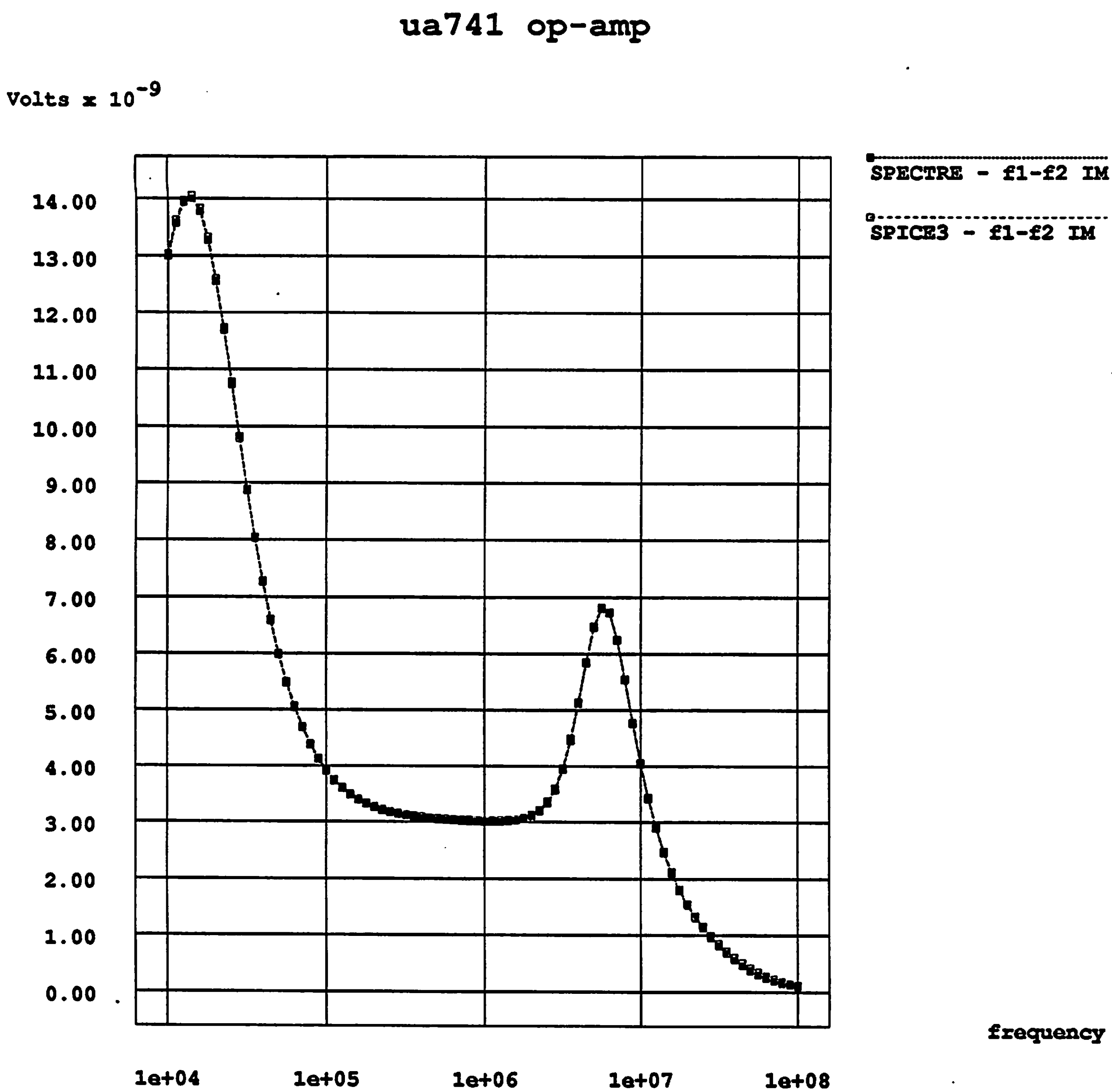

Figure D.36: ua741 op-amp - IM : $f_1 - f_2$

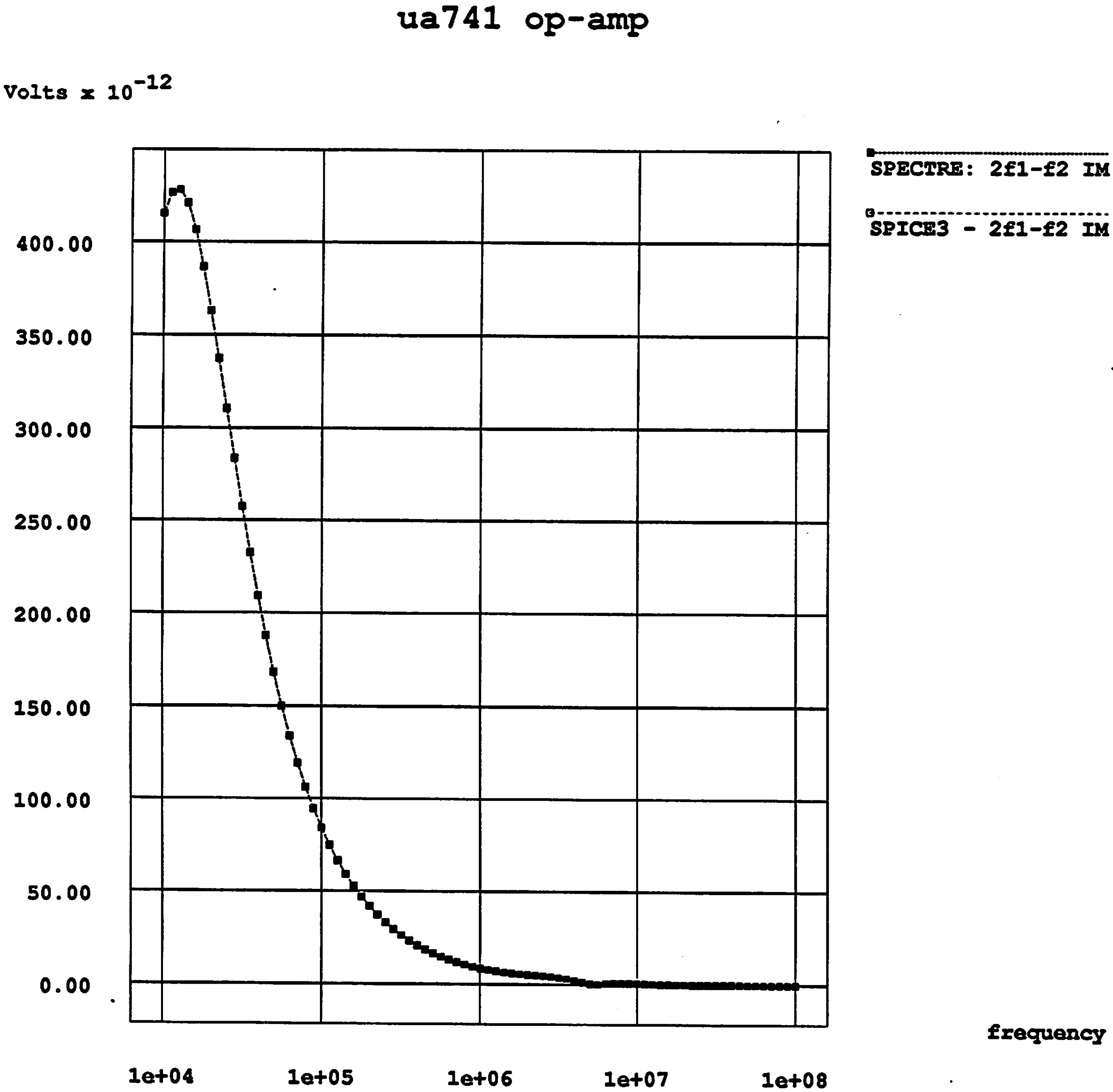

Figure D.37: ua741 op-amp - IM : $2f_1 - f_2$

D.8 ua733 wideband amplifier circuit

The circuit is ua733 wideband amplifier.

The data in Table 4.1 was obtained by running the given input files at 1kHz.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for the analyses running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	6.92	76.3	11
Spectral	17.71	2993.37	170

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: 170/2 = 85

```
UA733 CKT - UA 733 VIDEO PREAMPLIFIER

*.WIDTH IN=72
*.OPT ACCT
*.AC DEC 10 100 1GHZ
*.TRAN 2.5NS 250NS
VIN 30 0 0v sin(0 0.01 1000) distof1 0.01 distof2 0.01
*PWL(0 0 10NS 100M 90NS 100M 100NS -100M 190NS -100M 200NS 0) + AC 1
*.PLOT AC VM(19) VM(22) (.01,100)
*.PLOT TRAN V(19) V(22) V(30)
VCC 11 0 8
VEE 9 0 -8
Q1 3 1 4 9 Q1
Q2 14 2 13 9 Q1
Q3 17 14 16 9 Q1
Q4 18 3 16 9 Q1
Q5 11 18 19 9 Q1
Q6 11 17 22 9 Q1
Q7 6 7 8 9 Q1
Q8 7 7 10 9 Q1
Q9 16 7 15 9 Q1
Q10 19 7 20 9 Q1
Q11 22 7 21 9 Q1
R1 1 30 51
R2 2 0 51
R3 11 3 2.4K
R4 11 14 2.4K
R5 4 5 50
R6 13 12 50
R7 5 6 590
R8 12 6 590
R9 11 7 10K
R10 11 17 1.1K
R11 11 18 1.1K
R12 3 19 7K
R13 14 22 7K
R14 8 9 300
R15 10 9 1.4K
R16 15 9 300
R17 20 9 400
R18 21 9 400
.MODEL Q1 NPN(BF=100 BR=2 IS=0.9901E-15 RB=100 CJS=2PF TF=0.3NS TR=6NS  CJE=3PF CJC=2PF)
*.disto dec 1 1000 1000
*.disto dec 20 1.0e4 1.0e8
*.disto dec 20 1.0e4 1.0e8 0.9
* the output is v(22)
*.width out=70
.end
```

```
;UA733 CKT - UA 733 VIDEO PREAMPLIFIER
global 0

;models

model Q1 bjt npn=yes bf=100 br=2 is=0.9901e-15 rb=100 cjs=2e-12 tf=0.3e-9 tr=6e-9 cje=3e-12 cjc=2e-12

;circuit

VIN 30 0 vsource vdc=0 mag=0.01 mag1=0.01 mag1f=0.01 mag2f=0.01
VCC 11 0 vsource vdc=8
VEE 9 0 vsource vdc=-8
Q1 3 1 4 9 Q1
Q2 14 2 13 9 Q1
Q3 17 14 16 9 Q1
Q4 18 3 16 9 Q1
Q5 11 18 19 9 Q1
Q6 11 17 22 9 Q1
Q7 6 7 8 9 Q1
Q8 7 7 10 9 Q1
Q9 16 7 15 9 Q1
Q10 19 7 20 9 Q1
Q11 22 7 21 9 Q1
R1 1 30 resistor r=51
R2 2 0 resistor r=51
R3 11 3 resistor r=2.4K
R4 11 14 resistor r=2.4K
R5 4 5 resistor r=50
R6 13 12 resistor r=50
R7 5 6 resistor r=590
R8 12 6 resistor r=590
R9 11 7 resistor r=10K
R10 11 17 resistor r=1.1K
R11 11 18 resistor r=1.1K
R12 3 19 resistor r=7K
R13 14 22 resistor r=7K
R14 8 9 resistor r=300
R15 10 9 resistor r=1.4K
R16 15 9 resistor r=300
R17 20 9 resistor r=400
R18 21 9 resistor r=400

;analyses

;the output is v(22)
;boom harmonic maxharm=3 fund=1.0e3
boom harmonic maxharm=3 start=1.0e4 stop=1.0e8 dec=20
aagh spectral fund2=0.9e4 order=3 maxharm1=3 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e8 dec=20
```

