*Elektrotehniški vestnik XX(Y): 1–6, YEAR*
*Electrotechnical Review, Ljubljana, Slovenija*

# Computing the steady-state response of nonlinear circuits by means of the $\epsilon$-algorithm

**Borut Wagner, Árpád Bűrmen, Janez Puhan, Iztok Fajfar and Tadej Tuma**

*University of Ljubljana, Faculty of Electrical Engineering,*
*Tržaška cesta 25, SI-1000 Ljubljana, Slovenia*

*E-mail: borut.wagner@fe.uni-lj.si*

**Abstract.** The paper presents the problem of computing the steady-state response of electronic circuits. The direct approach to obtaining the steady-state response (transient analysis) necessitates a considerable amount of time.
The process of obtaining the steady-state response through transient analysis can be accelerated by using extrapolation methods. The $\epsilon$-algorithm is the most appropriate extrapolation algorithm for implementation in the SPICE OPUS circuit simulator developed at the Faculty of Electrical Engineering of the University of Ljubljana.
A short description of the algorithm is given upon which implementation details are discussed. The algorithm is tested on two circuits. The computation time needed by the direct approach is compared to the time needed by the $\epsilon$-algorithm.
The results show that the $\epsilon$-algorithm is appropriate for the rapid evaluation of the steady-state response of circuits excited by a single periodic signal. Finaly, directions for future research are given.

**Key words:** steady-state response, nonlinear circuits, epsilon algorithm, circuit simulation, SPICE

# Računanje stacionarnega odziva nelinearnih vezij z $\epsilon$-algoritmom

**Povzetek.** V članku je predstavljen problem računanja stacionarnega odziva nelinearnih elektronskih vezij s pomočjo časovne (tranzientne) analize, ki lahko za določena vezja traja zelo dolgo. Tranzientno analizo se da pospešiti s pomočjo ekstrapolacijskih metod. $\epsilon$-algoritem je najprimernejši za vgradnjo v simulator vezij SPICE OPUS, ki ga razvijamo na Fakulteti za elektrotehniko v Ljubljani.
Opisana sta princip delovanja $\epsilon$-algoritma in izvedba analize za določanje stacionarnega stanja na podlagi analize v časovnem prostoru (tranzientne analize).
Algoritem je bil testiran na dveh testnih vezjih. Vezji sta bili simulirani s tranzientno analizo in s pospešeno tranzientno anlizo. Primerjava časov, ki so bili potrebni za izračun stacionarnega odziva, kaže, da je $\epsilon$-algoritem zelo primeren za vezja, ki so vzbujana z enim signalnim virom.
Na koncu članka so podane smernice za nadaljnje raziskave na tem področju.

**Ključne besede:** stacionarni odziv, nelinearna vezja, algoritem epsilon, simulacija vezij, SPICE

## 1 Introduction

Computing the steady-state response of a nonlinear electrical circuit is usually much time and computer power consuming, especially for circuits with time constants much larger than the period of the steady-state response. A direct approach to computing the steady-state response necessitates running a transient analysis until all initial transients die off.

A problem occurs when the period of the steady-state response is much smaller than the largest time constant of the circuit. Such circuits must be simulated for hundreds or even thousands of periods before they reach the steady-state. To obtain sufficient accuracy, a hundred or more time points must be evaluated per period of the response. Therefore, the simulation often takes several million time points before steady-state is reached.

Several techniques exist that speed up the computation of the steady-state response [1, 2, 3, 4, 5]. The main difference between them is the domain in which the analysis is performed (frequency domain, time domain, mixed time-frequency domain).

Due to its simplicity and the fact that it doesn't require any major changes in the simulator, the most appropriate algorithm for implementation in SPICE OPUS [6, 7, 8] is the $\epsilon$-algorithm [3, 9, 10, 11]. Originally the $\epsilon$-algorithm was developed for accelerating the convergence of series.

In the sections that follow, a quick overwiev of

*Received 2. February, 2005*
*Accepted 20. October, 2005*

the problem of obtaining the steady-state response is given followed by an introduction to the $\epsilon$-algorithm. The implementation details of the $\epsilon$-algorithm are discussed with emphasis on the parameters that determine its performance. The algorithm is tested on two circuits and its performance is compared to the performance of the direct approach. In the end, conclusions are given and directions for the future research are discussed.

