Bergische Universität Wuppertal

Fachbereich Mathematik und Naturwissenschaften

Institute of Mathematical Modelling, Analysis and
Computational Mathematics (IMACM)

Preprint BUW-IMACM 15/10

B. Tasić, J.J. Dohmen, E.J.W. ter Maten, T.G.J. Beelen,
H.H.J.M. Janssen, W.H.A. Schilders, M. Günther

Fast Fault Simulation to identify subcircuits involving faulty components

February 2015

http://www.math.uni-wuppertal.de

Fast Fault Simulation to identify subcircuits involving faulty components

B. Tasić, J.J. Dohmen, E.J.W. ter Maten, T.G.J. Beelen, H.H.J.M. Janssen, W.H.A. Schilders, M. Günther

Abstract Imperfections in manufacturing processes may cause unwanted connections (faults) that are added to the nominal, "golden", design of an electronic circuit. By fault simulation we simulate all situations: new connections and each with different values for the newly added element. We also consider "opens" (broken connections). During the transient simulation the solution of a faulty circuit is compared to the golden solution of the fault-free circuit. A strategy is developed to efficiently simulate the faulty solutions until their moment of detection. We fully exploit the hierarchical structure of the circuit in the simulation process to bypass parts of the circuit that appear to be unaffected by the fault. Accurate prediction and efficient solution procedures lead to fast fault simulation in which the golden solution and all faulty solutions are calculated over a same time step. Finally, we store a database with detectable deviations for each fault. If such a detectable output "matches" a measurement result of a product that has been returned because of malfunctioning it helps to identify the subcircuit that may contain the real fault.

1 Time integration of circuit equations

The electronic circuit equations can be written as [4, 10]

B. Tasić, J.J. Dohmen, T.G.J. Beelen, H.H.J.M. Janssen
NXP Semiconductors, High Tech Campus 46, 5656 AE Eindhoven, the Netherlands. e-mail: {Bratislav.Tasic,Jos.J.Dohmen,Theo.G.J.Beelen,Rick.Janssen}@nxp.com

E.J.W. ter Maten, T.G.J. Beelen, W.H.A. Schilders
Eindhoven University of Technology, P.O.Box 513, 5600 MB Eindhoven, the Netherlands. e-mail: {E.J.W.ter.Maten,T.G.J.Beelen,W.H.A.Schilders}@tue.nl

E.J.W. ter Maten, M. Günther
Bergische Universität Wuppertal, Gaußstraße 20, D-42219 Wuppertal, Germany. e-mail: {Jan.ter.Maten,Michael.Guenther}@math.uni-wuppertal.de

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(\mathbf{x})+\mathbf{j}(\mathbf{x})=\mathbf{s}(\mathbf{x},t). \tag{1}$$

Here $\mathbf{s}(\mathbf{x},t)$ represents the specifications of the sources. The unknown $\mathbf{x}=\mathbf{x}(t)$ consists of nodal voltages and of currents through voltage defined elements. We assume that $\mathbf{q}(\mathbf{0})=\mathbf{0}$, and $\mathbf{j}(\mathbf{0})=\mathbf{0}$.
For time integration in circuit simulation we consider the BDF1, or Euler Backward method. Assuming time points $t_{k+1}=t_k+h_k$ ($k\geq 0$) with stepsizes h_k and approximation $\mathbf{x}^n$ at t_n, BDF1 calculates $\mathbf{x}^{n+1}$ by

$$\frac{\mathbf{q}^{n+1}-\mathbf{q}^n}{h_n}+\mathbf{j}^{n+1}=\mathbf{s}^{n+1}. \tag{2}$$