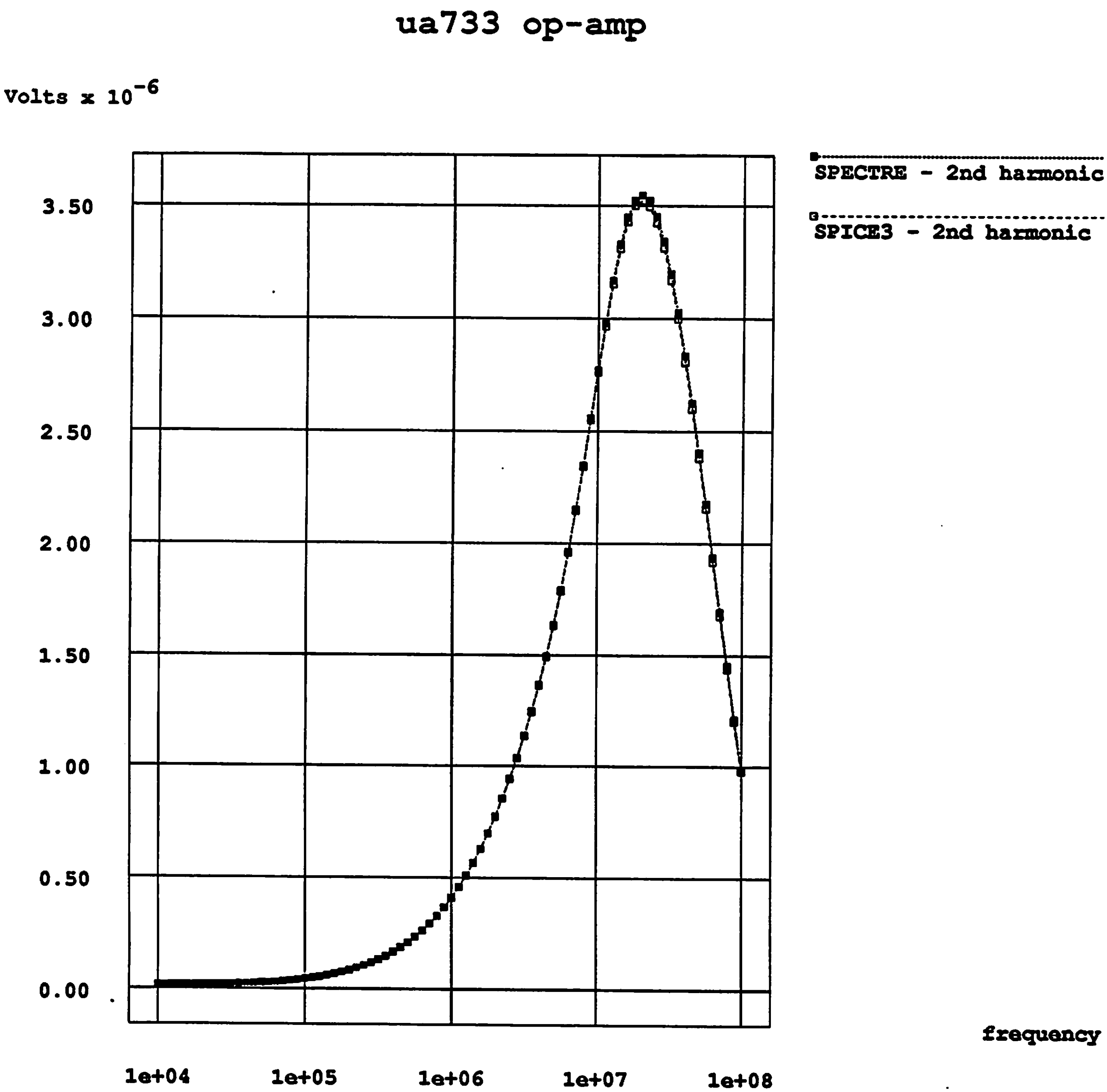

Figure D.38: ua733 wideband amplifier circuit - 2nd harmonic

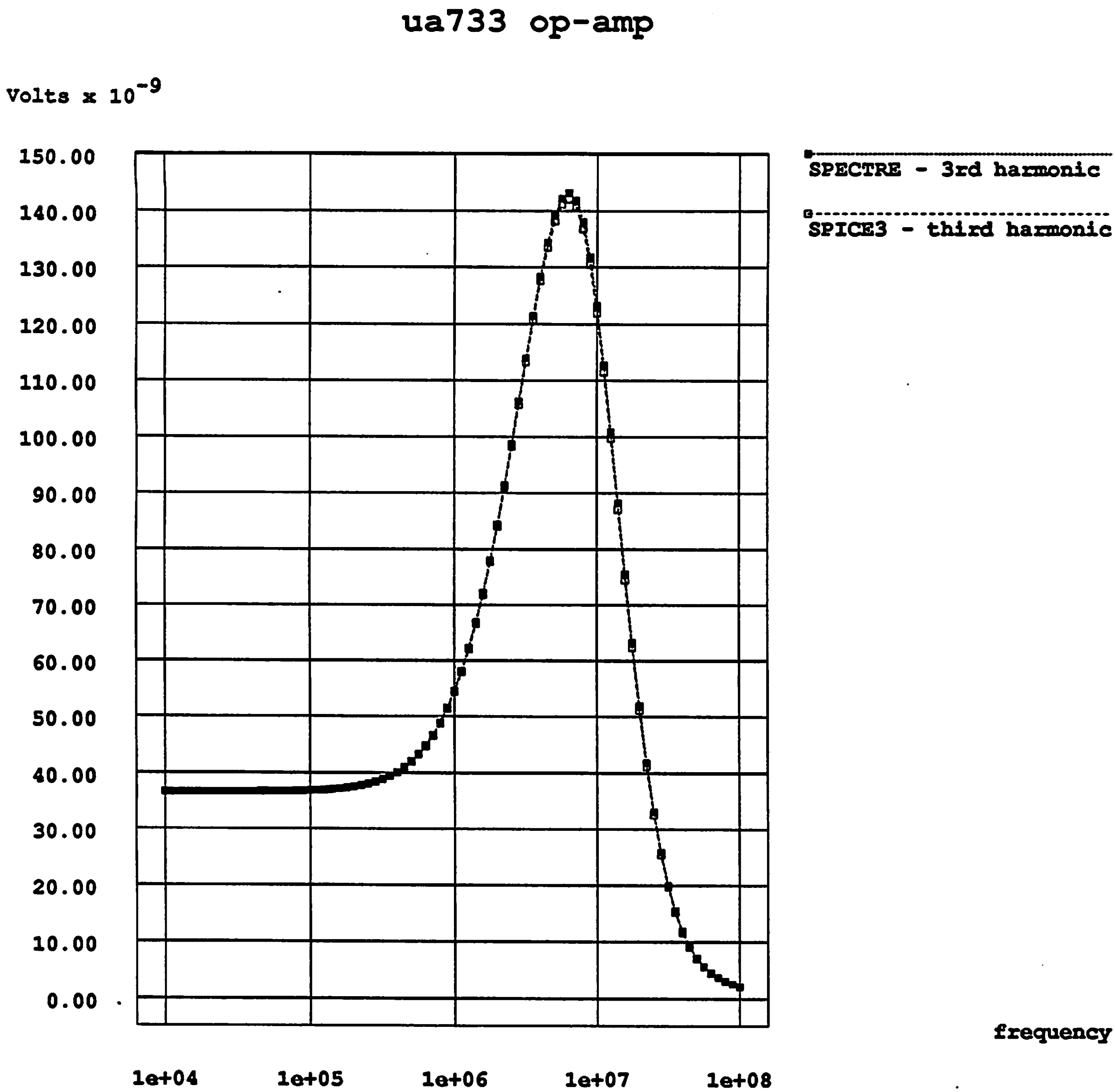

Figure D.39: ua733 wideband amplifier circuit - 3rd harmonic

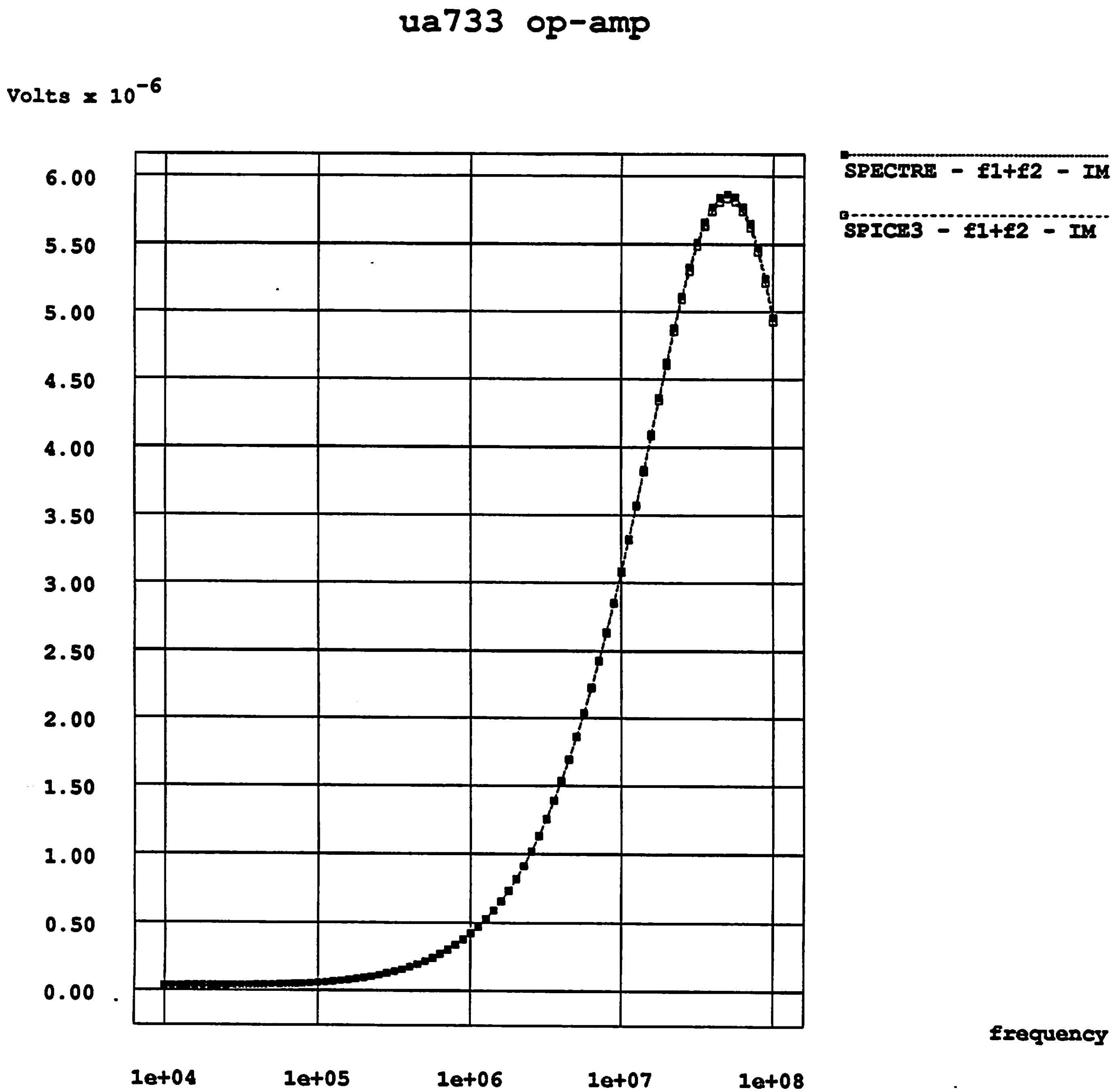

Figure D.40: ua733 wideband amplifier circuit - IM : $f_1 + f_2$

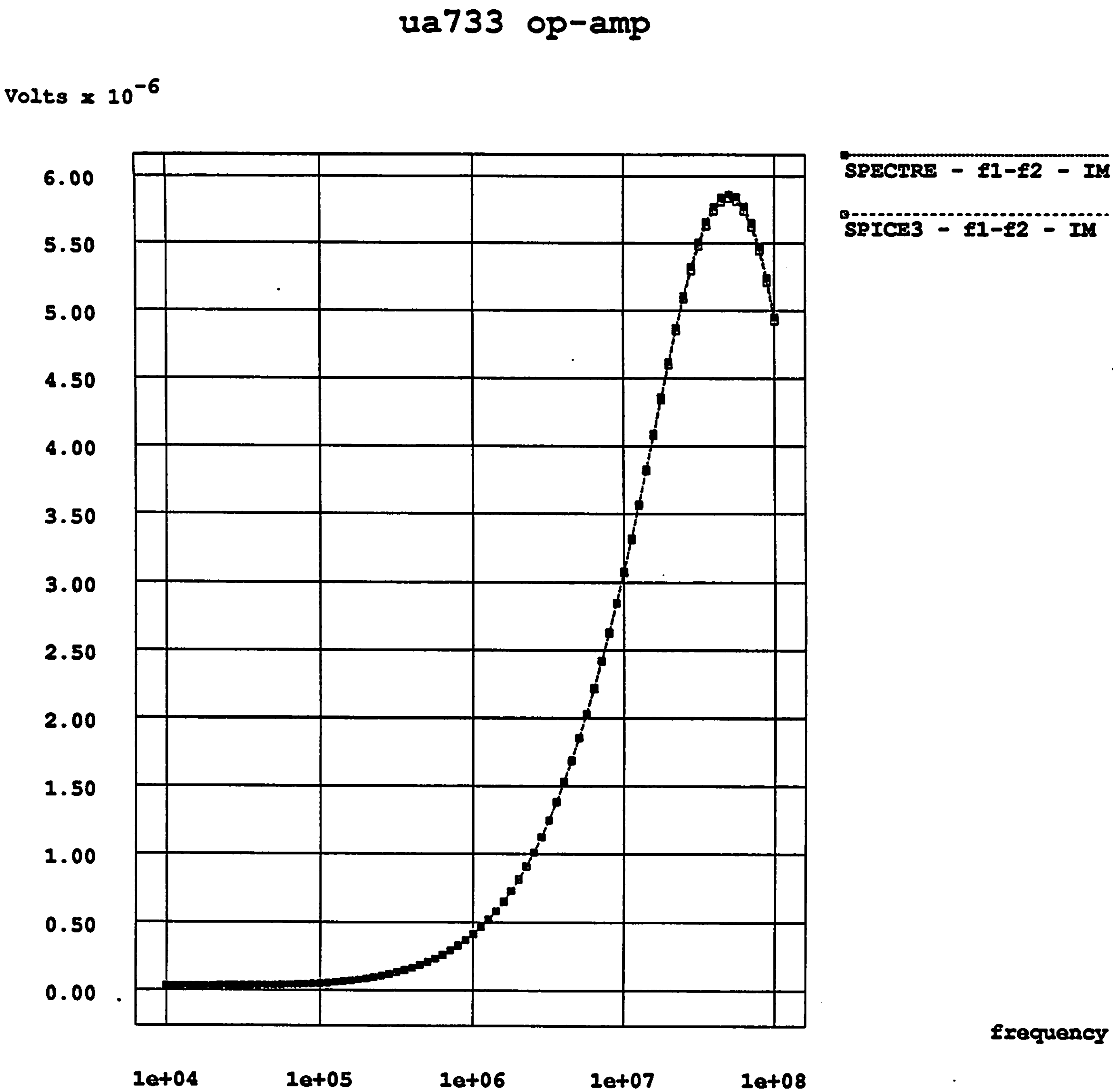

Figure D.41: ua733 wideband amplifier circuit - IM : $f_1 - f_2$

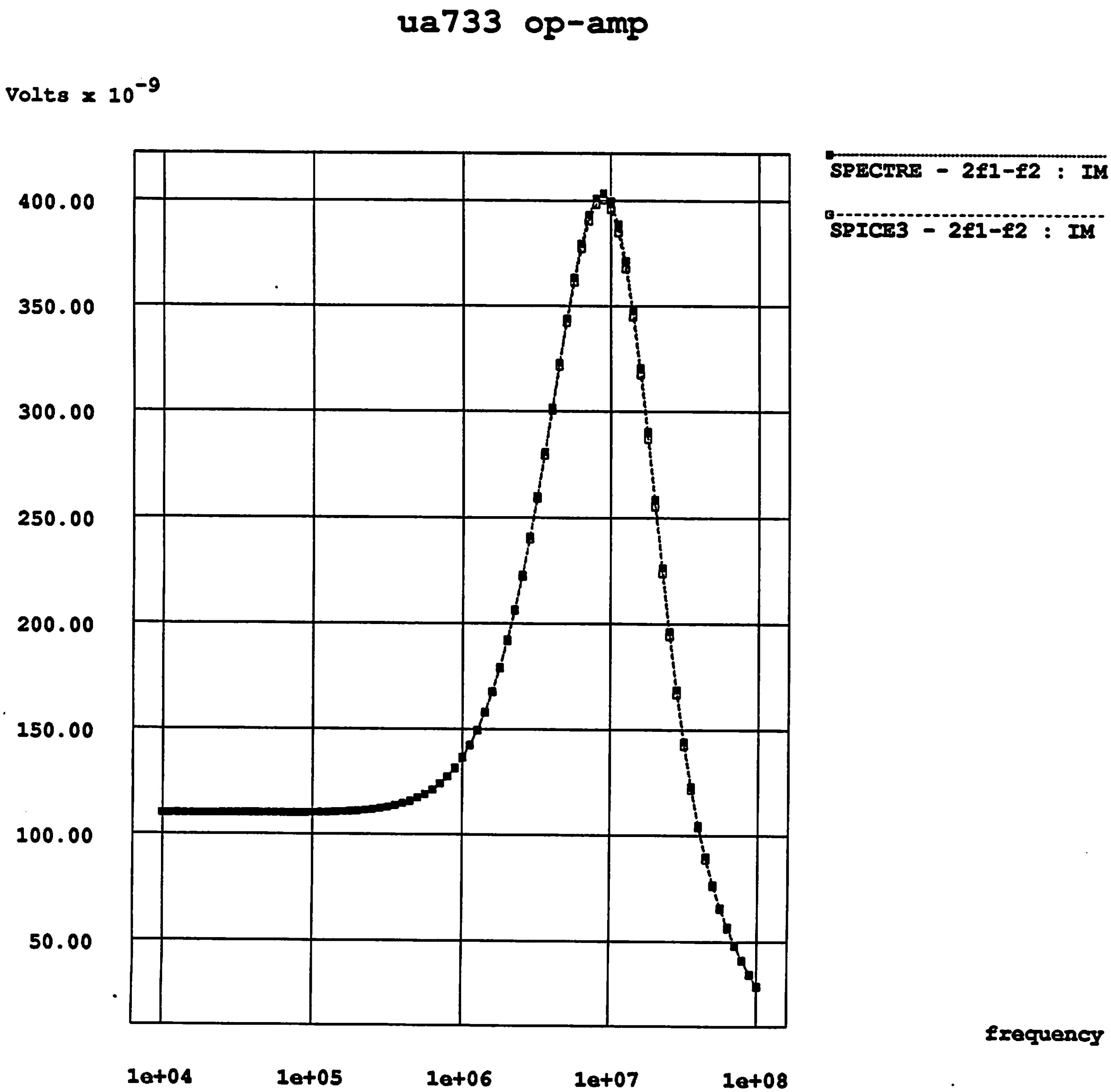

Figure D.42: ua733 wideband amplifier circuit - IM : $2f_1 - f_2$

D.9 MOSAMP2 mos amplifier

The circuit is a simple diode-resistor-voltage-source series circuit.

The data in Table 4.1 was obtained by running the given input file at 1kHz.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for a *.disto* analysis (using the level 2 model) running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	9.01	-	-
Spectral	18.48	-	-

```
mosamp2 - mos amplifier

*.options acct abstol=10n  vntol=10n
*.tran 0.1us 10us
m1  15 15  1 32 m w=88.9u  l=25.4u
m2   1  1  2 32 m w=12.7u  l=266.7u
m3   2  2 30 32 m w=88.9u  l=25.4u
m4  15  5  4 32 m w=12.7u  l=106.7u
m5   4  4 30 32 m w=88.9u  l=12.7u
m6  15 15  5 32 m w=44.5u  l=25.4u
