Analogue and Digital Mixed Parallel Simulation

Heiko Fricke

Doktorvater:
Prof Dr. -Ing. Dr. h.c. D. A. Mlynski
University of Karlsruhe (T.H.)
Germany

Koreferent:
Dr. Z. Kuznicki, Maitre de conférence
Université Louis Pasteur
Strasbourg
France

Internal report and preliminary notes on
standard circuit description and simulation techniques.

Suggestions, hints to errors and the like are always welcome.

Version 02/20.07.95

Chapter 1

Different ways of describing circuits

1.1 Nodal Approach (NA)

$$\mathbf{YV} = \mathbf{J} \tag{1.1}$$

$\mathbf{Y}$ = admittance matrix;
$\mathbf{V}$ = node voltages (except GND), state variables;
$\mathbf{J}$ = currents; Built from KCL

Problem: No possibility to describe current dependent elements and voltage sources.

1.2 Modified Nodal Approach (MNA)

$$\begin{bmatrix} \mathbf{Y_R B} \\ \mathbf{CD} \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{J} \\ \mathbf{F} \end{bmatrix} \tag{1.2}$$

$\mathbf{Y_R}$ = reduced matrix as above, reduced by current controlling elements and voltage sources.
$\mathbf{B}$ = partial derivatives of the KC equations, with resp. to additional current variables
$\mathbf{C},\mathbf{D}$ = branch constitutive relations, differentiated w.r.t. unknown vector
$\mathbf{J}$ = currents, s. NA
$\mathbf{F}$ = exitations from previous timesteps corresponding to capacitors and inductors, represented in terms of implicit integration scheme. ex.:

$$i_c(t) = C\frac{dv_c}{dt} \Longrightarrow i_c(t_n) = \frac{C}{h}[v_c(t_n) - v_{cp}] \tag{1.3}$$

$v_{cp} = v_c$, previous time step; becomes part of $\mathbf{F}$
Nonlinearities must be linearized around some operating point before introducing them into the equation systems.
Pivoting schemes exist do exploit sparsity to the highest degree possible.[**?**]

1.3 Tableau Approach (TA)

TA makes a very sparse tableau of the circuit. All node voltages, branch voltages, branch currents, designable parameters, fluxes and charges become state variables.[?] TA does not seem to be very much used nowadays.

1.4 General Nonlinear Differential Equation System (NDE)

Idea: as in MNA $\mathbf{v}$ is the vector of node voltages and to be determined currents.

$$\mathbf{F}(\mathbf{v}(t), \dot{\mathbf{v}}(t), \mathbf{p}(t)) = \mathbf{0} \tag{1.4}$$

$\mathbf{p}$ is the vector of inputs.
A more common way to formulate equation systems in the circuit simulation context is:

$$\mathbf{i}(\mathbf{v}, t) + \dot{\mathbf{q}}(\mathbf{v}, t) = \mathbf{0} \tag{1.5}$$

Here the vector $\mathbf{v}$ represents the state variables, inputs included.
([?], [?] for explanation of $\mathbf{i}, \dot{\mathbf{q}}$)

1.5 SPICE description

In SPICE a different approach is chosen:

$$\mathbf{i}(\mathbf{v}, t) + \mathbf{C}(\mathbf{v}, t)\dot{\mathbf{v}}(t) = \mathbf{0} \tag{1.6}$$

Because of this approach in SPICE every ciruit node must be connected to ground with a capacitor. The capacitor may be of no significance to the circuit but it has to be there. This description has some draw backs as accuracy is concerned. For further detail check: [?]

Chapter 2

Replacing Differentiation with Difference Expressions

Usuallly the time derivatives in a NDE system have to be replaced by some more or less accurate difference equivalent in order to compute values for the state variables at specific timepoints. Thus nonlinear algebraic equation (NAE) systems are produced. The following methods are demonstrated using a simple one dimensional example:

$$f(v(t)) + \dot{v}(t) = 0 \tag{2.1}$$

In the more general case of vector functions the same ideas hold in conjunction with Gauss-Jacobi or Gauss-Seidel iteration schemes (described further below). Implicit methods (explained below) are often referred to as Backward Differentiation Formulae (BDF).

2.1 Explicit Euler (EE) Method

EE is the most intuitive way to replace the time derivative of v.

$$\dot{v}(t) = \frac{\Delta v}{\Delta t}$$

$$\Delta v = v_{n+1} - v_n$$

$$\Delta t = h_{n+1}$$

$$v_{n+1} = v_n - h_{n+1} f(v_n) \tag{2.2}$$

Explicit means that the new to be computed v_{n+1} depends only on states and inputs at the previous time point. The problem with EE is that the time step has to remain small enough to guarantee numeric stability of the method. The complex poles of the considered system have to be in a circle around $-1/h_{n+1}$ of radius $1/h_{n+1}$. This is a severe limitation for stiff systems.