## 2 Circuit simulation with SPICE

Nonlinear dynamical electrical circuits can be described by a system of ordinary differential equations

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), t) \tag{1}$$

where $\mathbf{x}(t)$ represents branch voltages, node voltages and branch currents. In transient analysis, Eq. (1) is solved starting from the initial value $\mathbf{x}_0(t_0)$. Resulting waveforms are represented by $\mathbf{x}(t)$ for $t_0 \leq t \leq t_n$. For further reference, $\mathbf{x}(t)$ is a column matrix with $m$ components:

$$\mathbf{x}(t) = \left[\, x_1(t), \quad x_2(t), \quad \ldots \quad x_m(t) \,\right]^T. \tag{2}$$

Eq. (1) is numerically integrated and solved using the Newton-Raphson iterative method [6, 12] for times $t_1 \leq t_2 \leq \ldots \leq t_n$. Time step $t_{\Delta i} = t_{i+1} - t_i$ should be taken small enough to avoid numerical errors. Decreasing the time step increases the computation time, especially when the steady-state response of a circuit is sought with the direct approach.

Circuit simulation is a part of every circuit optimization [13] where parameters of the circuit are looked for subject to designer's requirements. During optimization, a circuit is simulated many times, so individual simulations should be as short as possible.

## 3 Steady-state response

The steady-state response of an electronic circuit is obtained after all initial transients disappear. The time required to reach the steady state depends on characteristics of the circuit and the excitation frequency. The simulation time is particularly large when the largest time constant of the circuit is much greater than the period of the highest frequency exciting the circuit.

In practice, the circuit should be simulated more than ten times the largest time constant of the circuit in order to attain a sufficient accuracy of the steady-state response. After the simulation is finished, the accuracy of the computed steady-state response can be checked. The circuit is in the steady state if

$$\begin{gathered} |x_i(t_n) - x_i(t_{n-1})| = 0 \\ t_n = t_0 + t + nT \\ t_{n-1} = t_0 + t + (n-1)T \\ i = 1, 2, \ldots, m \end{gathered} \tag{3}$$

for all $0 \leq t \leq T$, $n$ is large enough ($n \to \infty$) and the period of the steady-state response is $T$. A circuit is simulated for a finite number of periods $T$, so Eq. (3) is not exactly satisfied. We can assume that the steady state has been reached when relative and absoulte tolerance criteria

$$|x_i(t_n) - x_i(t_{n-1})| \leq \delta_a + \max\left[\,|x_i(t_n)|\,, |x_i(t_{n-1})|\,\right] \delta_r \tag{4}$$

for all $i = 1, 2, \ldots, m$, and $n$ large enough are satisfied.

In practice, it is sufficient if $\delta_r$ and $\delta_a$ are less than $10^{-5}$ and $10^{-4}$, respectively, but greater than precision of data representation (e.g. relative and absolute precision of *double* is $10^{-14}$ and $10^{-320}$, respectively).

Relative and absolute criteria (Eq. (4)) can also be used when the steady-state response of a circuit is sought by means of the $\epsilon$-algorithm.

The steady-state response is important in the analysis of power conversion circuits and in evaluation of nonlinear properties of narrow-band circuits excited by a single frequency.

## 4 $\epsilon$-algorithm

Suppose that the circuit has a steady-state response with period $T$. So the sequence

$$\mathbf{x}^{(i)} = \mathbf{x}(t_0 + t + iT), \qquad i = 0, 1, \ldots \tag{5}$$

converges for all $0 \leq t \leq T$.