Here $\mathbf{q}^k=\mathbf{q}(\mathbf{x}^k)$, $\mathbf{j}^k=\mathbf{j}(\mathbf{x}^k)$, for $k=n,n+1$, and $\mathbf{s}^{n+1}=\mathbf{s}(\mathbf{x}^{n+1},t_{n+1})$. The system is solved by a Newton-Raphson procedure. For efficient direct methods to solve the intermediate linear systems, see [1]. A fixed Jacobian can reduce the number of LU-decompositions, but, in general, will increase the number of iterations and thus the number of (costly) evaluations. Also, in case of circuit simulation, the assembly of the matrices does not need much more effort when compared to the function evaluations. A fixed decomposed Jacobian can be efficient within some Picard-iteration [8] in solving a linear system, or, more general, in using it as preconditioner within GMRES. When changing stepsizes during time integration similar remarks apply.
In case of an hierarchical linear solver one can profit from hierarchical bypassing [3], which we will also exploit in this paper. When applying it also in the time integration, it even supports a first form of multirate time-integration [13].

2 Fault simulation

We first consider the effect of adding faulty, linear elements to the circuit. F.i., in [2, 12] we did add linear bridges (resistors) to the circuit. For each fault only one element is added to the original, golden circuit. It may mean a new connection, while also different values are considered. In [12] a novel time-integration was involved: during each time-step, first the fault-less, golden solution was determined at the next time step. Next, all faulty problems were integrated over this time-interval. Hence, effectively, a parameter loop is placed inside the time integration. Also the hierarchical structure was enhanced such that the hierarchical solver could deal with all new elements. This enables to exploit an enhanced form of bypassing.
The golden solution at each new time point provides an estimate for the solution of a faulty problem (in addition to the one using extrapolation by Nordsieck vectors). Each faulty problem uses the stepsize of the golden solution as a maximum one. When the faulty solution really needs a stepsize that is significantly smaller then used by the golden solution, the traditional time integration is invoked, even without bypassing, until the time moment of synchronization with the golden solution.

In this paper we enhance the algorithm in also considering the case of adding linear capacitors. However, in practise, the linear resistor case is by far more important. Hence, we either have[1]

$$\mathbf{j}(\mathbf{x}(t,p),p) = \mathbf{j}_0(\mathbf{x}(t,p)) + p\,\mathbf{u}\mathbf{v}^T\mathbf{x}(t,p), \quad \text{or} \tag{3}$$

$$\mathbf{q}(\mathbf{x}(t,p),p) = \mathbf{q}_0(\mathbf{x}(t,p)) + p\,\mathbf{a}\mathbf{b}^T\mathbf{x}(t,p). \tag{4}$$

For simplicitly, to reduce notation and the amount of partial derivatives further on, we use p, both in (3) and in (4). Fault Analysis consists of simulations for a large number of pairs of vectors $(\mathbf{u},\mathbf{v})$, or $(\mathbf{a},\mathbf{b})$ and various values of p, and compare the result $\mathbf{x}_p(t)$ of (1) at specific time points with the "golden" solution $\mathbf{x}(t)$ of the fault-free circuit (corresponding with $p=0$). If the deviation exceeds some treshold, the fault triple $(\mathbf{u},\mathbf{v},p)$, or $(\mathbf{a},\mathbf{b},p)$, is marked as detectable and is taken out of the list. Clearly, for each fault we have a new contribution $p\,\mathbf{u}\mathbf{v}^T\mathbf{x}(t,p)$, or $p\,\mathbf{a}\mathbf{b}^T\mathbf{x}(t,p)$, as low-rank modification to the system of the golden solution, either added to $\mathbf{j}_0$ or to $\mathbf{q}_0$, see (3)-(4). Here $p \geq 0$ is just a scalar, by which the $\mathbf{p}$-sensitivity 'matrix' $\hat{\mathbf{x}}_p(t,p) = \frac{\partial \mathbf{x}(t,p)}{\partial p}$ reduces to a vector.
The golden solution $\mathbf{x}(t)$ used $\mathbf{j}(\mathbf{x}(t,p),p) = \mathbf{j}_0(\mathbf{x}(t,p))$, and $\mathbf{q}(\mathbf{x}(t,p),p) = \mathbf{q}_0(\mathbf{x}(t,p))$. Let $\mathbf{x}_p^k = \mathbf{x}^k(p) \approx \mathbf{x}(t_k,p)$ be the numerical approximations for $k = n, n+1$ of the faulty system and $\hat{\mathbf{x}}_p^k$ be the corresponding sensitivities. Then with $\mathbf{C}_p^k \equiv \frac{\partial \mathbf{q}(\mathbf{x}_p^k)}{\partial \mathbf{x}}$, $\mathbf{G}_p^k \equiv \frac{\partial \mathbf{j}(\mathbf{x}_p^k)}{\partial \mathbf{x}}$ (and including the effect of the rank-one term with the factor p) and $\mathbf{S}_p^k \equiv \frac{\partial \mathbf{s}(\mathbf{x}_p^k,t_k)}{\partial \mathbf{x}}$, by sensitivity analysis [6, 11], we obtain

