Injection Current Model Fit in p-Channel Floating Gate MOS Transistors Using the Levenberg-Marquardt Method

O. Hernández-Garnica[1], F. Gómez-Castañeda[1], J.A. Moreno-Cadenas[1], L.M. Flores-Nava[1]
[1]Department of Electrical Engineering, CINVESTAV-IPN, Mexico D.F., Mexico
Phone (52) 55 5747 3800 Ext. 6261
E-mail: ohernandezgarnica@gmail.com; {fgomez, jmoreno, lmflores}@cinvestav.mx

Abstract — **The Levenberg-Marquardt optimization method is an iterative numerical technique that can be used to deal with the problem of least squares curve fitting. In particular, in this work it is used to fit the parameters involved in a semi-empirical model of the pFGMOS describing the hot-electron injection current. The injection current is a physical mechanism by which the electrical charge on the floating gate in pFGMOS transistors can be modified. This paper describes the use of the Levenberg-Marquardt algorithm for processing measurement data from an experimental pFGMOS cell fabricated in 0.5-micron standard CMOS technology.**

Keywords — **Levenberg-Marquardt; curve fitting; FGMOS; pFGMOS; electron injection; impact ionization; floating gate; MOS.**

I. INTRODUCTION

The Levenberg-Marquardt (LM) curve-fitting method can be better understood as a combination of two other optimization methods namely, the gradient descent (also known as steepest descent) and the Gauss-Newton method. It considers a set of measured data points, say *m*, and a mathematical model with a set of *n* parameters or constants to fit. There is no information *a priori* whether the mathematical model is suitable to fit the experimental data or the number of parameters used in the model is the best. Nonetheless, the LM method can be used to validate the proposed mathematical model by finding the values of the parameters involved that best fit the measured data within some acceptable error.

The model whose parameters will be optimized using the LM method describes the injection of electrons flowing into the floating gate of the p-channel Floating Gate MOS Transistor (pFGMOS), see Fig. 1(b). This model has been proposed in [2] according to the expression (1).

$$y = I_{inj} = \alpha I_S exp\left(-\frac{\beta}{(\delta+v_{gd})^2} + \gamma v_{sd}\right) \qquad (1)$$

where:

I_{inj} – Electron injection current into the floating gate.
I_s - Channel current in the pFGMOS.
v_{gd} - Gate-Drain voltage difference.
v_{sd} - Source-Drain voltage difference.
$\alpha, \beta, \gamma, \delta$ - Parameters (or constants) to fit.

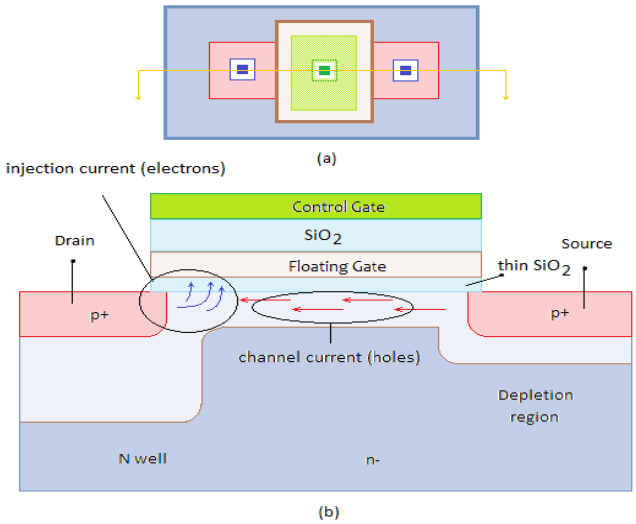

Fig. 1 The pFGMOS transistor. (a) Simplified layout. (b) Cross section, where curved blue lines represent the electron injection current heading to the floating gate according to Eq. **(1)**.

Fig. 1(a) illustrates a typical representative layout of a pFGMOS structure from which the cross-sectional view in Fig. 1(b) can be obtained; electrons generated by impact of holes near the drain can gain energy enough to surpass the thin SiO_2 energy barrier to be injected into the floating gate.