Now form a two-dimensional array depicted in Fig. 1 where the following equations represent the relationships of individual array entries for some value of $t$.

$$\begin{aligned} \epsilon_{-1}^{(i)} &= \mathbf{0}, & & i = 1, 2, 3, \ldots \\ \epsilon_{0}^{(i)} &= \mathbf{x}^{(i)}, & & i = 0, 1, 2, \ldots \\ \epsilon_{j+1}^{(i)} &= \epsilon_{j-1}^{(i+1)} + (\epsilon_{j}^{(i+1)} - \epsilon_{j}^{(i)})^{-1}, & & i = 0, 1, 2, \ldots \\ & & & j = 0, 1, 2, \ldots \end{aligned} \tag{6}$$

$$
\begin{array}{ccccccc}
 & \epsilon_0^{(0)} & & & & & \\
\epsilon_{-1}^{(1)} & & \epsilon_1^{(0)} & & & & \\
 & \epsilon_0^{(1)} & & \epsilon_2^{(0)} & & & \\
\epsilon_{-1}^{(2)} & & \epsilon_1^{(1)} & & \epsilon_3^{(0)} & & \\
 & \epsilon_0^{(2)} & & \epsilon_2^{(1)} & & \epsilon_4^{(0)} & \\
\epsilon_{-1}^{(3)} & & \epsilon_1^{(2)} & & \epsilon_3^{(1)} & \vdots & \\
 & \epsilon_0^{(3)} & & \epsilon_2^{(2)} & \vdots & & \\
\epsilon_{-1}^{(4)} & & \epsilon_1^{(3)} & \vdots & & & \\
\vdots & \epsilon_0^{(4)} & \vdots & & & & \\
 & \vdots & & & & &
\end{array}
$$

Figure 1. Two-dimension array for the $\epsilon$-algorithm

The procedure described by Eq. (6), is the so-called $\epsilon$-algorithm [3, 9, 10, 11]. If the inverse of the vector in Eq. (6) is taken component-wise

$$\mathbf{v}^{-1} = (v_1^{-1}, v_2^{-1}, \ldots, v_m^{-1}), \tag{7}$$

the scalar $\epsilon$-algorithm is obtained. If however

$$\mathbf{v}^{-1} = \mathbf{v} / \|\mathbf{v}\|_2^2, \tag{8}$$

(6) represents the vector $\epsilon$-algorithm. Here only the vector $\epsilon$-algorithm will be discussed.

By means of the $\epsilon$-algorithm the convergence of the sequence in Eq. (5) can be accelerated [3, 9, 10, 11]. Consecutive sequences $\{\epsilon_j^{(i)}\}_{i=0}^{\infty}$, $j = 2, 4, 6, \ldots$ converge faster than the original sequence $\{\epsilon_0^{(i)}\}_{i=0}^{\infty}$. The greater the value of $j$ the faster the sequence converges.

In one iteration of the $\epsilon$-algorithm a circuit is simulated for $i = 2k$ periods. Then $\epsilon_{2k}^{(0)}$ is obtained using Eq. (6) and represents the initial condition for the next iteration of the $\epsilon$-algorithm.

## 5 SPICE OPUS and the $\epsilon$-algorithm

The $\epsilon$-algorithm was tested in combination with the SPICE OPUS circuit simulator [7]. The circuit is first simulated for several periods. Results of the simulation are then interpreted by an external program which extracts the state of the circuit $\mathbf{x}^{(i)}(t_i)$ at $t_i = t_0 + t + iT$, $i = 0, 1, 2, \ldots, n$ for selected $t$. These values represent $\{\epsilon_0^{(i)}\}_{i=0}^{n}$. Next $\{\epsilon_j^{(i)}\}_{i=0}^{n-j}$, $j = 2, 4, 6, \ldots$ using Eq. (6) is obainted. The last calculated $\epsilon$-value $\epsilon_n^{(0)}$ represents new initial conditions $\mathbf{x}_0(t_0)$ for simulating the circuit in the next iteration of the $\epsilon$-algorithm. The simulation is started at time $t = 0$, so $t_0$ is always 0. The algorithm is depicted in Fig. 2.