$$[\frac{1}{h_n}\mathbf{C}_p^{n+1} + \mathbf{G}_p^{n+1} - \mathbf{S}_p^{n+1}]\hat{\mathbf{x}}_p^{n+1} = -\frac{1}{h_n}\mathbf{a}\mathbf{b}^T(\mathbf{x}^{n+1} - \mathbf{x}^n) - \mathbf{u}\mathbf{v}^T\mathbf{x}_p^{n+1} + \frac{1}{h_n}\mathbf{C}_p^n\hat{\mathbf{x}}_p^n. \tag{5}$$

For $p=0$, (5) gives the limit sensitivity $\hat{\mathbf{x}}^k = \hat{\mathbf{x}}_0^k$ for the golden, fault-free, solution $\mathbf{x}^k = \mathbf{x}_0^k$ $(k = n, n+1)$

$$[\frac{1}{h_n}\mathbf{C}^{n+1} + \mathbf{G}^{n+1} - \mathbf{S}^{n+1}]\hat{\mathbf{x}}^{n+1} = -\frac{1}{h_n}\mathbf{a}\mathbf{b}^T(\mathbf{x}^{n+1} - \mathbf{x}^n) - \mathbf{u}\mathbf{v}^T\mathbf{x}^{n+1} + \frac{1}{h_n}\mathbf{C}^n\hat{\mathbf{x}}^n, \tag{6}$$

where $\mathbf{C}^k = \mathbf{C}_0^k$ $(k = n, n+1)$, $\mathbf{G}^{n+1} = \mathbf{G}_0^{n+1}$ and $\mathbf{S}^{n+1} = \mathbf{S}_0^{n+1}$. By Taylor expansion we additionally have

$$\mathbf{x}_p^k = \mathbf{x}^k + p\,\hat{\mathbf{x}}^k + \mathscr{O}(p^2) \quad (k = n, n+1). \tag{7}$$

The golden solution satisfies the linearized equations of the fault-free circuit up to a term $\mathbf{R}$ that indicates the deviation from linearity (note that in (1) we did assume that $\mathbf{q}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{j}(\mathbf{0}) = \mathbf{0}$)

[1] Note that for inductors and for voltage-defined resistors we need two rank-one updates to describe the total contribution.

$$[\frac{1}{h_n}\mathbf{C}^{n+1}+\mathbf{G}^{n+1}-\mathbf{S}^{n+1}]\mathbf{x}^{n+1}=\mathbf{r}(t_{n+1},\mathbf{x}^n,\mathbf{x}^{n+1}), \tag{8}$$

where $\mathbf{r}(t_{n+1},\mathbf{x}^n,\mathbf{x}^{n+1})=\mathbf{s}^{n+1}+\frac{1}{h_n}\mathbf{C}^n\mathbf{x}^n+\mathbf{R}$. With (7) and (6) this gives