m7   5 20  8 32 m w=482.6u l=12.7u
m8   8  2 30 32 m w=88.9u  l=25.4u
m9  15 15  6 32 m w=44.5u  l=25.4u
m10  6 21  8 32 m w=482.6u l=12.7u
m11 15  6  7 32 m w=12.7u  l=106.7u
m12  7  4 30 32 m w=88.9u  l=12.7u
m13 15 10  9 32 m w=139.7u l=12.7u
m14  9 11 30 32 m w=139.7u l=12.7u
m15 15 15 12 32 m w=12.7u  l=207.8u
m16 12 12 11 32 m w=54.1u  l=12.7u
m17 11 11 30 32 m w=54.1u  l=12.7u
m18 15 15 10 32 m w=12.7u  l=45.2u
m19 10 12 13 32 m w=270.5u l=12.7u
m20 13  7 30 32 m w=270.5u l=12.7u
m21 15 10 14 32 m w=254u   l=12.7u
m22 14 11 30 32 m w=241.3u l=12.7u
m23 15 20 16 32 m w=19u    l=38.1u
m24 16 14 30 32 m w=406.4u l=12.7u
m25 15 15 20 32 m w=38.1u  l=42.7u
m26 20 16 30 32 m w=381u   l=25.4u
m27 20 15 66 32 m w=22.9u  l=7.6u
cc 7 9 40pf
cl 66 0 70pf
vin 21 0 0v ac 0.0001 sin(0 0.0001 1000) distof1 0.0001 distof2 0.0001
*pulse(0 5 1ns 1ns 1ns 5us 10us)
vccp 15 0 dc +15
vddn 30 0 dc -15
vb 32 0 dc -20
.model m nmos
+  level=2 cgso=1.5n cgdo=1.5n cbd=4.5f cbs=4.5f ld=2.4485u nss=3.2e10
+  kp=2e-5 phi=0.6 uexp=0.1 xj=2.95u)
*.disto dec 1 1000 1000
*.disto dec 20 1.0e4 1.0e8
*.disto dec 20 1.0e4 1.0e8 0.9
* the output is v(20)

*.print tran v(20) v(66)
*.plot  tran v(20) v(66)
*.width out=70

.end
```