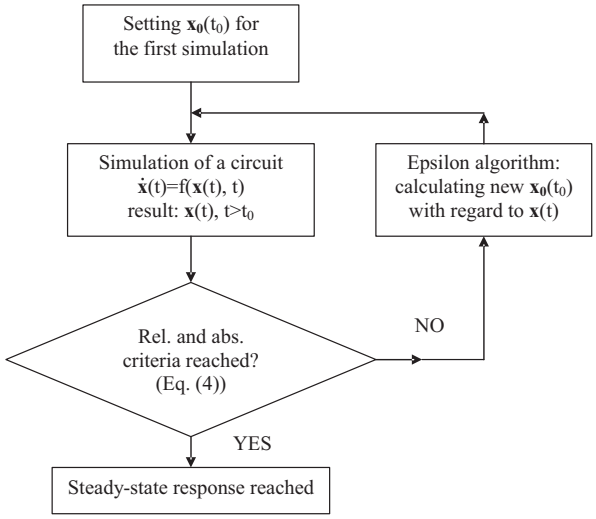


Figure 2. $\epsilon$-algorithm for determing the steady-state response

The speed of convergence depends on the following parameters.

**Parameter** *frequency*
The parameter *frequency* represents the base frequency of the response of the circuit ($T = 1/frequency$). The *frequency* depends on the frequency of the input signal. In case of autonomous circuits (e.g. oscillators), the *frequency* should be determined from the response of the circuit. If the circuit is not in its steady state yet, the *frequency* can change in every iteration of the $\epsilon$-algorithm.

**Parameters** *tdel* **and** *tdelSec*
Parameters *tdel* and *tdelSec* represent $t$, $t \geq 0$. $t = tdel$ in the first iteration of the $\epsilon$-algorithm and $t = tdelSec$ for the remaining iterations. Normally, *tdel* (and *tdelSec*) should be 0. For cases, where the response of the circuit includes some initial rapidly decreasing transients, *tdel* (or/and *tdelSec*) can be set to a non zero value to speed up the convergence.

**Parameters** *NumPointsMin*, *NumPointsMax* **and** *PointsStep*
Values of $\mathbf{x}^{(i)}$ are extracted from the response of the circuit. The number of $\mathbf{x}^{(i)}$ ($i = 0, 1, 2, \ldots, i_{max}$) that enter into the $\epsilon$-algorithm can be set with parameters $NumPointsMin$ and $NumPointsMax$. For the first iteration of the $\epsilon$-algorithm, $i_{max} = NumPointsMin$ is used. In next iterations, $i_{max}$ is increased by *PointsStep* until $i_{max} = NumPointsMax$ is reached.

**Parameter** *PeriodsPerPoint*

If differences between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ are very small, performing the $\epsilon$-algorithm with Eq. (6) could result in a large numerical error. In such cases, the time between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ can be multiplied by $PeriodsPerPoint = 1, 2, 3, 4, \ldots$ . With this multiplication we practically increased the period $T$ in Eq. (5) so the differences between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(i+1)}$ become larger.

**Parameter** *FreqMax*
When a circuit is simulated with SPICE OPUS, an appropriate initial time step $t_\Delta$ has to be chosen to get sufficiently accurate results for the $\epsilon$-algorithm. At least 100 points must be evaluated within one period $T$, so $t_\Delta \leq T/100$. $t_\Delta$ is calculated from $FreqMax$: $t_\Delta = 1/(2 \cdot FreqMax)$.

**Parameters** *EpsStopAbsTol* **and** *EpsStopRelTol*
Iterations of $\epsilon$-algorithm are stopped when the stopping criterion (Eq. (4)) is satisfied ($\delta_a = EpsStopAbsTol$ and $\delta_r = EpsStopRelTol$).

**Parameter** *MaxIters*
If the stopping criterion (Eq. (4)) can't be satisfied (bad choice of parameters), the $\epsilon$-algorithm is stopped when the number of iterations exceeds $MaxIters$.

Determing the above described parameters is very important for fast obtaining the steady-state response. If the choice of parameters is inappropriate, the steady state can't be reached by the algorithm and one ends up with wrong results.

In practice, parameters should be modified until the steady-state response is reached in a few iterations of the $\epsilon$-algorithm. If the circuit is being optimized, parameters need to be determined only once and can remain unchanged for the rest of the optimization. The optimal parameter values are similar for similar circuits. Therefore, the parameter values can be set depending on the type of the circuit and its characteristics.

## 6 Test circuits

Efficiency of the $\epsilon$-algorithm was tested with two test circuits. For both circuits, the time required for calculating the steady-state response by means of $\epsilon$-algorithm is substantiality smaller than the time required by the direct approach.