$$\begin{aligned}
&[\frac{1}{h_n}\mathbf{C}^{n+1}+\mathbf{G}^{n+1}-\mathbf{S}^{n+1}]\mathbf{x}_p^{n+1} \;= \\
&= [\frac{1}{h_n}\mathbf{C}^{n+1}+\mathbf{G}^{n+1}-\mathbf{S}^{n+1}]\mathbf{x}^{n+1}+p[\frac{1}{h_n}\mathbf{C}^{n+1}+\mathbf{G}^{n+1}-\mathbf{S}^{n+1}]\hat{\mathbf{x}}^{n+1}+\mathscr{O}(...), \\
&= \mathbf{r}(t_{n+1},\mathbf{x}^n,\mathbf{x}^{n+1})-\frac{p}{h_n}\mathbf{a}\mathbf{b}^T(\mathbf{x}^{n+1}-\mathbf{x}^n)-p\mathbf{u}\mathbf{v}^T\mathbf{x}^{n+1}+\frac{p}{h_n}\mathbf{C}^n\hat{\mathbf{x}}^n+\mathscr{O}(...), \\
&= -\frac{p}{h_n}\mathbf{a}\mathbf{b}^T(\mathbf{x}^{n+1}-\mathbf{x}^n)-p\mathbf{u}\mathbf{v}^T\mathbf{x}^{n+1}+\frac{1}{h_n}\mathbf{C}^n(p\hat{\mathbf{x}}^n)+\mathbf{r}(t_{n+1},\mathbf{x}^n,\mathbf{x}^{n+1})+\mathscr{O}(...), \\
&= -\frac{p}{h_n}\mathbf{a}\mathbf{b}^T(\mathbf{x}_p^{n+1}-\mathbf{x}^n)-p\mathbf{u}\mathbf{v}^T\mathbf{x}_p^{n+1}+\frac{1}{h_n}\mathbf{C}^n(\mathbf{x}_p^n-\mathbf{x}^n)+\mathbf{r}(t_{n+1},\mathbf{x}^n,\mathbf{x}^{n+1})+\mathscr{O}(...),
\end{aligned}$$

in which all $\mathscr{O}(...)$ terms are of the form $\mathscr{O}(p^2+\frac{p^2}{h_n})$. Hence

$$\begin{aligned}
&[\big(\frac{1}{h_n}\mathbf{C}^{n+1}+\mathbf{G}^{n+1}-\mathbf{S}^{n+1}\big)+\frac{p}{h_n}\mathbf{a}\mathbf{b}^T+p\,\mathbf{u}\mathbf{v}^T]\mathbf{x}_p^{n+1} \;= \\
&= \frac{p}{h_n}\mathbf{a}\mathbf{b}^T\mathbf{x}^n+\frac{1}{h_n}\mathbf{C}^n(\mathbf{x}_p^n-\mathbf{x}^n)+\mathbf{r}(t_{n+1},\mathbf{x}^n,\mathbf{x}^{n+1})+\mathscr{O}(p^2+\frac{p^2}{h_n}), \qquad (9) \\
&= \mathbf{r}(t_{n+1},\mathbf{x}^n,\mathbf{x}^{n+1})+\mathscr{O}(p^2+\frac{p^2}{h_n}+\frac{p}{h_n}). \qquad (10)
\end{aligned}$$

Note that (9) may be a more accurate alternative then (10). However, for simplicity, we just used (10), after ignoring the $\mathscr{O}(\cdot)$ terms at the right-hand side. This invites for applying the Sherman-Morrison formula [5]. Let $\mathbf{A}=\big(\frac{1}{h_n}\mathbf{C}^{n+1}+\mathbf{G}^{n+1}-\mathbf{S}^{n+1}\big)$, and $\mathbf{A}\mathbf{w}=p\mathbf{u}$, $\mathbf{A}\mathbf{c}=\frac{p}{h_n}\mathbf{a}$, and $\mathbf{A}\mathbf{y}=\frac{p}{h_n}\mathbf{C}^n\hat{\mathbf{x}}^n$. Then the sensitivity predictions for $\mathbf{x}_p^{n+1}$ become