2.2 Implicit Euler (IE) Method

In IE the problem of stability is solved by making the new to be computed v_{n+1} dependent on itself, i.e. it shows up on both sides of the equation:

$$v_{n+1} = v_n - h_{n+1} f(v_{n+1}) \tag{2.3}$$

IE is stable for all poles outside the circle around $-1/h_{n+1}$ of radius $1/h_{n+1}$.

2.3 Trapezoidal Rule (TR)

In TR the following mixture of past and present states on the right-hand-side is used:

$$v_{n+1} = v_n - \frac{1}{2} h_{n+1} [f(v_n) + f(v_{n+1})] \tag{2.4}$$

TR is stable for the complete left half plane.

2.4 Gear

In Gear - method steps reaching further into the history are used to calculate the new value of v. Additionally weighting factors are introduced:

$$v_{n+1} = \sum_{r=0}^{k} \alpha_r v_{n-r} - h_{n+1} \beta f(v_{n+1}) \tag{2.5}$$

For $k = 1$ the method becomes the two-step second order formula. It does not make much sense to use more back values of v as no A-stable formula (stable for poles in the left half plane) is ever of order greater than two.([?])

2.5 General Formula

Most of the above formulae can be represented by the following general formula:

$$\sum_{i=0}^{k} \alpha_i v_{n+1-i} = -h_{n+1} \sum_{i=0}^{l} \beta_i f(v_{n+1-i}) \tag{2.6}$$

Thus, for $k = l = 1$, $\alpha_0 = 1$ $\alpha_1 = -1$, $\beta_0 = 0$ and $\beta_1 = 1$ one get's EE.
To make working integration methods the following conditions must hold:

$$\sum_{i=0}^{k} \alpha_i = -1$$

$$\sum_{i=1}^{k} \alpha_i + \sum_{i=0}^{l} \beta_i = 0$$

assuming that $\alpha_0 = 1$. [**?**]
Using this general formula all the known integration formulae can be used to transfer NDE into NAE.
I have not yet found any program that would use any algorithm using more than 2 previous time steps for the computation of the new state. Possibilities would be: Simpson rule, Romberg inegration, TR for more than two time points . The stability regions of these methods have not been investigated so far.

2.6 Semi-Implicit Methods

Implicit methods may lead to NAE after applying BDF like IE:

$$\mathbf{v}_{n+1} = \mathbf{v}_n - h_{n+1}\mathbf{f}(\mathbf{v}_{n+1}) \tag{2.7}$$

with $\mathbf{f}(\mathbf{v}_{n+1})$ being a nonliner vector function. When linearizing these equations they are usually linearized around the previous timepoint and are then multiplied with a Δv composed of the to be computed and the most recent time step.

$$\mathbf{v}_{n+1} = \mathbf{v}_n - h_{n+1}\left[\mathbf{f}(\mathbf{v})_n + \left.\frac{\delta\mathbf{f}}{\delta\mathbf{v}}\right|_{\mathbf{v}_n}(\mathbf{v}_{n+1} - \mathbf{v}_n)\right] \tag{2.8}$$

Because of this mixture of previous and actual time point on the right hand side these methods are known as semi-implicit methods.
Their region of stability is good in all practical allplications but not the whole left half plane. I have not found any further hints on stability. The method seems to be reasonably robust.

Chapter 3

Linearization Algorithms

3.1 Newton-Raphson (NR)

NR can be used for linearization purposes. The resultant LAE system is then further treated by other means. NR can also be used as a method for the solution of NAE. This approach is usually only chosen in a single variable problem as in multiple variable problems the cost of the matric inversion of the jacobian becomes too dominant.

$$\mathbf{v}_{n+1} = \mathbf{v}_n - \mathbf{J}^{-1}(\mathbf{v}_n)\mathbf{f}(\mathbf{v}_n) \tag{3.1}$$

with

$$\mathbf{J}^{-1}(\mathbf{v}_n) = \left.\frac{\delta \mathbf{f}}{\delta \mathbf{v}}\right|_{\mathbf{v}_n} \tag{3.2}$$

Chapter 4

Solution of the Systems described in 1.)

4.1 Newton-Raphson (NR)

To solve a NAE system using NR check the paragraph above on linearization.

4.2 Secant Method

The secant method gets around computing derivatives. To find the solution to (cf [?]):

$$f(x) = 0 \tag{4.1}$$

two starting values x_0 and x_1 are needed. The new value is computed as follows:

$$x_{n+1} = x_n + h_n \tag{4.2}$$

$$h_n = -f_n \frac{x_n - x_{n-1}}{f_n - f_{n-1}} \tag{4.3}$$

The iteration is ended when f_n and f_{n-1} don't differ too much. The advantage over NR is that calculation per iteration is easier and that no derivatives have to be computed. The draw back is that a suitable choice of the starting values is more crucial than in NR to guarantee convergence.