**Voltage multiplier**
The first circuit is a voltage multiplier, Fig. 3.
At the input of the circuit (between nodes 9 and 10), a sine voltage source with amplitude 311 V and frequency 50 Hz is connected. Output of the circuit is at node 1. If the circuit is simulated for a sufficient amount of time, the response in Fig. 4 is obtained.

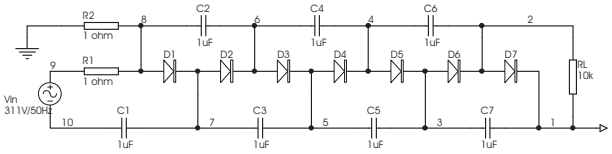


Figure 3. Test circuit 1 (voltage multiplier)

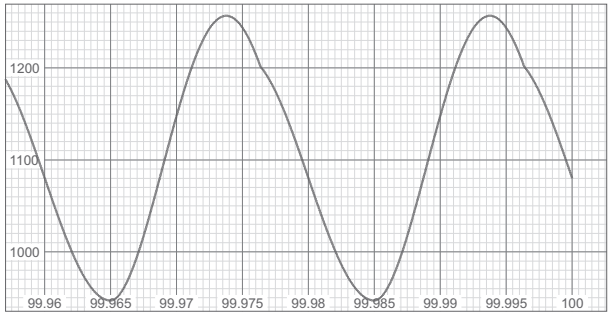


Figure 4. Steady-state response ($v_1(t)$) of the voltage multiplier

**Narrow-band filter**
Circuit in Fig. 5 is a narrow-band filter with quality $Q = 100$ and resonant frequency 1 MHz.

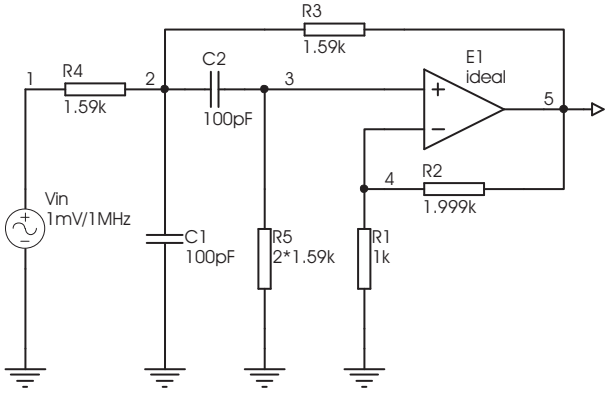


Figure 5. Test circuit 2 (narrow-band filter)

The input of the circuit is at node 1 and the output at node 5. If a sine voltage source with the amplitude 1 mV and frequency 1 MHz is connected to the input, the steady-state response in Fig. 6 is obtained.

## 7 Comparison of the $\epsilon$-algorithm with the direct approach

In this section, computation times needed for obtaining the steady-state response for the direct approach

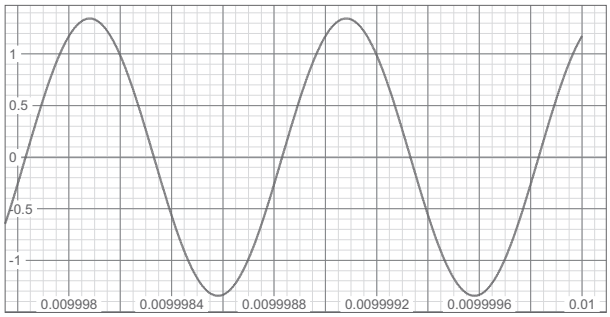


Figure 6. Steady-state response ($v_5(t)$) of the narrow-band filter

and the $\epsilon$-algorithm are compared. Results are shown in Table 1.

Transient analysis (direct approach)
The circuits were simulated from time $t = 0$ to $t = t_{SS}$ with initial time step $t_\Delta$. The number of periods simulated was $N_{TRAN}$.