$$\text{for (3):}\quad \mathbf{x}_p^{n+1}=\mathbf{x}^{n+1}-\frac{\mathbf{v}^T\mathbf{x}^{n+1}}{1+\mathbf{v}^T\mathbf{w}}\,\mathbf{w},\ \text{ or} \tag{11}$$

$$\mathbf{x}_p^{n+1}=(\mathbf{x}^{n+1}+\mathbf{y})-\frac{\mathbf{v}^T(\mathbf{x}^{n+1}+\mathbf{y})}{1+\mathbf{v}^T\mathbf{w}}\,\mathbf{w}, \tag{12}$$

$$\text{for (4):}\quad \mathbf{x}_p^{n+1}=\mathbf{x}^{n+1}-\frac{\mathbf{b}^T\mathbf{x}^{n+1}}{1+\mathbf{b}^T\mathbf{c}}\,\mathbf{c},\ \text{ or,} \tag{13}$$

$$\mathbf{x}_p^{n+1}=(\mathbf{x}^{n+1}+\mathbf{y})-\frac{\mathbf{b}^T\mathbf{y}}{1+\mathbf{b}^T\mathbf{c}}\,\mathbf{c}. \tag{14}$$

Note that the first term in (9) has a simplyfying effect in (14) (when compared to (12)). In this case one really needs $\mathbf{y}$ to get a first estimate that is different from $\mathbf{x}^{n+1}$. The advantage of the right-hand side in (10) is that it is independent of the solution $\mathbf{x}_p^k$ at the previous time steps. Of course, when followed by further Newton-Raphson

iterations, $\mathbf{x}_p^n$ is still needed. To judge the accuracy of the linear sensitivity prediction the nonlinear solver evaluates the circuit at the sensitivity solution and updates the solution. The difference in the initial sensitivity solution and the nonlinear update is a measure for the truncation error.
If we just stick to the prediction, we may calculate the prediction of the fault at a few selected time points, which significantly reduces the work load for the fault sensitivity analysis. We finally remark that in (11), (13) the sensitivity matrix $\hat{\mathbf{x}}^n$ is not explicitly calculated.

2.1 Modeling faulty "opens"

Next we consider a faulty resistor, with value R, in series with another, linear resistor, with value r. Clearly, this introduces an extra node n_e. If the golden system used $R(n_1,n_2) = r$, the faulty system uses $R(n_1,n_e) = R$, $R(n_e,n_2) = r$. The voltage at this new node can be simply eliminated by noting that $v(n_e) = (r\,v(n_1) + R\,v(n_2))/(r + R)$. Doing this directly, the remaining system can be formulated as in (3) in which $p = R/(r(R+r))$. If $R \to \infty$ we obtain an "open" between the nodes n_1 and n_e and $v(n_e) \to v(n_2)$. In [12] we did introduce an extra port to model bridges between models. This extra node can also become functional in providing the extra node.
For modeling a broken joint (or weld) at a node n, it is, mathematically, convenient to first split the node n into two nodes n_1 and n_2, with a simple voltage source in between for the golden circuit: $E(n_1,n_2) = 0$. Clearly this satisfies our assumption $\mathbf{j}(\mathbf{0}) = \mathbf{0}$. The faulty system uses $R(n_1,n_e) = R$, $E(n_e,n_2) = 0$. We assume local coordinates that correspond with $v(n_1)$, $v(n_2)$, $i(E)$. We deduce that the faulty system perturbs the golden system with $\mathbf{u}_1\mathbf{v}_1^T(R) + \mathbf{u}_2\mathbf{v}_2^T(R)$, in which, in local coordinates, $\mathbf{u}_1^T = (1,0,0)$, $\mathbf{v}_1^T = (-1/R,1/R,1)$, $\mathbf{u}_2^T = (0,0,1)$, $\mathbf{v}_2^T = (1,-1,R)$. This can be treated in a similar way as before.