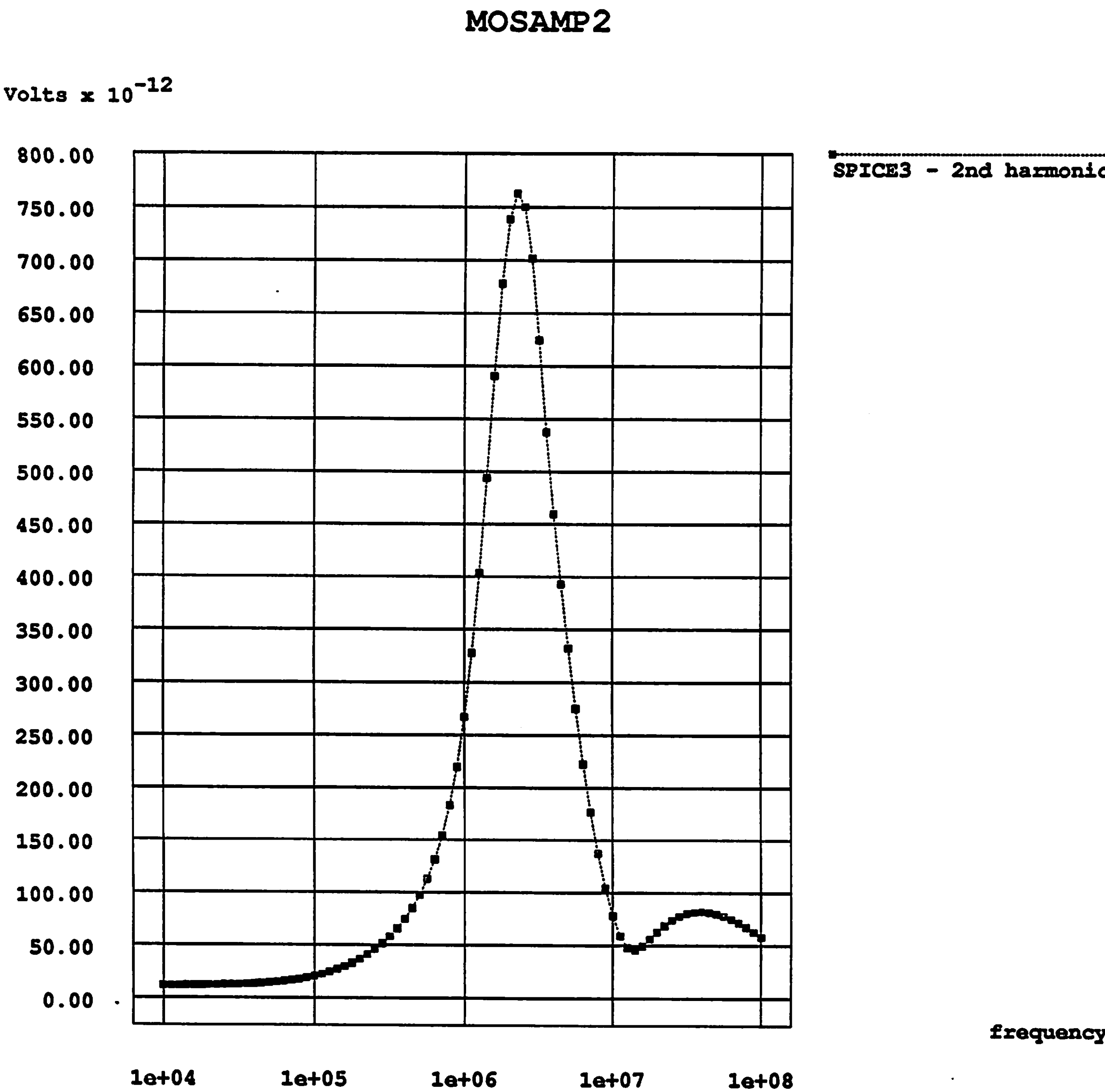

Figure D.43: MOSAMP2 mos amplifier - 2nd harmonic

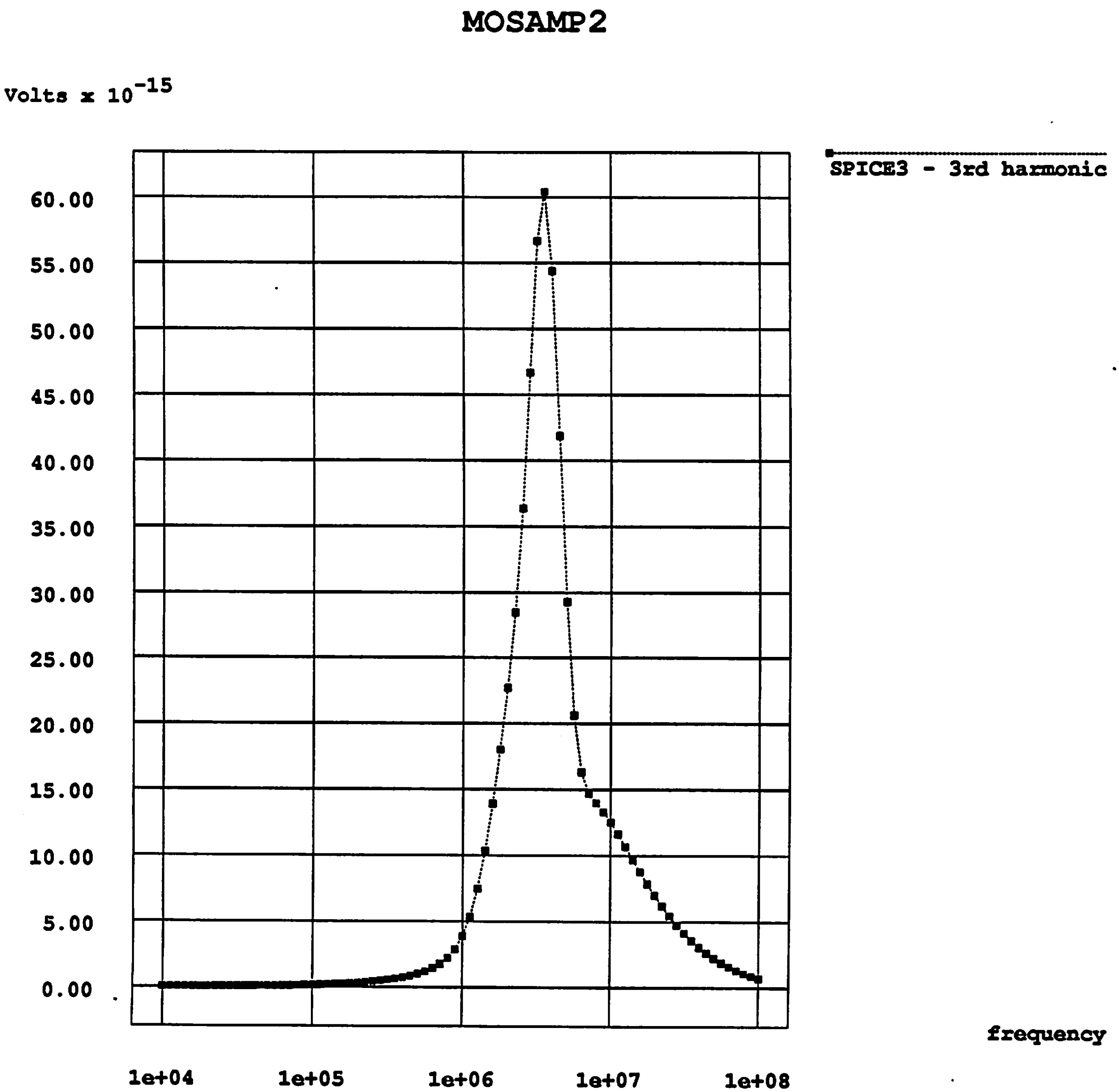

Figure D.44: MOSAMP2 mos amplifier - 3rd harmonic

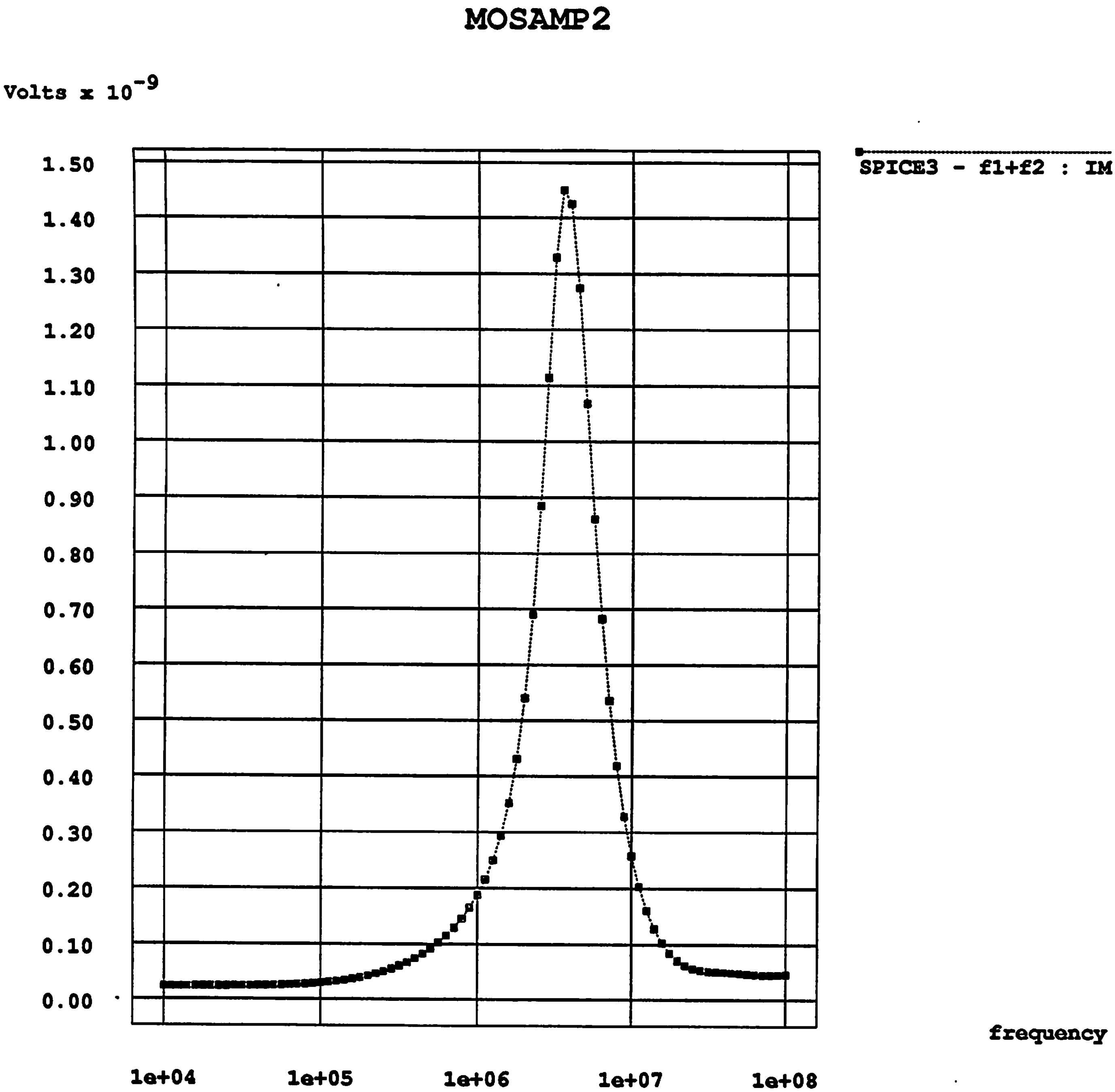

Figure D.45: MOSAMP2 mos amplifier - IM : $f_1 + f_2$

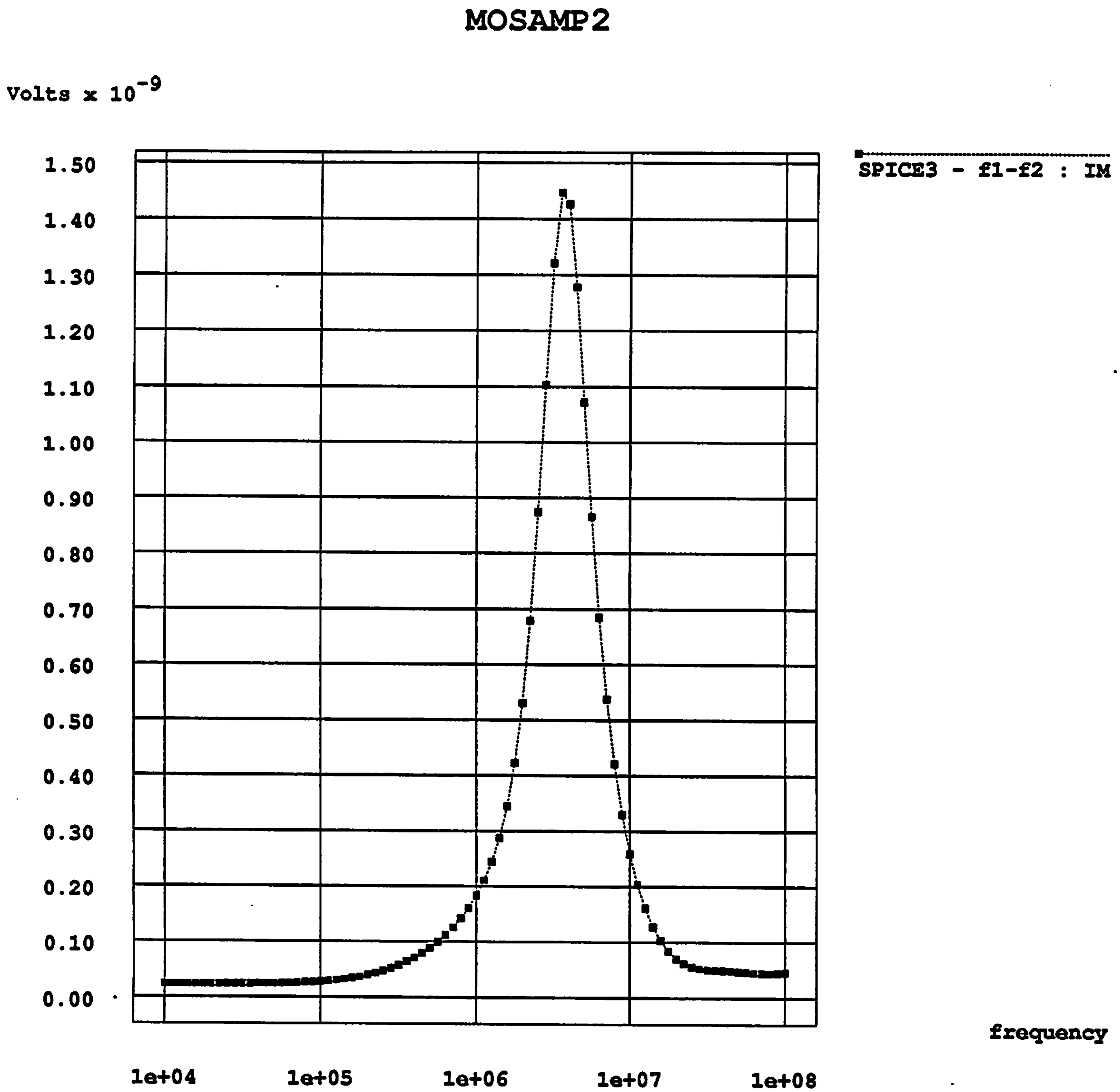

Figure D.46: MOSAMP2 mos amplifier - IM : $f_1 - f_2$

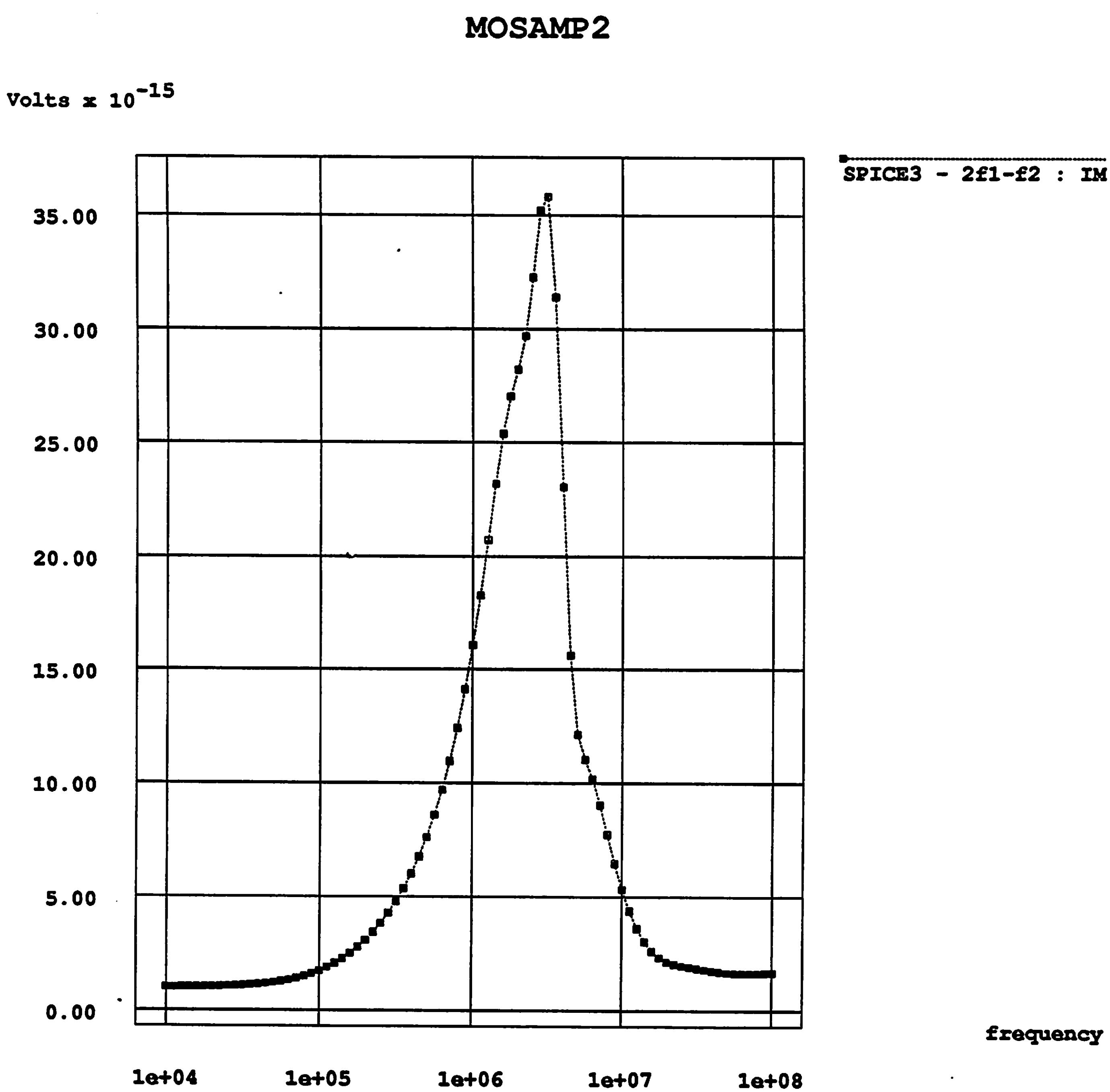

Figure D.47: MOSAMP2 mos amplifier - IM : $2f_1 - f_2$

D.10 Iee modulated Mixer

The circuit is a mixer circuit consisting of a differential pair whose Iee is modulated by an emitter-degenerated stage.

The data in Table 4.1 was obtained by running the given input file at 1kHz.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for a *.disto* analysis (using the level 2 model) running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	0.67	5.77	8
Spectral	1.45	155.98	107

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: 107/2 = 53

simplemixer.spice3 Wed Mar 15 02:00:31 1989 1

```
Simple mixer circuit - diffpair fed by modulated Iee

*the output is v(6)

* harmonic analysis done with the following:
*v1 1 0  0v ac 1.0 distof1 0.001
*v2 7 0 0v ac 1.0 distof1 0.001
*disto dec 20 1.0e3 1.0e8

* intermodulation done with the following:
v1 1 0  0v ac 1.0 distof1 0.001
v2 7 0 0v ac 1.0 distof2 0.001
*disto dec 20 1.0e3 1.0e8 0.9

q1 3 1 4 2 mod1
q2 6 0 4 2 mod1
rc1 5 3 10k
rc2 5 6 10k
vcc 5 0 10v
qee 4 7 8 2 mod1
re 8 2 9K
vee 2 0 -10v

* note - spectre does not implement rb nonlinearities

*.model mod1 npn is=1.0e-16 bf=100 this model matches with spectre