4.3 Direct Method (DM)

4.3.1 Solution of Linear Algebraic Equation Systems (LAE)

$$\mathbf{Ax} = \mathbf{b} \tag{4.4}$$

Either Gaussian-Elimination (GE) or LU-Factorization.
Programs to parallelize DM have been existing for a long time.

ex: Parallelism inherent in GE when all elements of one row/column have to be multiplied by the same factor ...

4.3.2 SPICE

SPICE executes the following steps:
1) TR to convert the system of NDE into a system of NAE
2) (damped) NR to convert to NAE into LAE
3) solve LAE using GE or LU-decomposition

If SPICE does not converge Gear may be chosen instead of TR. The region of stability for TR ideally is the whole left half plane. Sometimes the boundary becomes fuzzy if too many roundoff and truncation errors occur resulting in poles close to but left of the imaginary axis acting like poles in the right half plane. For further reference check: [?]
Parallelisation: [?] shared memory multiprocessor system DIGITAL 8800/8300; most time consuming part in SPICE simulation run: calculation of solution for LAE system (about 80% for big systems). Parallelisation on 3 levels:
1) construction of the system matrix can be performed with multiple processors writing to common memory (load phase)
2) LU decomposition (poor efficiency for sparse systems)
3) time step calculation; each slave process is assigned a set of storage elements and computes the time step required by its set of energy storage elements independently.

4.4 Relaxation Methods

4.4.1 Linear Relaxation

Relaxation is applied to the system of LAE:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4.5}$$

instead of solving with DM.

4.4.2 Nonlinear Relaxation

Relaxation is applied to the system of NAE, after the system of NDE has been converted to NAE system using (mostly) BDF methods.

4.4.3 Iterated Timing Analysis (ITA)

ITA is a nonlinear relaxation method.

```
repeat
   for t = t_start to t_end
      convert NDE to NAE using BDF
      for i = 1 to n
         solve only one NR iteration for state variable i,
         with all the other state variables unchanged for the considered time point
         using GS iteration (explained below)
      next i
   next t
until all LTE small enough
```

with n =number of node-voltages, LTE = Local Truncation Error

Program: SPLICE
Person: Saleh '83
Literature:[?]
Parallelisation described in [?]. The approach relies on a circuit graph description. Whenever a fanin element of some component is changed the specific subcircuit is scheduled and calculated anew. (very optimistic). This is called selective trace technique. Not very convincing results.

4.4.4 One Step Relaxation (OSR)

OSR is a nonlinear relaxation method.

```
repeat
   for t = t_start to t_end
      for i = 1 to n
      convert NDE to NAE using BDF
         repeat
            solve for state variable i, with all the other state variables
            unchanged for the considered time point
         using GS iteration (explained below)
         until exactly solved
      next i
   next t
until all LTE small enough
```

Program: ELDO of Anacad, VHDeLDO, Mix-VHDL of Mentor Graphics
There are but a few hints what solution algorithms are used in ELDO. In [?] Hennion writes that OSR get's around computing derivatives by using the secant method instead of NR.
[?] states that 'improved' NR is normally used for tightly coupled nodes and OSR is applied on loosely coupled nodes. This does not make much sense as NR could be used within OSR as solution algorithm.

The idea is probably the following: if OSR is used, the algorithm inside will be the secant method.
If OSR does not converge OSR is abolished in favor of some other algorithm.
Also, [?] experienced convergence problems using OSR. They were resolved using the BE option of ELDO. It is not clear if this changes the method of changing from NDE to NAE within OSR only, or if OSR is given up completely going back to DM or some other simulation algorithm. In [?] Hennion claims: 'In ELDO ... OSR and NR ... are mixed .. . ... strongly coupled elements are solved togehter by NR in so-called NR blocks, while strongly forward and weakly backward coupled elements are successively solved by relaxation iteration'
No hint about how strong/weak coupling is determined
Person: Hennion, '84, Rodriquez, '94
Literature: [?], [?],[?]