II. INJECTION MODEL FITTING USING THE LM METHOD

The injection model (1) can be conceived as a function of a vector $\boldsymbol{p_0} = (\alpha, \beta, \gamma, \delta) = (p_{01}, p_{02}, p_{03}, p_{04}) \in \mathbb{R}^4$ of parameters to optimize, so for the *k-th* measurement the model predicts the value:

$$y(t_k; \boldsymbol{p_0}) = p_{01} I_{sk} exp\left(-\frac{p_{02}}{(p_{04}+v_{gdk})^2} + p_{03} v_{sdk}\right) \qquad (2)$$

where $k = 1,\ldots,m$.

The LM method considers a *figure of merit function* (error function for short) that estimates the sum of squared weighted differences between the *m* measured injection currents y_k and the values of the model to fit for an initial guess of the parameter vector $\boldsymbol{p_0}$, according to:

$$\chi^2(\boldsymbol{p}) = \frac{1}{2}\sum_{k=1}^{m}\left(\frac{y_k - y(t_k; \boldsymbol{p_0})}{w_k}\right)^2 \qquad (3)$$

where:

IEEE Catalog Number CFP13827-ART
ISBN 978-1-4799-1461-6
978-1-4799-1461-6/13/$31.00 ©2013 IEEE

y_k - is the k-th measured injection current.
w_k - is the k-th weight.

Or in vector notation:

$$\chi^2(\boldsymbol{p}) = \frac{1}{2}[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p})]^T_{1xm}\boldsymbol{W_{mxm}}[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p})]_{mx1} \quad (4)$$

$$\forall\, \boldsymbol{y}, \hat{\boldsymbol{y}}(\boldsymbol{p}) \in \mathbb{R}^m, \forall\, \boldsymbol{p} \in \mathbb{R}^n$$

where:

$$\boldsymbol{W_{mxm}} = \begin{bmatrix} 1/w_{11}^2 & 0 & \dots & 0 \\ 0 & 1/w_{22}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/w_{mm}^2 \end{bmatrix} \quad (5)$$

Matrix (5) is a weighting squared diagonal matrix. The weight values are proposed to be proportional to the mean squared measurement error. The dimension of vectors and matrices are explicitly indicated in their respective sub indexes just for the sake of clarity.

The convergence of the algorithm can be observed through the error value $\chi^2(\boldsymbol{p})$ which is expected to decrease with each iterative step as better parameter values are found, as can be observed in (3). If the gradient of this error, which is as function of $\boldsymbol{p}$, is estimated, the direction of the resulting vector will give valuable information about the steepest ascend.

Without loss of generality, for the *j-th* parameter the rate of change of the error function can be depicted as:

$$\begin{aligned} \frac{\partial}{\partial p_j}\chi^2(\boldsymbol{p}) &\Rightarrow \frac{\partial}{\partial p_j}\left\{\frac{1}{2}\sum_{k=1}^{m}\left(\frac{y_k - y(t_k;\boldsymbol{p})}{w_k}\right)^2\right\} \\ &= \frac{1}{2}\sum_{k=1}^{m}\frac{\partial}{\partial p_j}\left(\frac{y_k - y(t_k;\boldsymbol{p})}{w_k}\right)^2 \\ &= \frac{1}{2}\sum_{k=1}^{m} 2\left(\frac{y_k - y(t_k;\boldsymbol{p})}{w_k}\right)\frac{\partial}{\partial p_j}\left(\frac{y_k - y(t_k;\boldsymbol{p})}{w_k}\right) \\ &= \sum_{k=1}^{m}\left(\frac{y_k - y(t_k;\boldsymbol{p})}{w_k}\right)\frac{\partial}{\partial p_j}\left(-\frac{y(t_k;\boldsymbol{p})}{w_k}\right) \\ &= -\sum_{k=1}^{m}\left(\frac{y_k - y(t_k;\boldsymbol{p})}{w_k}\right)\frac{\partial}{\partial p_j}\left(\frac{y(t_k;\boldsymbol{p})}{w_k}\right) \end{aligned} \quad (6)$$