$\epsilon$-algorithm (accelerated transient analysis)
Values for the parameters are listed in Table 1. Using these parameter values, the steady-state response was reached after $I_{ITER}$ iterations of the $\epsilon$-algorithm. The total number of periods simulated in the $\epsilon$-algorithm was $N_{EPS}$.

| Parameter value | Voltage multiplier | Narrow-band filter |
|---|---|---|
| $t_\Delta$ | 10 $\mu s$ | 0.5 $ns$ |
| $t_{SS}$ | 10 $s$ | 10 $ms$ |
| $N_{TRAN}$ | 5,000 | 10,000 |
| tdel/tdelSec | 2 $ms$ / 0 $ms$ | 1 $\mu s$ / 1 $\mu s$ |
| MaxFreq | 50 $kHz$ | 1 $GHz$ |
| NumPointMin | 4 | 4 |
| NumPointMax | 4 | 16 |
| PointStep | 2 | 2 |
| PeriodsPerPoint | 1 | 1 |
| EpsStopAbsTol | $5 \cdot 10^{-5}$ | $10^{-7}$ |
| EpsStopRelTol | $5 \cdot 10^{-6}$ | $10^{-7}$ |
| $I_{ITER}$ | 9 | 2 |
| $N_{EPS}$ | 36.1 | 12 |
| $C_{acc}$ | 139 | 833 |

Table 1. Comparison of the $\epsilon$-algorithm and the direct approach

By computing the steady-state response, the direct approach (transient analysis) was accelerated by factor $C_{acc} = N_{TRAN}/N_{EPS}$. The computation time required by the $\epsilon$-algorithm itself was not taken into consideration for being neglectable.

## 8 Application of the $\epsilon$-algorithm

The $\epsilon$-algorithm can be used for RF circuits when the steady-state response is needed. If the circuit is being optimized (many simulations of the same circuit in an optimization loop), the faster computation of the steady-state response considerably shortens the optimization process.
The $\epsilon$-algorithm can also be applied to other areas of electrical engineering, e.g. telecommunications [14].

## 9 Conclusion

The $\epsilon$-algorithm described in this paper proved successful in the computing of the steady-state response of electronic circuits. For the two test circuits, the steady-state response was obtained in more than 100 times shorter time than with the direct approach. This achievement is very important when a circuit has to be simulated many times, e.g. in parametric optimization of a circuit. The main disadvantage of using the $\epsilon$-algorithm is that some properties of the circuit must be known in advance in order to set the parameters of the $\epsilon$-algorithm. When the parameters are set, they can remain unchanged for the remaining steady-state simulations of the circuit.

## 10 Future work

Future research will focus on testing the $\epsilon$-algorithm with more circuits.
Automatic parameter tuning in the $\epsilon$-algorithm would help the user when characteristics of a circuit are unknown. Automatic parameter tuning will help determining parameters through several runs of the $\epsilon$-algorithm with different values of parameters.
The $\epsilon$-algorithm will be implemented into the SPICE OPUS circuit simulator [7] as a new analysis for computing the steady-state response of circuits.

Other steady-state evaluation techniques are also being considered [1, 2, 4, 5]. Currently, they are not quite appropriate for implementation in SPICE OPUS since some radical changes and additions are needed.
Once different methods for computing the steady-state response are implementated, a global steady-state analysis will be included in SPICE OPUS that will switch between them automatically.

## 11 References


[1] K. S. Kundert, J. K. White, A. Sangiovanni-Vincentelli *"Steady-state methods for simulation analog and microwave circuits"*, Kluwer Academic Publishers, 1990.

[2] K. S. Kundert, G. B. Sorkin, A. Sangiovanni- Vincentelli, "Applying Harmonic Balance to Almost-Periodic Circuits", *IEEE Transactios on Microwave Theory and Techniques*, vol. 36, no. 2, pp. 366-378, February 1988.

[3] S. Skelboe, "Computation of the Periodic Steady-State Response of Nonlinear Networks by Extrapolation Methods", *IEEE Transactions on Circuits and Systems*, vol. CAS-27, no. 3, pp. 161-175, March 1980.

[4] N. Soveiko, M. S. Nakhla, "Steady-State Analysis of Multitone Nonlinear Circuits in Wavelet Domain", *"IEEE Transactios on Microwave Theory and Techniques"*, vol. 52, no. 3, pp. 785-797, March 2004.