.model mod1 npn is=1e-16 bf=100 cjs=2.0e-12 tf=0.3e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50
*the above model matches with spectre
*.model mod1 npn is=1e-16 bf=100 cjs=2.0e-12 tr=6.0e-9 tf=0.3e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50 this agrees with spectre
*.model mod1 npn is=1e-16 bf=100 rb=100 cjs=2.0e-12 tf=0.3e-9 tr=6e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50 this model causes bad match with spectre
.end
```

simplemixer.spectre Wed Mar 15 01:46:14 1989 1

```
; Simple mixer - diffpair fed by modulated Iee

; Circuit
global 0

v1 1 0 vsource vdc=0 mag=0.0001 phase=0.0 mag1=0.001 mag1f=0.001
v2 7 0 vsource vdc=0 mag=0.0001 phase=0.0 mag1=0.001 mag2f=0.001
q1 3 1 4 2 mod1 region=1
q2 6 0 4 2 mod1 region=1
q3 4 7 8 2 mod1 region=1
rc1 5 3 resistor r=10k
rc2 5 6 resistor r=10k
re  8 2 resistor r=9k
vcc 5 0 vsource vdc=10
vee 2 0 vsource vdc=-10

;models

;model mod1 bjt npn=yes is=1e-16 bf=100 this matches .disto
model mod1 bjt npn=yes is=1e-16 bf=100 cjs=2.0e-12 tf=0.3e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50 ;this matches .disto
;model mod1 bjt npn=yes is=1e-16 bf=100 cjs=2.0e-12 tf=0.3e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50 tr=6.0e-9 this matches .disto
;model mod1 bjt npn=yes is=1e-16 bf=100 rb=100 cjs=2.0e-12 tf=0.3e-9 tr=6e-9 cje=3.0e-12 cjc=2.0e-12  vaf=50 this does not match .disto

; Analyses

;output is v(6)
boom harmonic maxharm=3 start=1.0e4 stop=1.0e8 dec=10 ;fund=1.0e3
aagh spectral order=3 maxharm1=3 maxharm2=3 param="fund1" start=1.0e4 stop=1.0e8 dec=10 fund2=0.9e4;fund1=1.0e6 fund2=0.9e6
```