By observing this rate of change in (6) the gradient can then be sketched in vector notation as:

$$\nabla\chi^2(\boldsymbol{p}) = \left[\frac{\partial}{\partial p_1}\chi^2(\boldsymbol{p}), \frac{\partial}{\partial p_2}\chi^2(\boldsymbol{p}), \cdots, \frac{\partial}{\partial p_n}\chi^2(\boldsymbol{p})\right]$$

$$\nabla\chi^2(\boldsymbol{p}) = -[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p})]^T_{1xm}\boldsymbol{W_{mxm}}\boldsymbol{J_{mxn}} \quad (7)$$

where:

$$\boldsymbol{J_{mxn}} = \begin{bmatrix} \frac{\partial}{\partial p_1}y(t_1;\boldsymbol{p}) & \frac{\partial}{\partial p_2}y(t_1;\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_n}y(t_1;\boldsymbol{p}) \\ \frac{\partial}{\partial p_1}y(t_2;\boldsymbol{p}) & \frac{\partial}{\partial p_2}y(t_2;\boldsymbol{p}) & \cdots & \frac{\partial}{\partial p_n}y(t_2;\boldsymbol{p}) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial p_1}y(t_m;\boldsymbol{p}) & \frac{\partial}{\partial p_2}y(t_m;\boldsymbol{p}) & \dots & \frac{\partial}{\partial p_n}y(t_m;\boldsymbol{p}) \end{bmatrix} \quad (8)$$

Equation (8) is the Jacobian matrix of size *m* x *n* (having *m* measurements and *n* model parameters) evaluated at vector $\boldsymbol{p}$ which represents the rate of change of the model to fit in variation of each parameter component.

To find the steepest descent using (7), a vector of the opposite direction must be considered, so if a vector $\boldsymbol{p_1}$ should be found better than $\boldsymbol{p_0}$ then a small perturbation vector $\boldsymbol{\delta h}$ around $\boldsymbol{p_0}$ must be pointing towards the following direction:

$$\boldsymbol{\delta h} = \alpha[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p})]^T_{1xm}\boldsymbol{W_{mxm}}\boldsymbol{J_{mxn}} \quad (9)$$

This perturbation vector $\boldsymbol{\delta h}$ moves the parameter vector $\boldsymbol{p}$ just in the direction of the steepest descent and the scalar $\alpha > 0$ determines the length of the step in the steepest descent. Note that the vector in (7) is just in the opposite direction of that of (9). Equation (9) is just the contribution of the gradient descent method to the LM algorithm.

The Gauss-Newton contribution to the LM method comes from assuming that the model to fit $y(t_k;\boldsymbol{p})$ is approximately quadratic in the parameter vector in the neighborhood of the optimal solution. The first order Taylor series for the model $y(t_k;\boldsymbol{p})$ around some vector $\boldsymbol{p_i}$ will be an acceptable approximation to consider, as given by:

$$\hat{\boldsymbol{y}}(\boldsymbol{p})_{mx1} \approx \hat{\boldsymbol{y}}(\boldsymbol{p_i})_{mx1} + \boldsymbol{J}(\boldsymbol{p_i})[\boldsymbol{p} - \boldsymbol{p_i}]_{nx1} \quad (10)$$

If $\boldsymbol{\delta h} = \boldsymbol{p} - \boldsymbol{p_i}$ then:

$$\hat{\boldsymbol{y}}(\boldsymbol{p} + \boldsymbol{\delta h}) \approx \hat{\boldsymbol{y}}(\boldsymbol{p_i}) + \mathbf{J}\boldsymbol{\delta h} \quad (11)$$

By substituting (11) in (4) a new function of the perturbation vector can be observed:

$$\chi^2(\boldsymbol{p} + \boldsymbol{\delta h}) \approx \frac{1}{2}[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p} + \boldsymbol{\delta h})]^T\boldsymbol{W}[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p} + \boldsymbol{\delta h})]$$

$$\chi^2(\boldsymbol{p} + \boldsymbol{\delta h}) \approx \tfrac{1}{2}[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p}) - \mathbf{J}\boldsymbol{\delta h}]^T\boldsymbol{W}[\boldsymbol{y} - \hat{\boldsymbol{y}}(\boldsymbol{p}) - \mathbf{J}\boldsymbol{\delta h}] \quad (12)$$

The perturbation vector $\boldsymbol{\delta h}$ that minimizes $\chi^2(\boldsymbol{p} + \boldsymbol{\delta h})$ can be estimated from the gradient:

IEEE Catalog Number CFP13827-ART
ISBN 978-1-4799-1461-6
978-1-4799-1461-6/13/$31.00 ©2013 IEEE

$$\nabla\chi^2(\boldsymbol{p}+\boldsymbol{\delta h}) = \frac{\partial}{\partial \boldsymbol{\delta h}}\chi^2(\boldsymbol{p}+\boldsymbol{\delta h}) = 0 \qquad (13)$$

As can be seen in (12), (13) involves a series of multiplications of terms, from which only those related to the perturbation vector will be considered in the following equation:

$$\frac{\partial}{\partial \boldsymbol{\delta h}}\chi^2(\boldsymbol{p}+\boldsymbol{\delta h}) \approx$$

$$\approx \frac{\partial}{\partial \boldsymbol{\delta h}}\left\{\frac{1}{2}[\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p})-\boldsymbol{J\delta h}]^T\boldsymbol{W}[\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p})-\boldsymbol{J\delta h}]\right\}$$

$$\approx \frac{1}{2}\frac{\partial}{\partial \boldsymbol{\delta h}}\{-2\boldsymbol{y}^T\boldsymbol{WJ\delta h}+2\hat{\boldsymbol{y}}(\boldsymbol{p})^T\boldsymbol{W}\,\boldsymbol{J\delta h}+\boldsymbol{\delta h}^T\boldsymbol{J}^T\boldsymbol{W}\,\boldsymbol{J\delta h}\}$$

$$\approx -[\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p})]^T\boldsymbol{WJ}+\boldsymbol{\delta h}^T\boldsymbol{J}^T\boldsymbol{W}\,\boldsymbol{J}=0 \qquad (14)$$

This last conclusion will give the condition for $\boldsymbol{\delta h}$ that will make the error function the minimum, resulting:

$$\boldsymbol{\delta h}^T\left[\boldsymbol{J}^T\boldsymbol{W}\,\boldsymbol{J}\right] = [\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p})]^T\boldsymbol{WJ} \qquad (15)$$

II. The Levenberg-Marquardt Algorithm

The core of the Levenberg-Marquardt algorithm can be expressed as:

$$\boldsymbol{\delta h}^T\left[\boldsymbol{J}^T\boldsymbol{WJ}+\lambda\boldsymbol{I}\right] = [\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p})]^T\boldsymbol{WJ} \qquad (16)$$

where small values of λ result in a Gauss-Newton update and large values of λ result in a steepest descent update. Within the iterative process of the algorithm there is an adaptive variation of λ so as the solution approaches the optimum value for $\vec{p}$ then λ is decreased and the algorithm becomes a Gauss-Newton update.

In this work, (16) is applied to the injection model (1) along with its parameter vector $\boldsymbol{p_0} = (\alpha, \beta, \gamma, \delta) \in \mathbb{R}^4$ using a set of *m* measurements made from the experimental cell described in the following section. The main logic structure of the LM algorithm can be synthesized in the following pseudo-code.

LM pseudo code:

1. Propose an initial guess for parameter vector $\boldsymbol{p}$ e.g.: $\boldsymbol{p_0} = (1,1,1,1)$.
2. Propose a modest initial value for λ, e.g.: $\lambda = 0.001$.
3. Propose weight values proportional to the mean squared error:
   $$w_{ii}^2 = [\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p_0})]_{1xm}^T[\boldsymbol{y}-\hat{\boldsymbol{y}}(\boldsymbol{p_0})]_{mx1}$$
4. For each *i-th* iteration *(i=1, 2, …., k)*:
   - 4.1 Estimate $\chi^2(\boldsymbol{p_i})$ from (3).
   - 4.2 Find the perturbation $\boldsymbol{\delta h_i}$ from (16).
   - 4.3 Estimate the new vector $\boldsymbol{p_{i+1}} = \boldsymbol{p_i} + \boldsymbol{\delta h_i}$.
   - 4.4 If $\chi^2(\boldsymbol{p_{i+1}}) \geq \chi^2(\boldsymbol{p_i})$ increase λ by a factor of 10 (this value can be modified to test the convergence speed). No updates are taken for $\boldsymbol{p_{i+1}}$, instead go to step 4.2.
   - 4.5 If $\chi^2(\boldsymbol{p_{i+1}}) < \chi^2(\boldsymbol{p_i})$ decrease λ by a factor of 5 (this value can be modified to test the convergence speed). Update $\boldsymbol{p_{i+1}}$.
   - 4.6 Estimate the error.
   - 4.7 Go to step 4.1 to estimate $\chi^2(\boldsymbol{p_{i+1}})$.