4.4.5 Waveform Relaxation (WR)

WR is applied to the system of NDE as opposed to ITA and OSR being applied to the system of NAE.

repeat
  for $i = 1$ to n
    for $t = t_{start}$ to t_{end}
      repeat
        solve for state variable <u>waveform</u> i, with all the other state variable <u>waveforms</u> unchanged for the considered <u>time interval</u>
      until <u>exactly</u> solved
    next t
  next i
until <u>all</u> LTE small enough

WR offers inherent parallelism. Problems occur when tightly coupled systems are to be simulated. In [?] a partitioning method for tightly coupled nodes is presented. Tightly coupled subcircuits are interconnected with a resistor network instead of their original link. The two thus decoupled circuits have an overlappimg part in common. To determine the values for the resistors the spectral radius of the submatrix connecting the subcircuits is taken into consideration.
Convergence is slow for circuits that produce oszillating waveforms within the circuit and at the outputs. (Hence these are the prefered examples to show the superiority of OSR).
Window partitioning for the time interval under simulation is introduced to overcome these limitations.
Because of these features WR is extremely suitable for simulation of MOS and other directional circuits that offer a high degree if independance between the circuit nodes.

Bipolar circuits are usually difficult to simulate as coupling changes during the run. A method to group nodes together is described in [?]. The node coupling two circuit elements is replaced by a simple resistor network with the resistors depending on the adjacent elements. Depending on the resistor values nodes are grouped together. Fairly heuristic approach.
Program: RELAX
Person: Leelarasmee, '82, White '84
Literature:[?], [?], [?], [?]

4.4.6 Waveform Relaxation Newton-Raphson (WRN)

repeat
  for $i = 1$ to n
    for $t = t_{start}$ to t_{end}
      solve only one NR iteration for state variable waveforms i,
      with all the other state variables waveforms
      unchanged for the considered time interval
    next t
  next i
until all LTE small enough

Program: SPLAX
Person: Saleh, '90
Literature:[?], [?]

Parallelisation of WRN has been attempted. The method in [?] relies on the fact that the most time consuming part of simulation is LU decomposition and other matrix handling operations. Different processors can parallely load and treat matrices. (shared memory computer). When a new variable get's available the processors can start calculating straight away without having to start the matrix load phase again. There is no circuit partitioning used in this approach.

Chapter 5

Iteration Methods

Whenever iterations are needed one state variable or waveform is computed with the other state variables or waveforms held unchanged. There are two commonly used ways of assigning values to the variables not considered: Gauss-Jacobi and Gauss-Seidel. When the solution of LAE systems is attempted some more algorithms are encountered. They are: Successive Over-Relaxation and Conjugate Gradient Methods and others. [?]

5.1 Gauss-Jacobi (GJ)

In GJ the variables not computed are assigned the value they had in the previous iteration regardless of their having already been computed for the new time point or not.
This is simple and offers easy parallelism. Convergence is slow compared to GS.

5.2 Gauss-Seidel (GS)

In GS the variables that are not being computed actually are devided into two groups. Those that have already been computed for the actual time point and those that have not. The already computed ones get their new values assigned while those not yet dealt with remain unchanged, i.e. keep their value of the previous time step.
This offers less parallelism but converges more quickly in general.

5.3 Successive Over-Relaxation (SOR)

SOR introduces in GJ or GS a weighting factor ω for the variables not calculated. This factor must be between 0 and 2 (and usually is around one) and can be increased or decreased in accordance with the observed speed of convergence. The described approach was obsered in literature and programs about the solution of LAE systems but never in combination with the simulation of NAE or NDE.

5.4 Conjugate Gradient Methods

Already for the solution of LAE systems Conjugate Gradient Methods are mathematically difficult to handle compared to the methods described above. Further more, Conjugate Gradient Methods require considerably more storage space than other methods. Hence, they do not seem to be a good tactic to solve large systems.

Bibliography

[Nich94] Nichols, K.G., Kazmierski, T.J., Zwolinski, M., Brown, A.D.: **Overview of SPICE-like circuit simulation algorithms**, *IEE Proc.-Circuits Devices Syst.*, Vol. 141, No. 4, 1994, pp. 242-250

[Bray72] Brayton, R.K., Gustavson, F.G., Hatchel, G.D.: **A New Efficient Algorithm for Solving Differential-Algebraic Systems Using Implicit Backward Differentiation Formulas**, *Proc. IEEE*, Vol. 60, 1972, pp. 98-108

[Ho75] Ho, C.W., Ruehli, A.E., Brennan, P.A.: **The Modified Nodal Approach to Network Analysis**, *IEEE Trans. Circuits and Systems*, Vol. CAS-22, No. 6, pp. 504-509

[Hach71] Hachtel, G.D., Brayton, R.K., Gustavson, F.G.: **The Sparse Tableau Approach to Network Analysis and Design**, *IEEE Trans. Circuit Theory*, Vol. CT-18, No. 1, 1971, pp. 101-113

[Sand67] Sandberg, I.W., Shichman, H.: **Numerical Integration of Systems of Stiff Nonlinear Differential Equations**, *Bell Syst. Tech. J.*, 1968, pp. 511-517