[5] Janne Roos, *"Frequency-domain analysis of nonlinear circuits using Chebyshev polynomials"*, Master's Thesis, Helsinki University of Technology, February 22, 1994.

[6] T. Quarles, A. R. Newton, D. O. Pederson, A. Sangiovanni-Vincentelli, *"SPICE3 Version 3f4 User's Manual"*, University of California, Berkeley, California, 1989.

[7] SPICE OPUS circuit simulator homepage:
URL: http://www.fe.uni-lj.si/spice/
Faculty of Electrical Engineering, Electronic Design Automation Laboratory:
URL: http://www.fe.uni-lj.si/edalab/ .

[8] J. Puhan, T. Tuma, I. Fajfar, "SPICE for Windows 95/98/NT", *Electrotechnical Review, vol. 65, no. 5, pp. 267-271, Ljubljana, Slovenia, 1998.*

[9] J. A. Steele, A. T. Dolovich, "Toward the kernel of the vector epsilon algorithm", *International Journal for Numerical Methods in Engineering*, vol. 48, no. 5, pp. 721-730, June 2000.

[10] X. Gourdon, P. Sebah, *Convergence acceleration of series*, January 10, 2002,
URL: http://numbers.computation.free.fr/Constants/Miscellaneous/seriesacceleration.html .

[11] A. Sidi, W. F. Ford, D. A. Smith, "Acceleration of Convergence of Vector Sequences", *SIAM Journal on Numerical Analysis* vol. 23, no. 1, pp. 178-196, February 1986.

[12] G. A Korn, T. M. Korn, *"Mathematical handbook for scientists and engineers"*, McGraw-Hill Book Company, 1961.

[13] J. Puhan, T. Tuma, "Optimization of analog circuits with SPICE 3f4", *Proceedings of the ECCTD'97*, vol. 1, pp. 177-180, 1997.

[14] S. Tomažič, "On optimality of quadrature amplitude modulation", *Proceedings of the International Conference on Advances in the Internet, Processing, Systems, and Interdisciplinary Research*, Sveti Stefan, Montenegro, IPSI, 2004, October 2-9, 2004.



**Borut Wagner** received his B. Sc. degree in Electrical Engineering in 2003 from the Faculty of Electrical Engineering of the University of Ljubljana, Slovenia, where he is employed as a member of the national young researcher scheme. He is currently working towards a Ph. D. degree at the Electronic Design Automation Laboratory at the same faculty. His research interests include steady-state analyses of nonlinear electrical circuits, circuit simulation and circuit optimization.

**Árpád Bűrmen** received his B. Sc. and Ph. D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 1999 and 2003, respectively. From 1999-2002 he worked as a junior researcher at the Faculty of Electrical Engineering. Since 2002 he has been a Teaching Assistant at the same faculty. His research interests include analog and mixed-mode simulation of circuits and systems, automated design of analog circuits, optimization methods and their convergence and applications, and parallel distributed algorithms.

**Janez Puhan** received his B. Sc., M. Sc. and Ph. D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, Slovenia, in 1993, 1998 and 2000, respectively. Since 1996 he has been with the same faculty where he is an Assistant Professor. His research interests include computer-aided design of analog circuits, optimisation methods, and computer-aided circuit analysis.

**Iztok Fajfar** received his B. Sc., M. Sc. and Ph. D. degrees in electrical engineering from the Faculty of Electrical Engineering, University of Ljubljana, in 1991, 1994 and 1997, respectively. Since 1992 he has been with the same faculty where he is currently an Associate Professor. He teaches several introductory and advanced courses in computer programming. His research interests include design and optimisation of electronic circuits.

**Tadej Tuma** received his B. Sc, M. Sc. and Ph. D. degrees from University of Ljubljana, Faculty of Electrical Engineering, in 1988, 1991, and 1995, respectively. He is an Associate Professor at the same faculty where he teaches four undergraduate and three postgraduate courses. His research interests are mainly in the field of computer-aided circuit analysis and design.


## 12 Acknowledgment


The research has been supported by the Ministry of Higher Education, Science and Technology of Republic of the Slovenia within programme P2-0246 - Algorithms and optimization methods in telecommunications.