The Jacobian matrix $\boldsymbol{J_{mxn}}$ can be numerically approximated through the small differences between $\boldsymbol{p_{i+1}}$ and $\boldsymbol{p_i}$ as:

$$j_{ij} = \frac{\partial \hat{y}_i}{\partial p_j} = \frac{\hat{y}(t_i;\, \boldsymbol{p_i}+\boldsymbol{\delta h_i}) - \hat{y}(t_i;\, \boldsymbol{p_i})}{\|\boldsymbol{\delta h_i}\|} \qquad (17)$$

It is possible to control a set of parameters of the iterative process to improve the convergence of the algorithm, for example, we set the maximum limit of iterations to be 200; the value of λ when increased (step 3.4) is different when decreased (step 3.5); a minimum error figure also might be considered. There are excellent references about the code implementation of the LM algorithm in **[1]** and **[3]**.

III. Experimental Cell

A set of 256 measurements were made using the test cell of Fig. 2, where the channel current of transistor $\mathbf{M_{inj}}$ is held constant through the current source **Is** and its source to drain potential **Vsd** is also held constant through **Vref**. With the imposed conditions of constant **Is** and **Vsd**, the voltage **Vgd** is self biased to fulfill its imposed conditions. In these static conditions a constant injection current is held as can be seen from (1). The operational amplifier in the control gate provides a linear change of voltage **[9]** at node **Vout**, as a result of the dynamic integration of the injected current through capacitor **Cin**. By taking the slope of this linear change at **Vout** in different time instants it is possible to indirectly estimate the injected current by considering:

$$i_{inj} = C_{fg}\frac{dV_{fg}}{dt} = C_T\frac{V_{fg2}-V_{fg1}}{t_2-t_1} = C_T\frac{\alpha V_{out2}-\alpha V_{out1}}{t_2-t_1} = C_{in}\frac{\Delta V_{out}}{\Delta t} \qquad (18)$$

where:

C_T - Total capacitance seen by the floating gate including parasitic capacitances.

IEEE Catalog Number CFP13827-ART
ISBN 978-1-4799-1461-6
978-1-4799-1461-6/13/$31.00 ©2013 IEEE

(V_{fg1}, t_1) - Floating gate voltage at instant time t_1.
(V_{fg2}, t_2) - Floating gate voltage at instant time t_2.
C_{fg} – Floating gate capacitance.
α - Coupling capacitive factor from V_{out} to V_{fg}: $\alpha = \frac{C_{in}}{C_T}$

In essence, it is possible to measure the injection current just by observing the voltage change in **Vout** within some finite time as can be seen from (18).

The geometry of the pFGMOS within our experimental cell was L = 0.6µm, W = 0.9µm. It was fabricated in 0.5-micron, n-well, standard CMOS MOSIS technology.

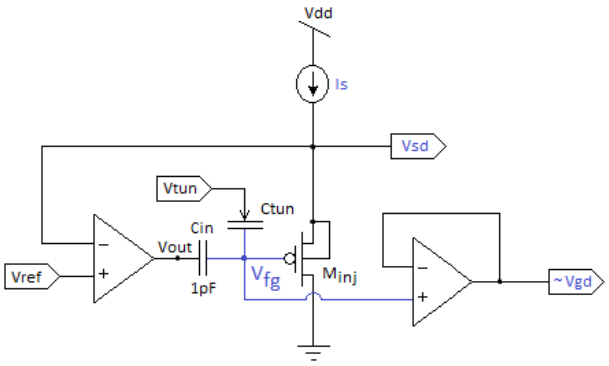

Fig. 2 Experimental test cell used to measure parameters in model **(1)**.