[Casi94] Casinovi, G., Yang, J.-M.: **Multi-Level Simulation of Large Analog Systems Containing Behavioral Models**, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 13, No. 11, pp.1391-1399

[Whit84] White, J., Sangiovanni-Vincentelli,A.L.: **Relax 2.1: A Waveform Relaxation Based Circuit Simulation Program**, *Proc. Custom Integrated Ciruit Conf. 1984*, 1984

[ElTa93] El Tahawy, H., Rodriguez, D., Garcia-Sabiro, S., Mayol, J.-J.: **VHDeLDO: A New Mixed Mode Simulation**, *Proc. Euro-DAC/ Euro-VHDL 93*, 1993, pp. 546-551

[Witt90] Wittum, G.: **Mehrgitterverfahren**, *Spektrum der Wissenschaft*, Apr. 1990, pp. 78-90

[Grim95] Grimm, C., Oehler, P., Waldschmidt, K.: **Modellierung gemischt analog/digitaler Systeme zur Entwurfsunterstützung**, , pp. 1-2

[Chan88] Chang, M.C., Hajj, I.N.: **iPride: A Parallel Integrated Circuit Simulator Using Direct Method**, *Proc. Int. Conf. Computer-Aided Design (ICCAD-88)*, pp. 304-307

[Cox87] Cox, P., Burch, R., Hocevar, D., Yang, P.: **SUPPLE: Simulatorr Utilizing Parallel Processing and Latency Exploitation**, *Proc. Int. Conf. Compute-Aided Design (ICCAD-87)*, 1987, pp. 368-371

[Bisc86] Bischoff, G., Greenbeg, S.: **CAYENNE: A Parallel Implementation of the Circuit Simulator Spice**, *Proc. Int. Conf. Computer-Aided Design*, 1986, pp. 182-185

[Naka91] Nakano, M., Youichirou, M., Shirakawa, I.: **A Waveform Relaxation Method Applicable to Bipolar Digital Circuits**, , pp. 2276-2279

1991 IEEE Int. Symp. on Circuits and Systems, Singapore 06/11-14/91

[Hung91] Hung, G.G., Gallivan, K., Saleh, R.: **Parallel Circuit Simulation Based on Nonlinear Relaxation Methods**, , pp. 2284-2287

[Song93] Song, H., Divekar, D., Mills, L., Wang, P.: **A Method for Improving the Efficiency of Simulating Large Electronic Circuits**, *1993 IEEE Int. Symp. Circuits and Systems*, 1993, pp. 1631-1634

[Saka80] Sakallah, K., Director, S.W.: **An Activity-Directed Circuit Simulation Algorithm**, *Proc. IEEE Conf. Circuits and Computers*, 1980, pp. 1032-1035

[Trot90] Trotter, J.A., Agrawal, P.: **Circuit Simulation Algorithms on a Distributed Memory Multiprocessor System**, *Proc. Int.Conf. Computer-Aided Design (ICCAD-90)*, 1990, pp. 438-441

[Henn85/1] Hennion, B., Senn, P.: **ELDO: A New Third Generation Circut Simulator Using the One-Step Relaxation Method**, *Proc. Int. Conf. Circuits and Systems (ICCAD-85)*, pp. 1065-1068

[Henn85/2] Hennion, B., Senn, P.: **A New Algorithm for Third Generation Circuit Simulators: The One-Step Relaxation Method**, *22nd Design Automation Conference (ACM IEEE)*, 1985, pp. 137-143

[Vasq85] Vasquez, A.V., Director, S.W., Sakallah K.A.: **PIMO: A VLSI Circuit Partitioner for Simulation Applications**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 1075-1078

[Shima85] Shima, T., Mizuta, C.: **Modified Gauss-Seidel Iterative Scheme for High Speed Circuit Simulation**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 1061-1064

[Deut84] Deutsch, J.T., Newton, A.R.: **A Multiprocessor Implementation of Relaxation-Based Electrical Circuit Simulation**, *Proc. 21st Design Automation Conf.*, 1984, pp. 350-357

[Whit85] White, J., Sangiovanni-Vincentelli, A.L.: **Partitioning Algorithms and Parallel Implementations of Waveform Relaxation Algoritms for Circuit Simulation**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 221-224

[Wang91] Wang, R., Wing, O.: **Waveform Relaxation on Tightly Coupled Systems**, , pp. 2280-2283

[Vlac95] Vlach, J., Wojciechowski, J.M., Opal, A.: **Analysis of Nonlinear Networks with Inconsistent Initial Conditions**, *IEEE Trans. Circuits and Systems*, Vol. 42, No. 4, 1995, pp. 195-200