3 Results

The FFS-algorithm has been implemented in Pstar[2]. For fault simulations in DC-simulations a significant speed-up (> 100) was obtained by exploiting bypassing and abandoning only, but inclusion of sensitivity analysis appeared essential to get significant speed up for a broad class of problems during transient simulation. Table 1 shows the speed-up by including sensitivity prediction for a LIN Converter IP Block (first part), as well as for a nonlinear control DAC (2nd part). Clearly, the linear sensitivity estimate offers an interesting speed-up. Following nonlinear corrections do reduce this effect. For the LIN Converter IP Block the effect of more iterations remains quite bounded (with 100 iterations still a speed-up of more than

[2] Pstar: in-house circuit simulator of NXP Semiconductors.

10 was found, see [12]). For the nonlinear control DAC until 5 iterations a speed-up of 10 was obtained. Further speed-up scenarios are currently considered by initiating the fault later. If one can simply skip the initial integration of the faults until $t_1 > 0$, for a large collection of faults no initial simulations have to be made. The scenarios differ in how the fault is started: suddenly, or using a smooth start-up, similar as for the source-stepping-by-transient method as described in [12]. Because of the many faults that are possible, a short start-up is a balance between efficiency and robustness.

Analysis	**LIN Converter IP Block**			**Control DAC**		
	#iterations per step Δt	**CPU Time [sec]**	**Speed up**	**#iterations per step Δt**	**CPU Time [sec]**	**Speed up**
Standard AS/DOTSS	-	100437	1	-	52513	1
Linear Sensitivity	0	458	219	0	916	58
Nonlinear Correction	5	2341	43	1	4808	11

Table 1 Speed-up by including sensitivity prediction. Left: a LIN Converter IP Block, #faults=412. Right: a Control DAC, #faults=100. See also [12].

4 Relation to Uncertainty Quantification

Interchanging the time integration loop with a parameter-sweep loop for a given pair of connection nodes, also has an interesting opportunity for Uncertainity Quantification [7, 9, 14]. F.i., when considering Stochastic Collocation in which all L_2 inner-products in parameter space are replaced by quadrature, a list of deterministic parameter values $p_k, k = 1, \ldots, K$, is defined for which the solution $\mathbf{x}(t, p_k)$ has to be calculated. Then $\mathbf{x}(t, p) = \sum_{i=0}^{m} \mathbf{v}_i(t)\phi_i(p)$, in which $\mathbf{v}_i(t) = \sum_{k=1}^{K} w_k \mathbf{x}(t, p_k)\phi_i(p_k)$. This expansion is a so-called generalized Polynomial Chaos expansion, using polynomials $\phi_i(p)$ that are orthogonal with respect to some probability density function f in the parameter space for p. For FFS, where parameter values are positive, one may think about an exponential decay (for an infinite range; here one generates Laguerre polynomials), or a (α, β)-density function (when considering a finite range for p; here one generates Jacobi-polynomials). Now, first, one can simulate the K deterministic solutions (in which one can exploit the sensitivity estimate, as described before). Next, the actual FFS is done as a post-processing action in which one compares $\mathbf{x}(t, p)$ with $\mathbf{x}(t, 0)$, at specific time moments and at circuit nodes. Note that mean and variance are cheaply provided by the $\mathbf{v}_i(t)$. During the time-integration one also has at each completed time-level t $\mathbf{x}(t, p)$ and $\partial \mathbf{x}(t, p)/\partial p$ available from the expansion.Clearly, FFS is just an example for varying particular parameters. Also more general, Stochastic Collocation, can benefit by moving the parameter loop inside the time integration loop.

Acknowledgements We acknowledge the support from the EU project nanoCOPS, Nanoelectronic COupled Problems Solutions (FP7-ICT-2013-11/619166), http://www.fp7-nanoCOPS.eu/.