To collect useful data to achieve the curve fit, a voltage **Vref** was held constant while a set of 32 discrete values for **Is** where used to obtain its corresponding **Iinj** and **Vgd** values. The purpose of this strategy is to fully describe (1) for a discrete set of currents **Is** except for the parameter values to fit. By following this strategy through a discrete set of voltages **Vsd** it is possible to collect a useful dataset to use in the LM algorithm. Fig.(3) shows the dataset collected for 8 different **Vsd** voltages, each containing 32 discrete values for **Is** and **Iinj**.

As a result of the injection process, the floating gate voltage is diminished as a result of the increasing number of electrons **[5], [8]**, so for every set of 32 injection currents a tunneling pulse **Vtun** of ~15.4V was applied during 10 seconds to rise the floating gate voltage to ~2.4V, making the cell ready for another round of 32 injection currents.

IV. RESULTS

The measured data set (256 points) can be observed in circles in Fig. 3 which is a log **Iinj** vs. log **Is** plot. The continuous lines are the mathematical model following (19) having the parameters found by the LM algorithm.

$$i_{inj} = 70x10^{-5} I_s \exp\left(-\frac{1415}{(v_{gd}+4.683)^2} + 1.695 v_{sd}\right) \qquad (19)$$

The LM algorithm was executed several times changing the initial parameter vector $\boldsymbol{p_0} = (1,1,1,1)$ at each trial as lower values of the error function $\chi^2(\boldsymbol{p_0})$ were found.

Fig. 4 shows the evolution of the error function $\chi^2(\boldsymbol{p})$ in a semi-log plot, which is used to observe the convergence of the algorithm and the values of λ adopted within the iterative process. It is possible to test different values for increasing lambda and decreasing lambda to improve the convergence of the algorithm.

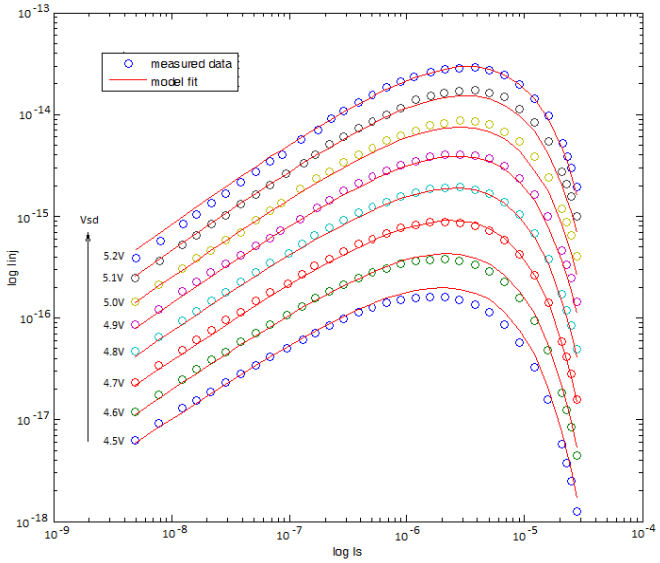

Fig. 3 Model fit comparison with the measured data set. Each curve contains 32 measured (**Is**, **Iinj**) data points (circles) for each different **Vsd** proposed.

Also, it is possible to observe the evolution of the parameter vector through the iterative process of the algorithm as show in Fig. 5.

IEEE Catalog Number CFP13827-ART
ISBN 978-1-4799-1461-6
978-1-4799-1461-6/13/$31.00 ©2013 IEEE