[Sale90a] Saleh, R,A., White, J.K.: **Accelerating Relaxation Algorithms for Circuit Simulation Using Waveform-Newton and Step-Size Refinement**, *IEEE Trans. Computer-Aided Design*, Vol. 9, No.9, 1990, pp. 951-958

[Gall88] Gallivan, K.A., Koss, P., Lo, S., Saleh, R.A.: **A Comparison of Parallel Relaxation-Based Circuit Simulation Techniques**, *Electro/88 Conference Record*, 1988, pp. 1-6

[Cott90] Cottrell, R.A.: **Event-Driven Behavioural Simulation of Analogue Transfer Functions**, *Proc. Euro-DAC '90*, 1990, pp. 240-243

[Sale89] Saleh, R.A., Newton, R.: **The Exploitation of Latency and Multirate Behaviour Using Nonlinear Relaxation for Circuit Simulation**, *IEEE Trans. on CAD*,Vol. 8, No. 12, 1989, pp. 1286-1298

[John91] Johnson, T.A., Zukowski, D.J.: **Waveform-Relaxatio-Based Circuit Simulation on the Victor V256 Parallel Processor**, *IBM J. Res. Develop.*, Vol. 35, No. 5/6, 1991, pp. 707-720

[Pete93] Peterson, L., Mattison, S.: **Dynamic Partitioning for Concurrent Waveform Relaxation-Based Circuit Simulation**, *1993 IEEE Int. Symp. on Circuits and Computers*, 1993, pp. 1639-1642

[Whit91] White, J., Odeh, F.: **Connecting Waveform-Relaxation Convergence Properties to the A-Stability of Multirate Integration Methods**, *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 10, No. 4, pp. 497-508

[Oden90] Odent P., Claesen, L., De Man, H.: **A Combined Waveform Relaxation - Waveform Relaxation Newton Algorithm for Efficient Parallel Circuit Simulation**, *Proc. Euro-DAC '90*, 1990, pp. 244-248

[Patr90] Patrick, D., Lyden, C.: **An Event-Driven Transient Simulation Algorithm for MOS and Bipolar Circuits**, *Proc. Euro-DAC '90*,1990, pp. 230-234

[Eijn90] Eijndhoven, van J.T.J., Stiphout, van M.T., Buurman, H.W.: **Mutirate Integration in a Direct Simulation Method**, *Proc. Euro-DAC '90*, 1990, pp. 306-309

[Besn91] Besnard, J., Benkoski, J., Hennion, B.: **ELDO-XL: A Software Accelerator for the Analysis of Digital MOS Circuits by an Analog Simulator**, *Proc. EDAC '91*, 1991, pp. 136-141

[Ko85] Ko, F.H., Sangiovanni-Vincentelli, A.: **BLOSSOM: An Algorithm and Architecture for the Solution of Large-Scale Linear System**, *IEEE Int. Conf. on CAD: VLSI in Computers*, 1983, pp. 400-403

[Duml85] Dumulgol, D., Cockx, J., De Man, H., Odnet, P.: **Segmented Waveform Relaxation Algotithms for Large Scale Circuit Simulation**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 1069-1072

[Carl85] Carlin, C.H.: **DC Operating Point Computation for Waveform Relaxation Analysis**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 1073-1074

[Nord85] Nordsieck, A.W., Lin, P.M., Hsieh, H.Y.: **Exploitation of Latency and Near-Latency for Transient Analysis of Large Scale Circuits**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 1079-1082

[Casi85] **Aggregation Methods for the Solution of Large Scale Systems of Linear Algebraic Equations**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 1055-1058

[Leel85] Leelarasmee, E.: **Exact Simulation of Large Scale Circuits by Using Approximating Models**, *Proc. Int. Symp. on Circuits and Systems (ISCAS-85*, 1985, pp. 1059-1060

[Odeh83] Odeh, F., Carlin, C.H.: **Robustness Aspects of an Adaptive Waveform Relaxation Scheme**, *IEEE Int. Conf. on CAD: VLSI in Computers*, 1983, pp. 396-399

[Wyat83] Wyatt, J.L., Zukowski, C., Glasser, L.A., Bassett, P., Penfield, P.: **The Waveform Bounding Approach to Timing Analysis of Digital MOS IC's**, *IEEE Int. Conf. on CAD: VLSI in Computers*, 1983, pp. 392-395,

[Guar83] Guarini, M., Palusinski, O.A.: **Integration of Partitioned Dynamic Systems unsig Waveform-Relaxation and Modified Functional Linearization**, *Computer Simulation, Summer Conference Papers, 1983 JUL Vancouver*, 1983, pp. 72-77

[Kim89] Kim, Y.H., Hwang, S.H., Newton R.: **Electrical-Logic Simulation and It's Application**, *IEEE Trans. on Computer-Aided Design*, Vol. 8, No. 1, pp. 8-22