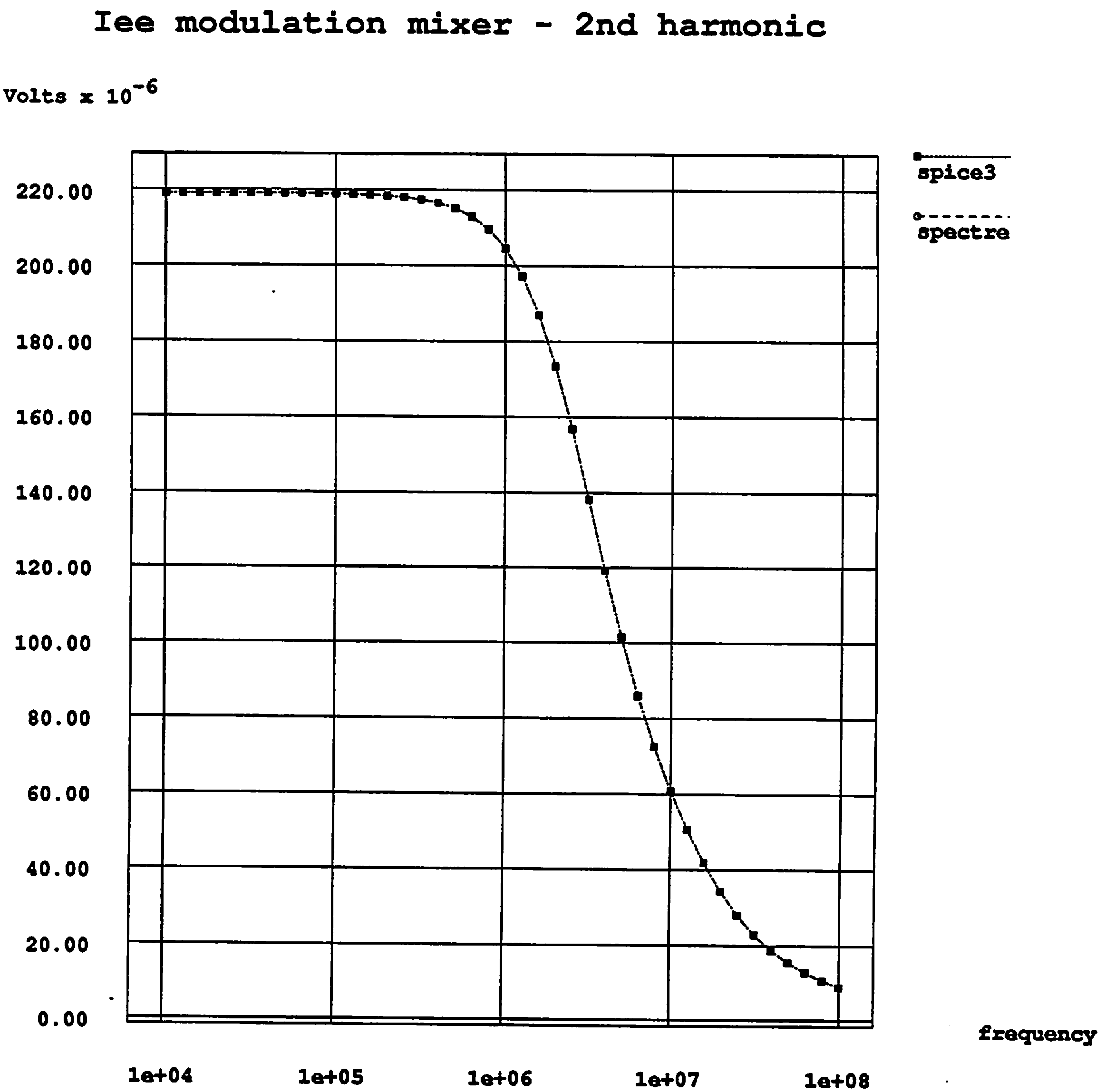

Figure D.48: Iee modulated Mixer - 2nd harmonic

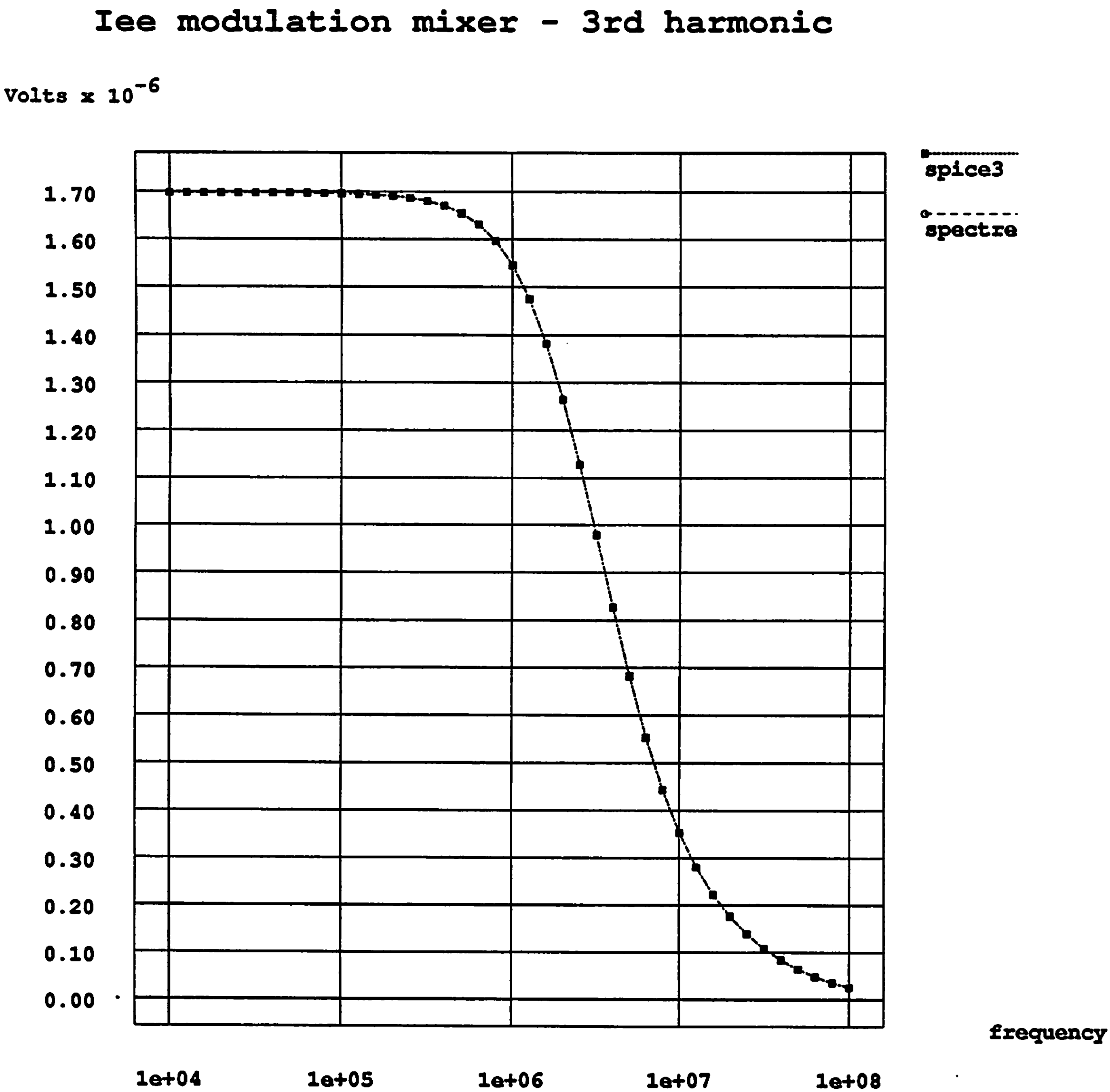

Figure D.49: Iee modulated Mixer - 3rd harmonic

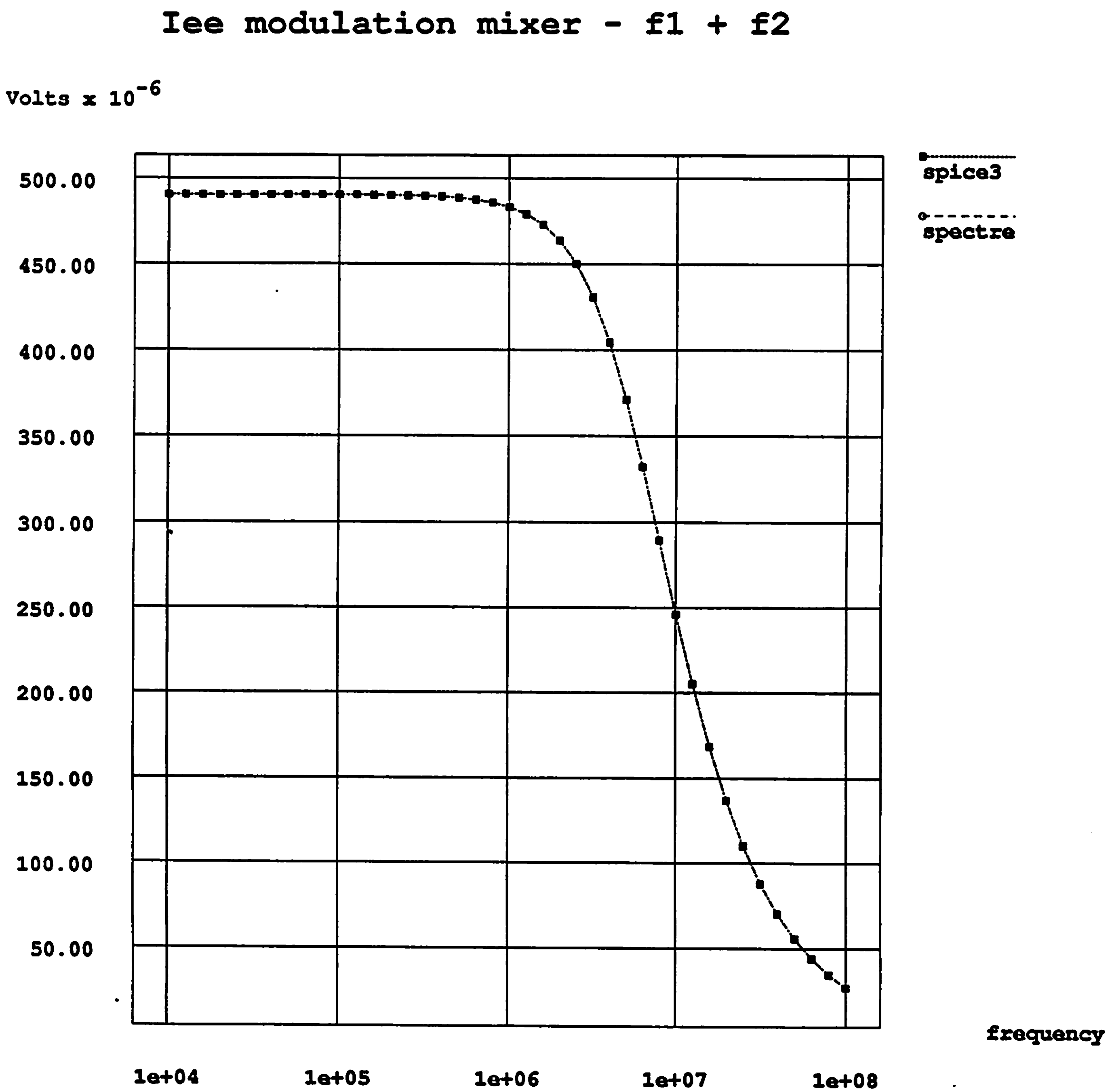

Figure D.50: Iee modulated Mixer - IM : $f_1 + f_2$

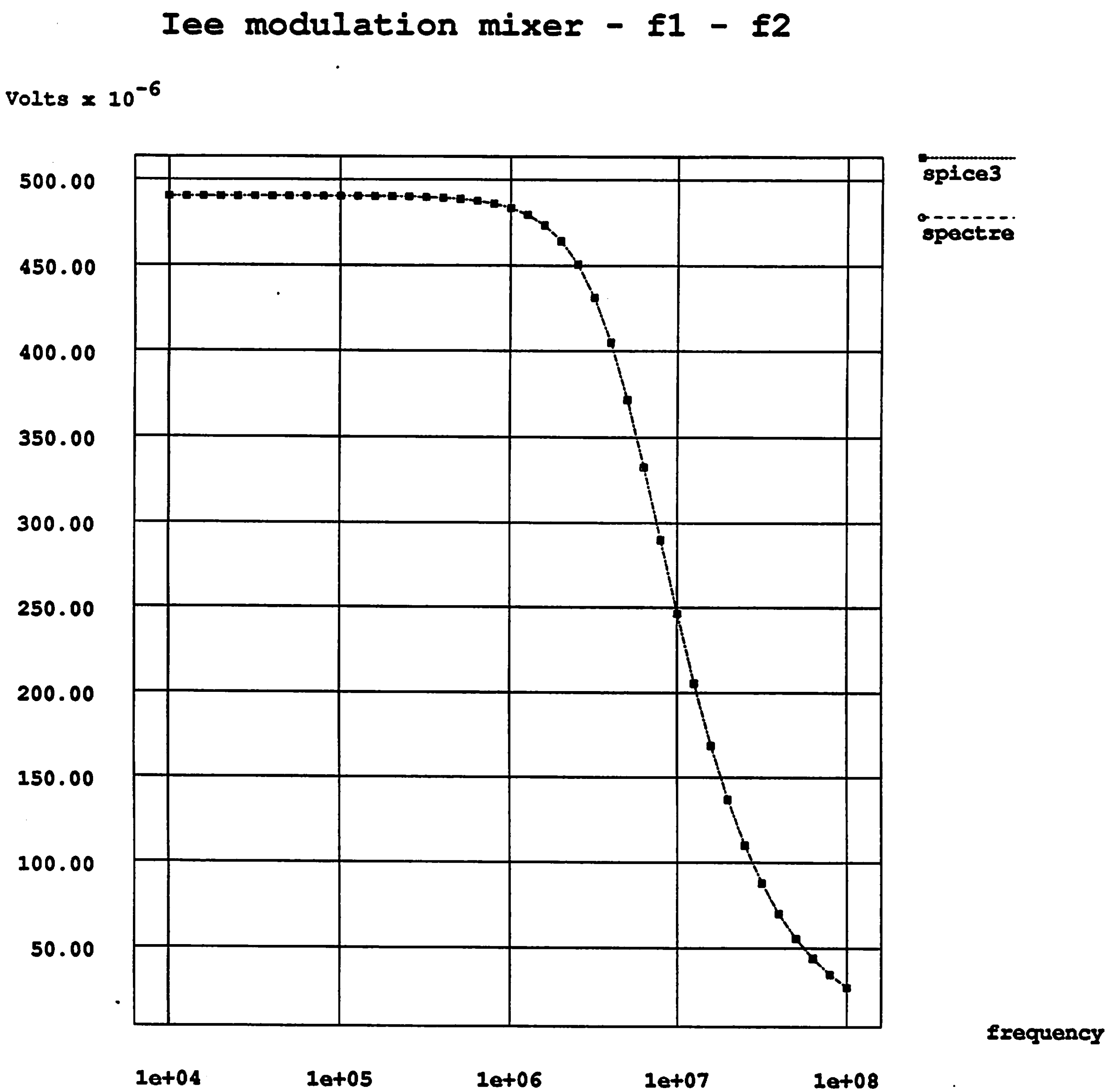

Figure D.51: Iee modulated Mixer - IM : $f_1 - f_2$

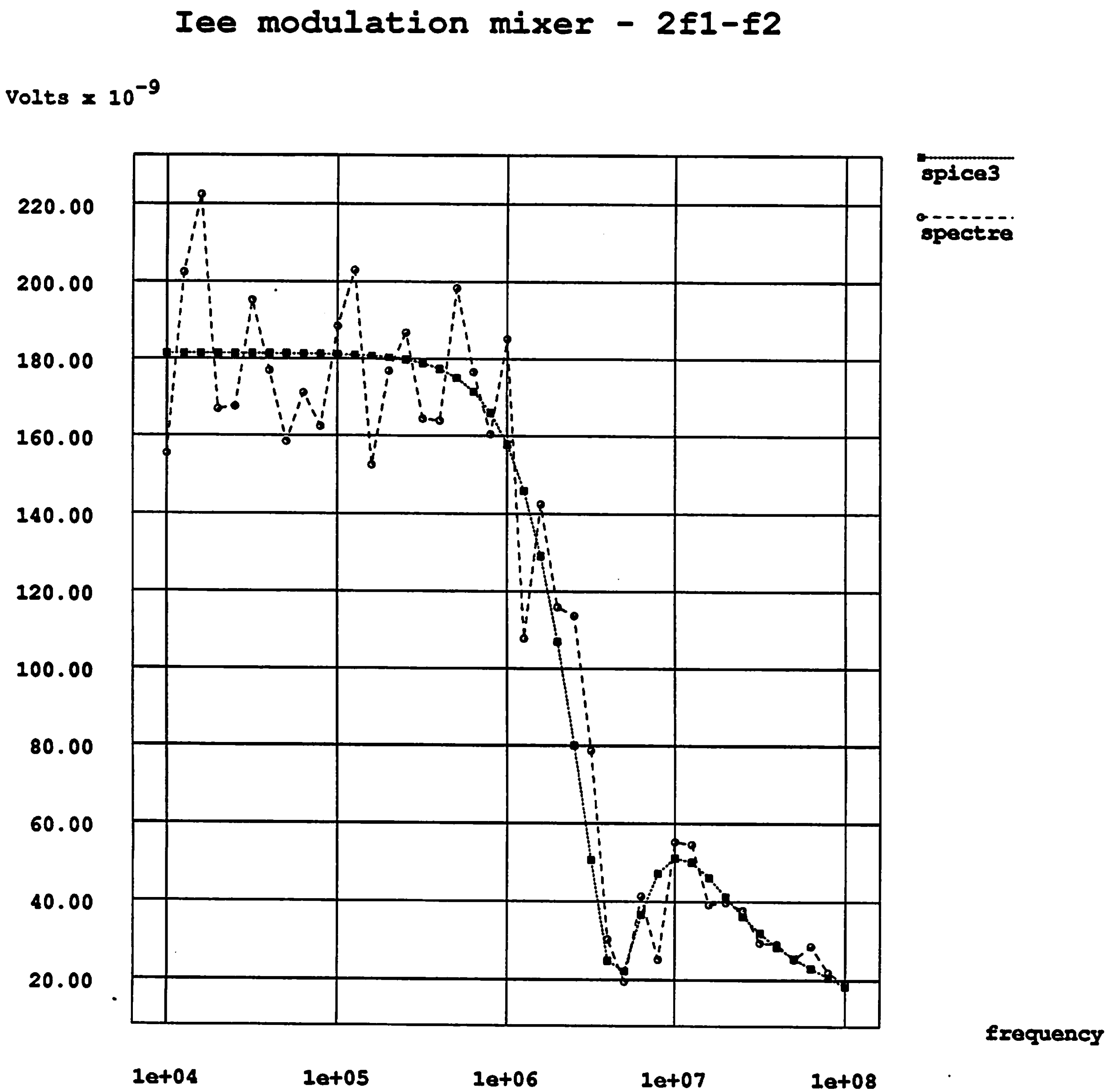

Figure D.52: Iee modulated Mixer - IM : $2f_1 - f_2$

D.11 Double Balanced Mixer

The circuit is a double balanced mixer, often used in Gilbert cells. One of the features of this circuit is that odd harmonics and powers in the Taylor series cancel in the output.

The data in Table 4.1 was obtained by running the given input file at 1kHz.