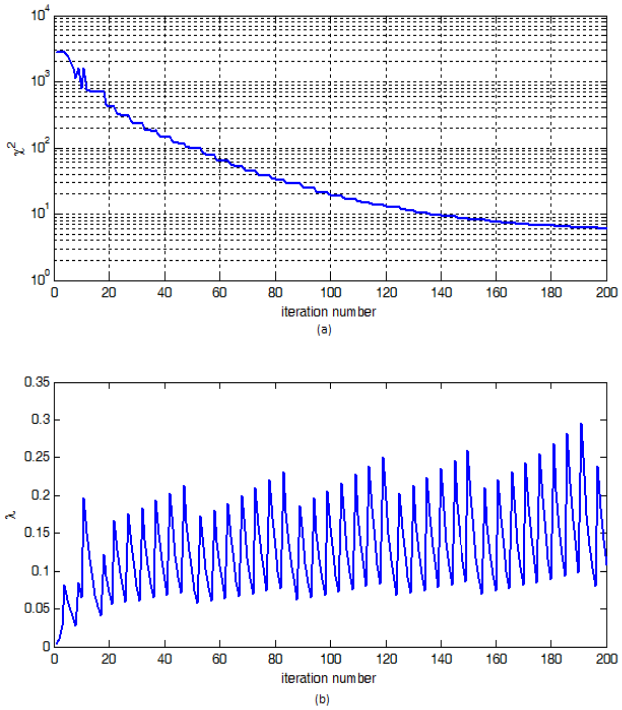

Fig. 4 (a) Error function $\chi^2(\vec{p})$ evolution; at each iteration the initial parameter vector was modified by a $\overrightarrow{\delta h}$ making the error smaller in the iterative process. (b) Evolution of λ according to steps 3.4 and 3.5 of the pseudo-code, starting from 0.001.

Through the several runs of the algorithm trying to find a better fitting, the final values observed in the parameter vector shown in Fig. 5 were taken in the initial parameter vector of the next run of the algorithm, in this manner we confirmed that the error was constantly decreasing and confirming also the convergence of the algorithm.

The model fitting was validated by considering the figure of merit function to decrease below some accepted error value. This does not tell *per se* whether we have found a local minimum nor a global minimum.

As proposed in **[2]**, Fig. (6) shows the macro model circuit that can be used to implement the electron injection current by using a voltage controlled current source (VCCS) **Ginj** embedding the model fit (19). The additional voltage-controlled voltage source with series resistor is used to improve the convergence in Spice. Transistor $\mathbf{M_{tun}}$ models the MOS capacitor used for tunneling.

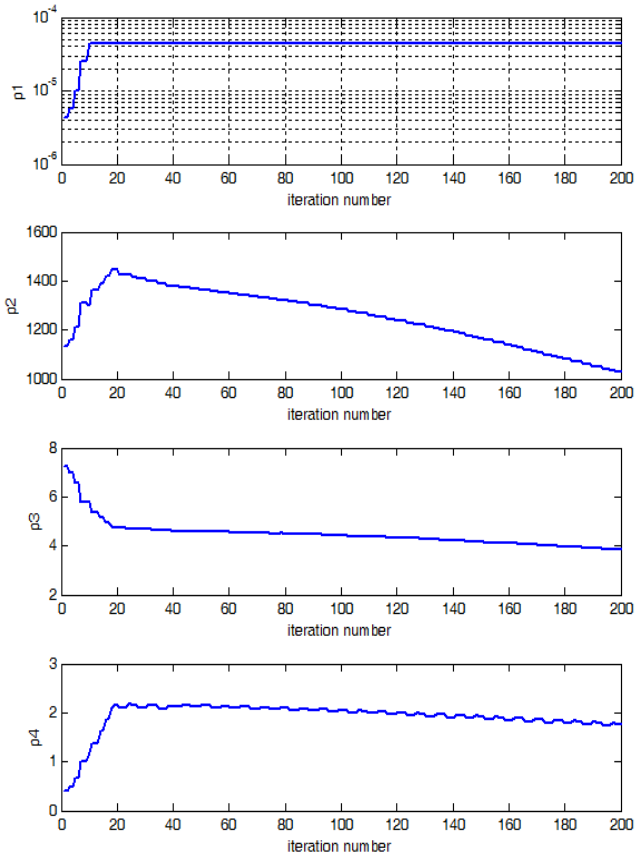

Fig. 5 Evolution of the parameters involved in the fitting of **(2)**. These values are not the best fit found of **(2)**, are intermediate values evolving in the iterative process.