[Odry86] Odryna, P., Nassif, S.: **The ADEPT Timing Simulation Algorithm**, *VLSI Systems Design*, Mar. 1986, pp. 24-34

[Altm90] Altmaë, M.: **MINT - a VHDL Simulation System**, *Proc. EDAC-'90*, 1990, pp. 102-106

[Saka85] Sakallah, K.A., Director, S.W.: **SAMSON2: An Event Driven VLSI Circuit Simulator**, *IEEE Trans. on CAD*, Vol. CAD 4. No. 4, 1985, pp. 668-684

[Chan93] Chang, M.-C., Chern, J.-H., Yang, P.: **Efficient and Robust Path Tracing Algorithm for DC Convergence Problems**, *1993 IEEE Int. Symp. on Circuits and Systems*, 1993, pp. 1635-1638

[Sale87] Saleh, R.A., Newton, A.R.: **An Event-Driven, Relaxation-Based Multirate Integration Scheme for Circuit Simulation**, *Proc. Int. Symp. Circuits and Systems*, May, 1985, pp. 600-603

[Cell95] Cellier, F.E., Otter, M., Elmqvist, H.: **Bond Graph Modeling of Variable Structure Systems**, *Proceedings ICBGM'95, Second Int. Conf. on Bond Graph Modeling and Simulation, Las Vegas, Nevada, January 15-18*, pp.49-55.

[Elmq95] Elmqvist, H., Otter, M., Cellier, F.E.: **Inline Integration: A New Mixed Symbolic/Numeric Approach for Solving Differential-Algebraic Equation Systems**, *Keynote Address, Proc. ESM'95, European Simulation Multiconference, Prague, Czech Republic, June 5.-8.*, pp. 1-12

[Lin95] Lin, S., Kuh, E.: **SWEC Speeds VLSI Simulation**, *IEEE Circuits and Devices Magazine*, Vol. 11, No. 1, pp. 10-15

[Pelz95] Pelz, G., Bielefeld, J., Zappe, F.-J., Zimmer, G.: **Simulating Micro-Electromechanical Systems**, *IEEE Circuits and Devices Magazine*, Vol. 11, No. 2, pp. 10-13

[Ruan91] Ruan, G: **A Behavioral Model of A/D Converters Using a Mixed-Mode Simulator**, *IEEE J. Solid-State Circuits*, 1991, Vol. 26, No.3, pp.283-290

[Yama90] Yamamura, K., Horiuchi, K.: **A Globally and Quadratically Convergent Algorithm for Solving Nonlinear Resistive Networks**, *IEEE Trans. Computer-Aided Design*, 1990, Vol. 9., No. 5, pp. 487-499

[Chad92] Chadha, R., Visweswariah, C., Chen, C.-F.: **M3 - A Multilevel Mixed-Mode Mixed A/D Simulator**, *IEEE Trans. Computer-Aided Design*, 1992, Vol. 11., No. 5, pp. 575-585

[Airi94] Airiau, R., Bergé, J.-M., Olive, V.: **Circuit synthesis with VHDL**, *Kluwer, Kluwer international series in engineering and computer scienc, 261*, Boston, MA, 1994

[Enge93] Engeln-Müllges, G., Reutter, F.: **Numerik-Algorithmen mit ANSI-C Programmen**, *BI-Wiss.-Verl.*, Mannheim, Leipzig ,Wien, Zürich, 1993

[Sche89] Schendel, U.: **Sparse matrices: numerical aspects with applications for scientists and engineers**, *Horwood: Mathematics and its Applications*, Chichester, 1989

[Bane94] Banerjee, P.: **Parallel Algorithms for VLSI Computer-Aided Design Applications**, *Prentice Hall*, Englewood-Cliffs, NJ, 1994

[Hämm91] Hämmerlin, G., Hoffmann, K.H.: **Numerische Mathematik**, 2. Aufl., *Springer*, Berlin, Heidelberg, New York, 1991

[Tuin88] Tuinenga, P.W.: **SPICE, a Guide to Circuit Simulationt and Analysis with PSPICE**, *Prentice Hall*, Englewood-Cliffs, NJ, 1988

[Arms93] Armstrong, J.R., Gray, F.G.: **Structured Logic Design with VHDL**, *Prentice Hall*, Englewood-Cliffs, NJ, 1993

[Deuf93] Deuflhard, P., Hohmann, A.: **Numerische Mathematik: eine algorithmisch orientierte Einführung**, *de Gruyter*, Berlin, 1993

[Zurm84] Zurmühl, R., Falk, S.: **Matrizen und ihre Anwendungen für Angewandte Mathematiker, Physiker und Ingenieure**, 5. Aufl. *Springer*, Berlin, Heidelberg, 1984