The fundamental frequency is swept from 10kHz to 100Mhz. In the case of the spectral (intermodulation) analysis, the second frequency is 9kHz kept constant while sweeping the primary frequency. The CPU times for a *.disto* analysis (using the level 2 model) running on ic (a VAX 8650 running Ultrix 3.0) are as follows:

Analysis	SPICE3 *.disto* time (s)	SPECTRE time (s)	Speedup
Harmonic	1.59	17.73	11
Spectral	2.8	385.1	138

In the harmonic analysis case, SPECTRE produces the same amount of data as SPICE3 for the given times. For the spectral analysis, SPECTRE produces 12 intermodulation plots while SPICE3 does only 6. A more reasonable speedup factor in the spectral case is therefore: 138/2 = 69

```
double balanced mixer - from jean

*input
*vin1 1 2 0 ac .01 sin(0 10m 1.1meg) distof1 .01
*vin2 8 9 0 sin(0 1m 1meg) distof2 .001
vin1 1 0 0 distof1 0.005
vin2 0 2 0 distof1 0.005
vin3 8 0 0 distof2 0.0005
vin4 0 9 0 distof2 0.0005
*vdum1 2 0 0
*vdum2 8 0 0
*vin2 8 9 0 sin 0 1m 1meg
*vin1 1 2 0 sin 0 10m 1.1meg
*vin1 1 2 0
*vt1 1 8 0
*vt2 2 9 0

*resistors to help compute dc
*r1 1 0 10.0e6
*r2 2 0 10.0e6
*r3 8 0 10.0e6
*r4 9 0 10.0e6

*load
rc1 5 6 5k
rc2 5 7 5k

*two top ECP
q1 6 1 15 14 na214sw
q2 7 2 16 14 na214sw
q3 6 2 17 14 na214sw
q4 7 1 18 14 na214sw

*bottom ecp
q5 3 8 10 14 na214sw
q6 4 9 11 14 na214sw

*emitter degenration, if needed
re1 15 3 1
re2 16 3 1
re3 17 4 1
re4 18 4 1
re5 10 12 1k
re6 11 12 1k

*voltage and current sources
iee1 12 14 1m
vcc 5 0 5
vee 14 0 -5
*****************************************************************
*.model na214sw npn is=.051fa bf=160 nf=1.000 vaf=25.0v ikf=15.3ma
*+ise=0.00aa ne=1.50 br=62.0 nr=1.000 var=3.0v isc=0.00aa
*+nc=1.5 irb=28.8u rbm=94.0 re=1.60 rc=36.0 xtb=1.5
*+eg=1.205 xti=1.0 cje=58.0ff vje=.85v mje=0.30 tf=10.0ps xtf=0.00
*+itf=0.00ma ptf=0.0 cjc=63.0ff vjc=0.74v mjc=0.40 xcjc=0.41
*+cjs=83.0ff vjs=0.60v mjs=0.40 fc=0.875
*+rb=576 tr=200p

.model na214sw npn is=0.051e-15 bf=160 cjs=83.0ff vjs=0.60v mjs=0.40 fc=0.875 tf=10.0ps cje=58.0ff vje=.85v mje=0.30
*****************************************************************
```

```
;double balanced mixer
;simulation file to be used with spectre

global gnd

;options
Spectre options ascii=no resourceusage=yes digits=8

;sources
;vin2 8 9 vsource vdc=0 mag1f=1.0e-3
;vin1 1 2 vsource vdc=0 mag2f=10.0e-3
vin1 1 gnd vsource vdc=0 mag1=5.0e-3 mag1f=5.0e-3
vin2 gnd 2 vsource vdc=0 mag1=5.0e-3 mag1f=5.0e-3
vin3 8 gnd vsource vdc=0 mag2f=0.5e-3
vin4 gnd 9 vsource vdc=0 mag2f=0.5e-3
;vdum1 1 gnd vsource vdc=0
;vdum2 8 gnd vsource vdc=0
;vin 1 2 vsource vdc=0
vee 14 gnd vsource vdc=-5
vcc 5 gnd vsource vdc=5
iee1 12 14 isource idc=1m
;Vt2 2 9 vsource vdc=0
;Vt1 1 8 vsource vdc=0

;r1 1 gnd resistor r=10meg
;r2 2 gnd resistor r=10meg
;r3 8 gnd resistor r=10meg
;r4 9 gnd resistor r=10meg
rc1 5 6 resistor r=5k
rc2 5 7 resistor r=5k
re1 15 3 resistor r=1
re2 16 3 resistor r=1
re3 17 4 resistor r=1
re4 18 4 resistor r=1
re5 10 12 resistor r=1k
re6 11 12 resistor r=1k

;devices
Q1 6 1 15 14 na214sw
Q2 7 2 16 14 na214sw
Q3 6 2 17 14 na214sw
Q4 7 1 18 14 na214sw
Q5 3 8 10 14 na214sw
Q6 4 9 11 14 na214sw

;models
;model na214sw bjt npn=yes is=.051fa bf=160 nf=1.000 vaf=25.0 ikf=15.3ma\
;         ise=0.00aa ne=1.50 br=62.0 nr=1.000 var=3.0 isc=0.00aa\
;         nc=1.5 irb=28.8u rbm=94.0 re=1.60 rc=36.0 xtb=1.5\
;         eg=1.205 xti=1.0 cje=58.0ff vje=.85 mje=0.30 tf=10.0ps xtf=0.00\
;         itf=0.00ma ptf=0.0 cjc=63.0ff vjc=0.74 mjc=0.40 xcjc=0.41\
;         cjs=83.0ff vjs=0.60 mjs=0.40 fc=0.875\
;         tr=200p rb=576
model na214sw bjt npn=yes is=0.051e-15 bf=160 \
cjs=83.0e-15 vjs=0.60 mjs=0.40 fc=0.875 tf=10.0e-12 cje=58.0e-15 vje=.85 mje=0.30
;analysis
opt options vreltol=1.0e-10 ireltol=1.0e-10 vabstol=1.0e-10 iabstol=1.0e-10
;op dc
;dc_op dc sweep=vin param="vdc" start=-500e-3 stop=500e-3 step=50e-3
Harmonic harmonic start=1.0e4 stop=1.0e8 dec=10 maxharm=3
;Spectrum spectral  param="fund1" fund2=0.9e4 start=1.0e4 stop=1.0e8 \
;dec=10 maxharm1=3 maxharm2=3 order=3
```