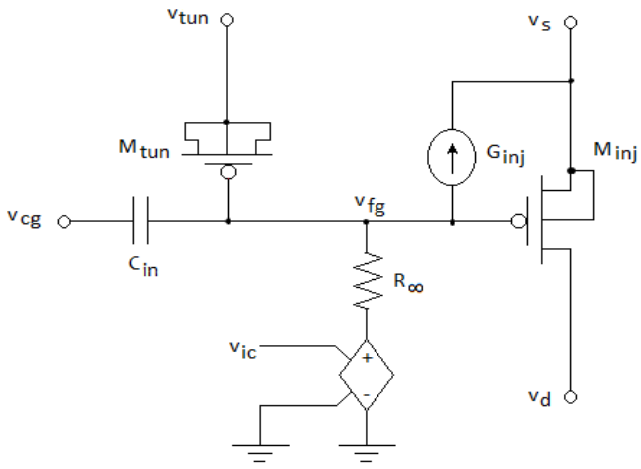

Fig. 6 pFGMOS circuit macro model used to simulate the electron injection current. The voltage **Vic** is used to set any initial voltage condition in the floating gate. This macro model is useful in standard transient analysis of Spice simulators. DC simulations are likely to have convergence issues **[4]**.

IEEE Catalog Number CFP13827-ART
ISBN 978-1-4799-1461-6
978-1-4799-1461-6/13/$31.00 ©2013 IEEE

V. CONCLUSION

The Levenberg-Marquardt method can be used as an iterative algorithm to fit nonlinear mathematical models to experimental data within some accepted error criterion. The method provides robust and rapid convergence. As an example of the implementation of this method, the hot electron injection semi-empirical model proposed by Chris Diorio *et al* **[2]** has been fitted to our experimental measurements of injected currents for a test cell fabricated in the 0.5-micron, standard CMOS MOSIS technology. A method for measuring the injection current was also provided.

As shown in this work, the main result through the LM curve fitting is to provide a circuit macro model suitable for transient simulation of the hot electron injection phenomenon, in particular for the low-cost 0.5-micron, standard CMOS technology. The ability to design and simulate floating gate programming cells for 0.5-micron CMOS technology using the proposed fit would also be a valuable advantage **[6]**.

ACKNOWLEDGMENTS

We would like to thank to all participants, advisors, classmates, colleagues, enthusiasts and fellows directly and indirectly involved in the set of experiments, discussions, simulations, reflections and algorithmic implementations. Also we would like to thank the reader for having enough patience and good mood to analyze this work. Constructive comments are always welcomed.

Special and infinite thanks to my very supportive, kind and lovely family.

REFERENCES

[1] Henri Gavin, "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problem", Department of Civil and Environmental Engineering, Duke University, 2011.

[2] Chris Diorio, Cecilia Hernández, M. Dean Brockhausen, Kambiz Rahimi "A simulation model for floating gate MOS synapse transistors*" IEEE Trans. Electron Devices*, vol.2, 2002, pp. 532–535.

[3] W. Press, S. Teukolsky, W. Vetterling and Brian Flannery "Numerical Recipes in C" 2nd Edition, Cambridge University Press 1992.

[4] Esther Rodriguez-Villegas, "Low Power and Low Voltage Circuit Design with the FGMOS transistor", The Institution of Engineering and Technology, London, United Kingdom, 2006.

[5] Shih-Chii Liu, Jörg Kramer, Giacomo Indiveri, Tobias Delbrück, and Rodney Douglas , "Analog VLSI: Circuits and Principles", Massachusetts Institute of Technology 2002.

[6] Paul Hasler, Bradley A. Minch, Chris Diorio, "Floating-gate devices: They are not just for digital memories anymore", *in Proc. IEEE. Int. Symp. Circuits Syst*., 1999, pp. 388-391.

[7] C.A. Mead, Analog VLSI and Neural Systems, Addison-Wesley, Reading, MA, 1989.

[8] Chris Diorio, "A p-Channel MOS Synapse Transistor with Self-Convergent Memory Writes", *IEEE Transactions on Electron Devices*, Vol. 47, No. 2, pp. 464-472, February 2000.

[9] Chenling Huang, Pikul Sarkar and Shantanu Chakrabartty, "Rail-to-rail, linear hot-electron injection programming of floating-gate voltage bias generators at 13-bit resolution", *IEEE J. Solid-State Circuits*, vol. 46, no. 11, pp. 2685-2692, Nov.2011.

IEEE Catalog Number CFP13827-ART
ISBN 978-1-4799-1461-6
978-1-4799-1461-6/13/$31.00 ©2013 IEEE