[Pres92] Press, W., Flannery, B.P., Teukolsky, S.A., Vetterling, W.T.: **Numerical recipes in Pascal, the art of scientific Computing**,

[Whit85/1] White, J., Odeh, F., Sangiovanni-Vincentelli, A., Ruehli, A.: **Waveform Relaxation: Theory and Practice**, Memorandum No. UCB/ERL M85/65, Electronics Research Lab., College of Engineering, University of Calif. at Berkeley

[Whit85/2] White, J.: **The Multirate Integration Properties of Waveorm Relaxation with application to circuit simulation and parallel computation**, Dissertation, Memorandum No. UCB/ERL M85/90, Electronics Research Lab., College of Engineering, University of Calif. at Berkeley

[Rodr94] Rodriguez, D.: **Déscription et Simulation Mixte Analogique-Numérique: Analyse de VHDL Analogique, realisation d'un simulateur mixte**, Dissertation, Université Joseph Fourier, Grenoble

[Merm92] Mermet, J.: **VHDL for Simulation, Synthesis and Formal Proofs of Hardware**, *Kluwer, Kluwer international series in engineering and computer science; 183*, Dordrecht, 1992

[Harr91] Harr, R.E.: **Applications of VHDL to circuit design**, *Kluwer*, Boston, MA, 1991

[Stoe94] Stoer, J., Bulirsch, R.: **Numerische Mathematik 1, Eine Einführung-unter Berücksichtigung von Vorlesungen von F. L. Bauer**, 7. Aufl., *Springer*, Berlin, Heidelberg, New York, 1994

[Stoe90] Stoer, J., Bulirsch, R.: **Numerische Mathematik 1, Eine Einführung-unter Berücksichtigung von Vorlesungen von F. L. Bauer**, 3. Aufl., *Springer*, Berlin, Heidelberg, New York, 1990

[Breh92] Brehm, J.: **Parallele lineare Algebra**, *Dt. Univ.-Verl., 1992*, Dissertation Univ. Erlangen, Nürnberg

[Bjoe72] Bjoerck, A., Dahlquist, G.: **Numerische Methoden**, *Oldenbourg*, München, 1972

[Jord76] Jordan-Engeln, G., Reutter, F.: **Formelsammlung zur numerischen Mathematik, Anhang: Fortran-IV Programme**, *BI- Hochschultaschenbücher/ Bibliographisches Institut: 106*, Mannheim, 1976

[Kiel94] Kielkowski, R.M.: **Inside SPICE: overcoming the obstacles of circuit simulation**, *McGraw-Hill*, New York, 1994

[Vlad94] Vladimirescu, A.: **The SPICE Book**, *Wiley*, New York, 1994

[Orte88] Ortega, J.M.: **Introduction to parallel and vector solution of linear systems**, *Plenum Pr.*, New York, 1988

[Scho92] Schoen, J.M.: **Performance and fault modeling with VHDL**, *Prentice Hall*, Englewood Cliffs, NJ, 1992

[Sche79] Scheid, F.: **Numerische Analysis: Theorie u. Anwendung**, *McGraw-Hill*, Düsseldorf, 1979

[Scha92] Schaback,R., Werner, H.: **Numerische Mathematik**, 4. Aufl., *Springer*, Berlin, Heidelberg, 1992

[Geor93] George, A., Gilbert, J.R., Liu, J.W.H.: **Graph theory and sparse matrix computation**, *Springer*, New York, Berlin, Heidelberg, 1993

[Beck77] Becker, J.: **Numerische Mathematik für Ingenieure**, *Teubner*, Stuttgart, 1977

[Pete92] Peterson, L.: **Dynamic Circuit Partitioning for Waveform Relaxation-Based Circuit Simulation** Dissertation, Department of Applied Electronics, Lund Institute of Technology, Lund, Sweden

[Mali94] Malinak, L.: **A/D Simulators: An Expanding Array of Choices**, *Electronic Design, Dec. 5, 1994*, pp. 95-102

[Moye95] Moyer, P.: **Rapport du Projet de Fin d'Etudes, Modelisation d'un Injecteur a l'aide du language VHDL-A**, Ecole Nationale Superieure de Physique de Strasbourg, 1995

[Libs94] Libs, J.-M.: **Architecture Multiprocesseur Dédiée à la Résolution des Equations aux Dérivées Partielles en 3D**, Dissertation, Ecole Centrale de Lyon, 1994

[Sale90b] Saleh, R.A., Newton, A.R.: **Mixed mode simulation**, *Kluwer, (Kluwer international series in engineering and computer science; 98)*, Boston, 1990

[] : , , pp.

[] : , , pp.

[] : , , pp.