References

1. Davis, T.A., Natarajan, E.P.: Sparse matrix methods for circuit simulation problems. In: B. Michielsen, J.-R. Poirier (Eds.): *Scientific Computing in Electrical Engineering SCEE 2010*, Series Mathematics in Industry, Vol. 16, Springer, pp. 3–14, 2012.
2. De Jonghe, D., Maricau, E., Gielen, G., McConaghy, T., Tasić, B., Stratigopoulos, H.: Advances in variation-aware modeling, verification, and testing of analog ICs. *Proc. DATE 2012*, pp. 1615–1620, 2012.
3. Fijnvandraat, J.G., Houben, S.H.M.J., ter Maten, E.J.W., Peters, J.M.F.: Time domain analog circuit simulation. *J. of Comput. and Applied Maths.*, **185**, pp. 441–459, 2006.
4. Günther, M., Feldmann, U., ter Maten, J.: Modelling and discretization of circuit problems. In: W.H.A. Schilders, E.J.W. ter Maten (Eds.): *Handbook of Numerical Analysis, Vol. XIII, Special Volume on Numerical methods in electromagnetics*, Elsevier BV, North-Holland, pp. 523–659, 2005.
5. Golub, G.H., Van Loan, C.F.: *Matrix Computations, Third edition*. The Johns Hopkins University Press, Baltimore, MD, 1996.
6. Ilievski, Z., Xu, H., Verhoeven, A., ter Maten, E.J.W., Schilders, W.H.A., Mattheij, R.M.M.: Adjoint transient sensitivity analysis in circuit simulation. In: G. Ciuprina, D. Ioan (Eds.): *Scientific Computing in Electrical Engineering SCEE 2006*, Series Mathematics in Industry 11, Springer, pp. 183–189, 2007.
7. Le Maître, O.P., Knio, O.M.: *Spectral methods for uncertainty quantification, with applications to computational fluid dynamics*. Springer, Science+Business Media B.V., Dordrecht, 2010.
8. Li, Z., Shi, C.-J.R.: SILCA: Spice-accurate iterative linear-centric analysis for efficient time-domain simulation of VLSI circuits with strong parasitic couplings. *IEEE Trans. on Comp.-Aided Design of Integr. Circuits and Systems (TCAD)*, **25-6**, pp. 1087–1103, 2006.
9. ter Maten, E.J.W., Pulch, R., Schilders, W.H.A., Janssen, H.H.J.M.: Efficient calculation of Uncertainty Quantification. In: M. Fontes, M. Günther, N. Marheineke (Eds): *Progress in Industrial Mathematics at ECMI 2012*, Series Mathematics in Industry Vol. 19, Springer, pp. 361–370, 2014.
10. Ogrodzki, J.: *Circuit simulation methods and algorithms*. CRC Press, Boca Raton, FL, USA, 1994.
11. Petzold, L., Li, S.T., Cao, Y., Serban, R.: Sensitivity analysis of differential-algebraic equations and partial differential equations, *Computers and Chemical Engineering*, **30**, 1553–1559, 2006.
12. Tasić, B., Dohmen, J.J., ter Maten, E.J.W., Beelen, T.G.J., Schilders, W.H.A., de Vries, A., van Beurden, M.: Robust DC and efficient time-domain fast fault simulation. *COMPEL Int. Journal for Computation and Mathematics in Electrical and Electronic Engineering*, **33-4**, pp. 1161–1174, 2014.
13. Verhoeven, A.: *Redundancy reduction of IC models by multirate time integration and model order reduction*. PhD-Thesis Eindhoven University of Technology, http://alexandria.tue.nl/extra2/200712281.pdf, 2008.
14. Xiu, D.: *Numerical methods for stochastic computations - A spectral method approach*. Princeton Univ. Press, Princeton, NJ, USA, 2010.