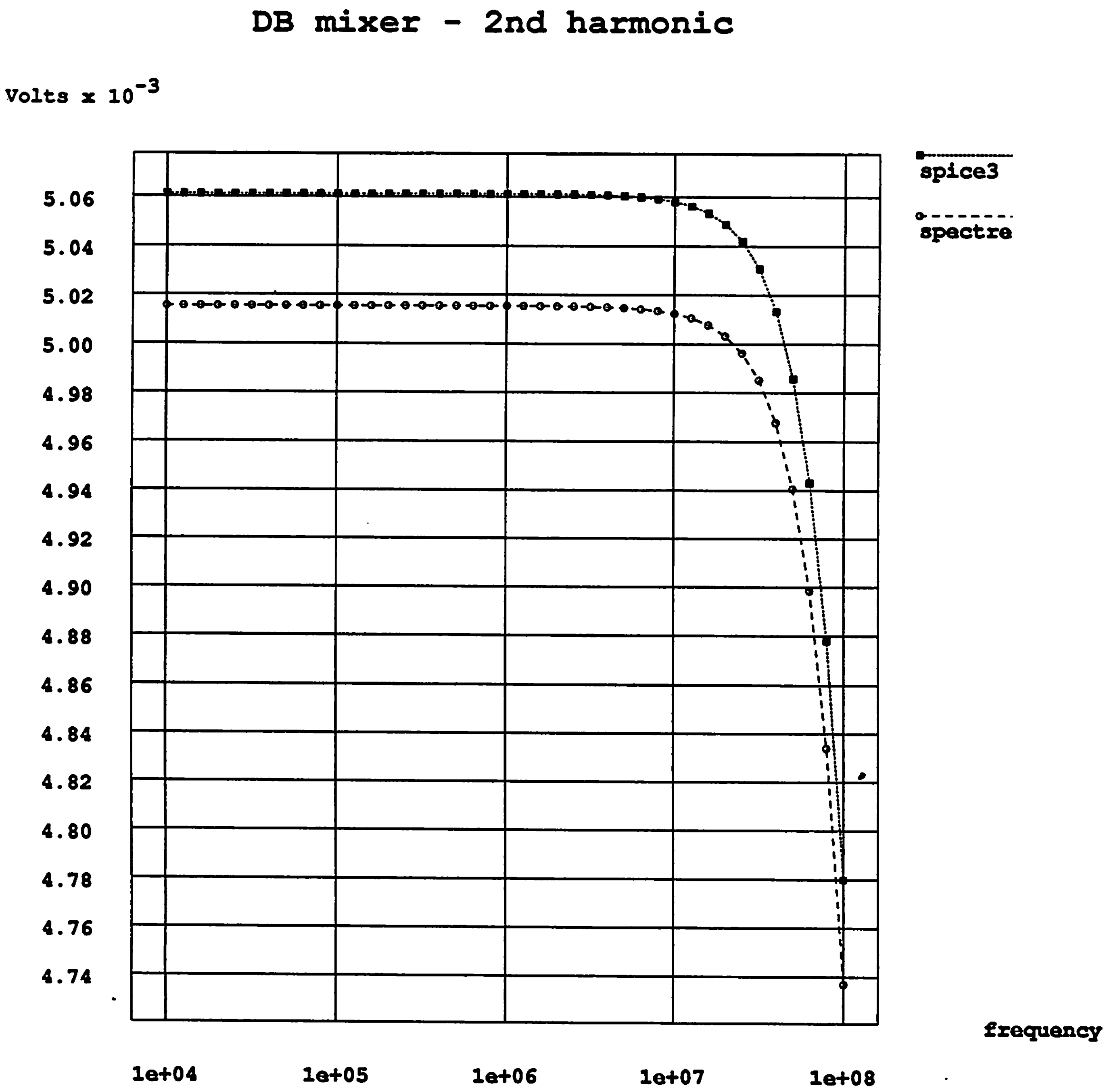

Figure D.53: Double Balanced Mixer - 2nd harmonic

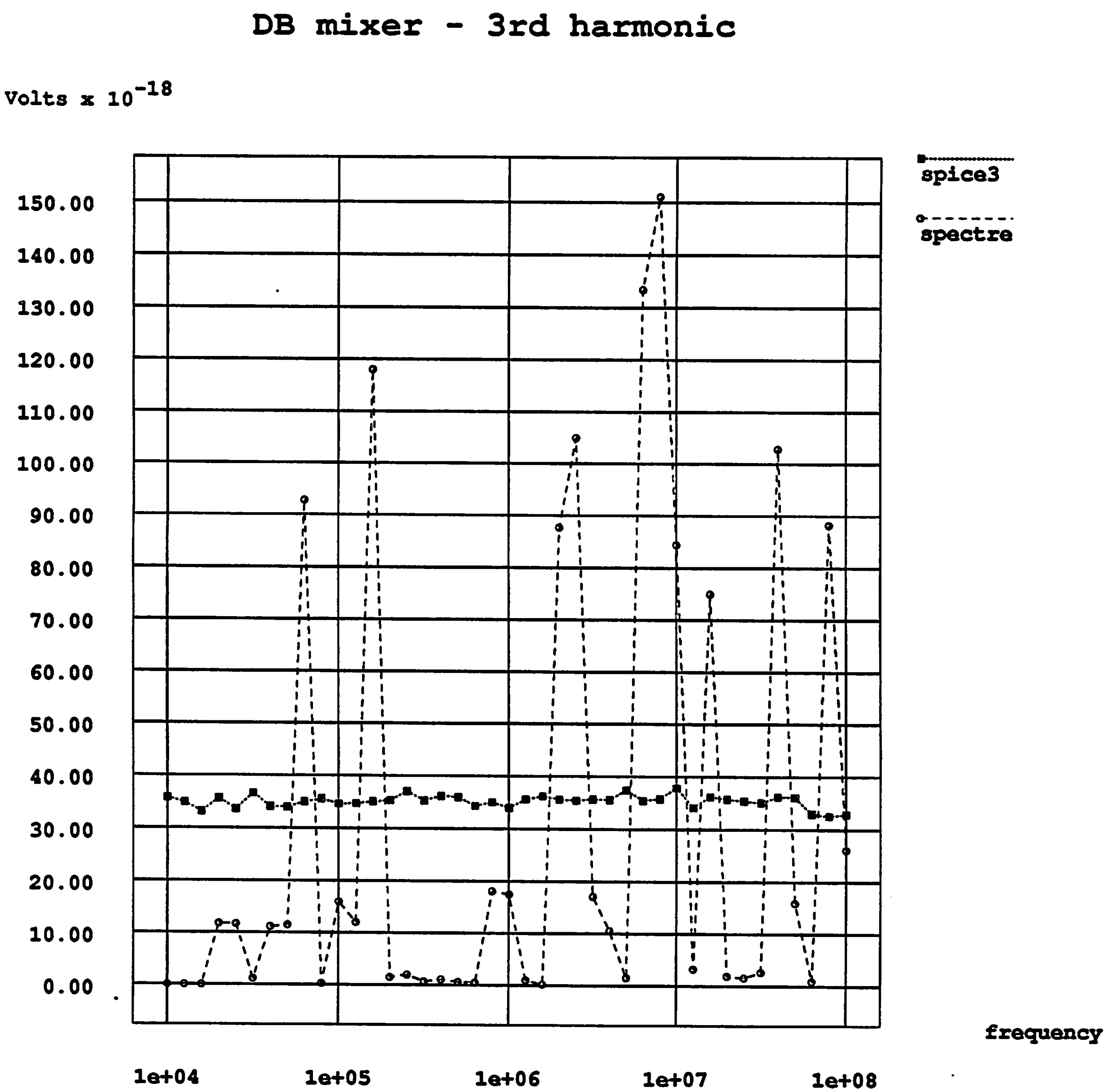

Figure D.54: Double Balanced Mixer - 3rd harmonic

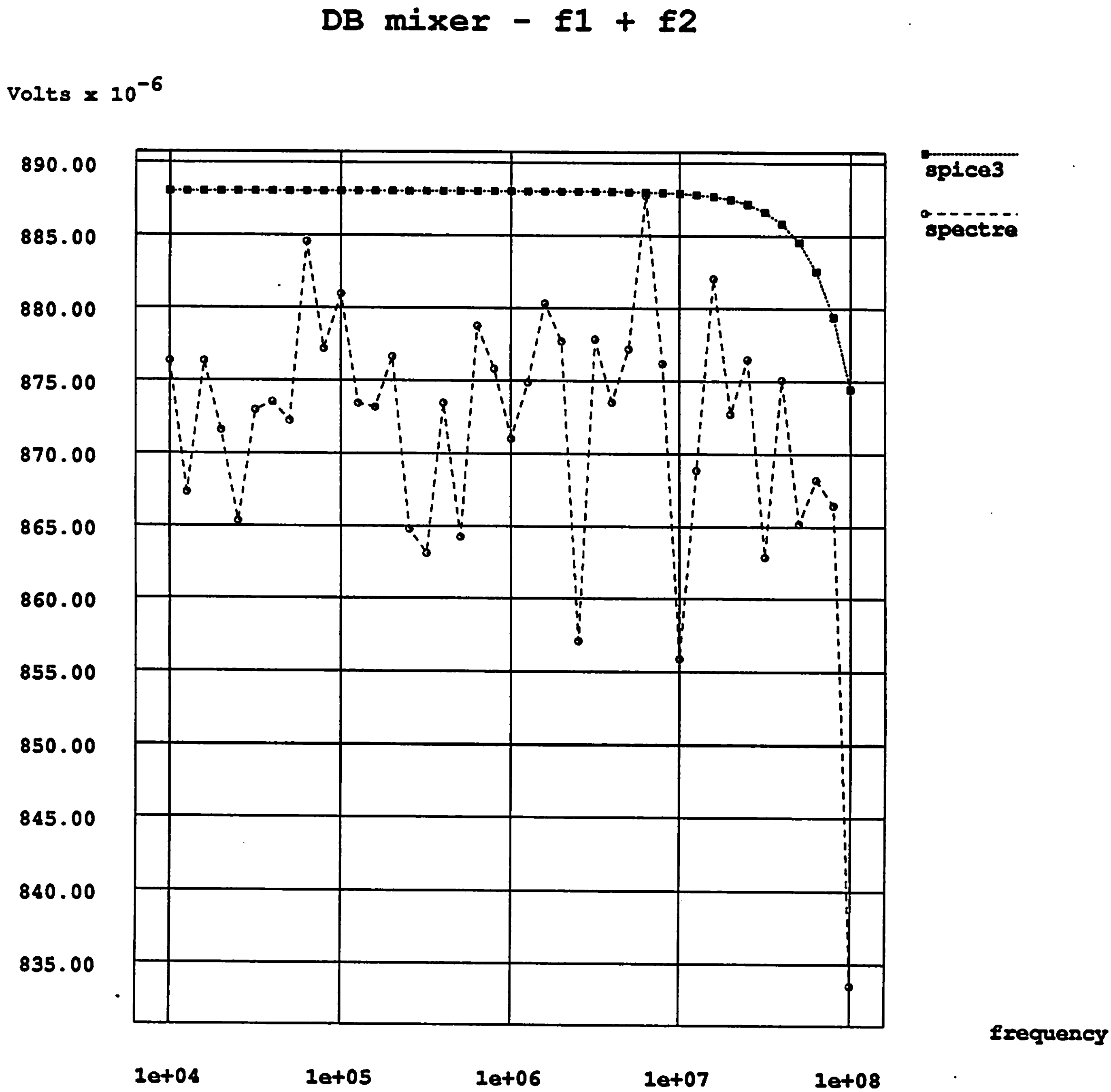

Figure D.55: Double Balanced Mixer - IM : $f_1 + f_2$

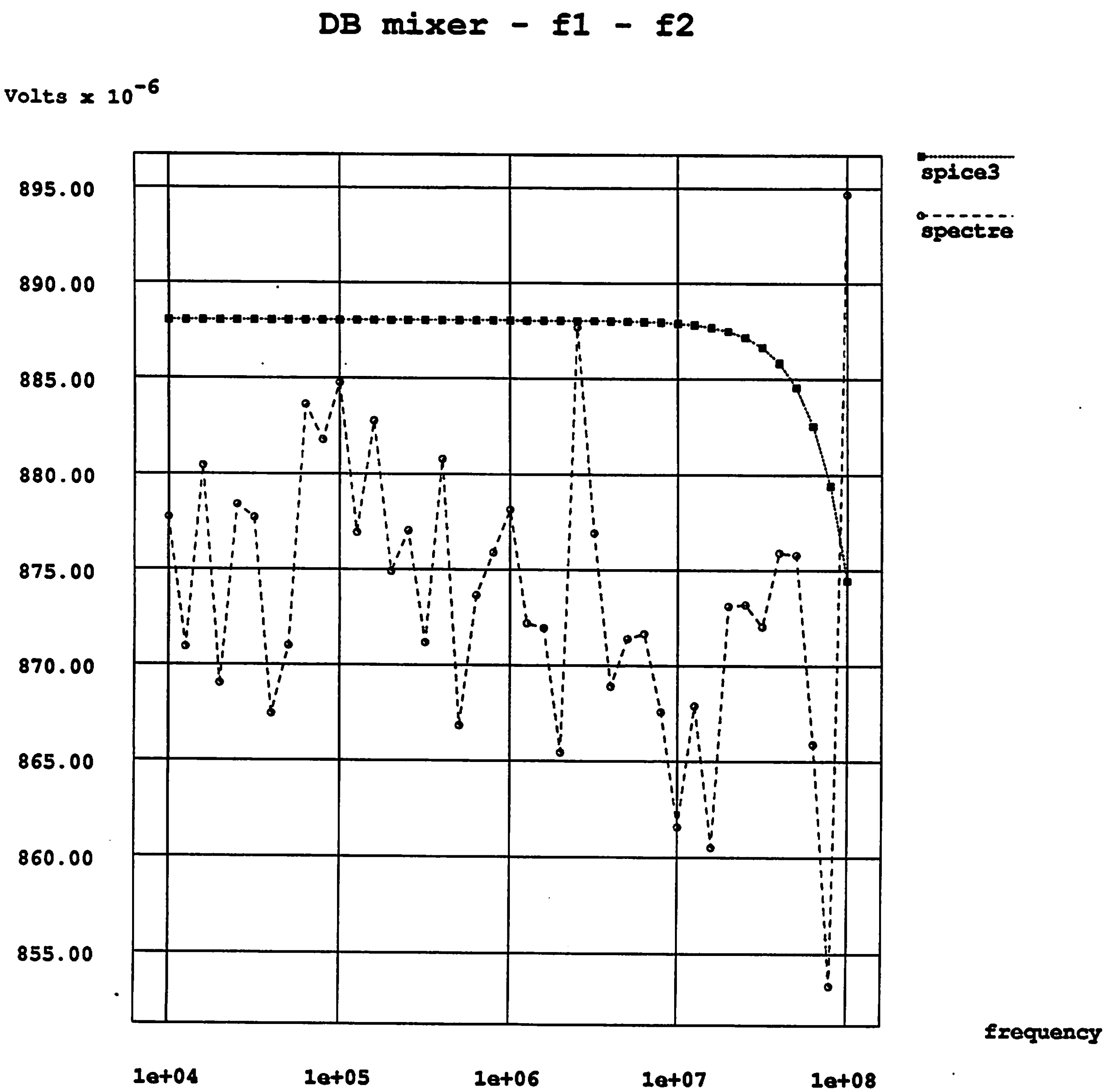

Figure D.56: Double Balanced Mixer - IM : $f_1 - f_2$

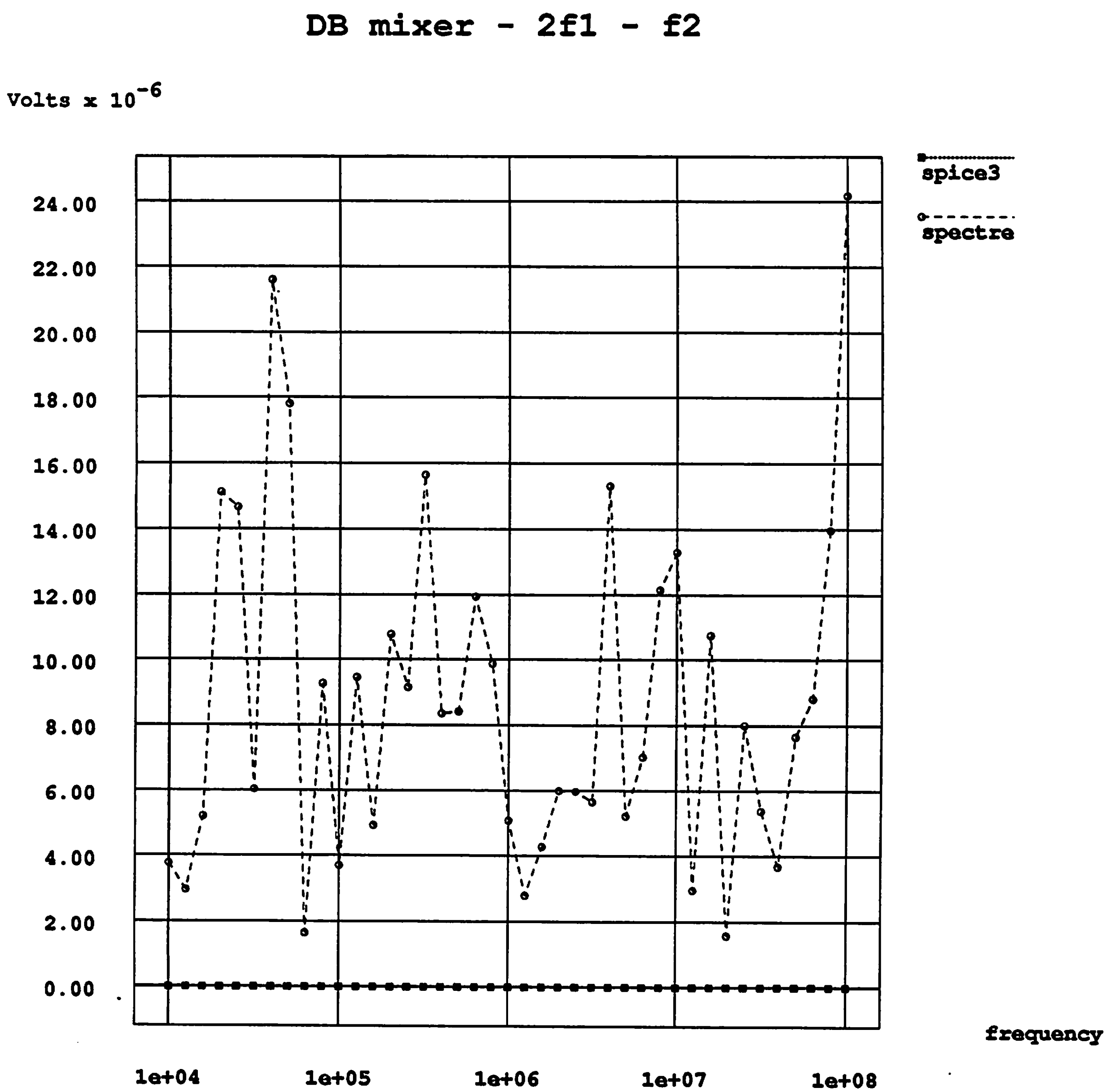

Figure D.57: Double Balanced Mixer - IM : $2f_1 - f_2$