Institut für Angewandte Analysis und Stochastik

im Forschungsverbund Berlin e.V.

Numerical solution of hierarchically structured systems of algebraic–differential equations

F. Grund

submitted: 29th October 1993

Institut für Angewandte Analysis
und Stochastik
Mohrenstraße 39
D - 10117 Berlin
Germany
e–mail: grund@iaas–berlin.d400.de

Preprint No. 71
Berlin 1993

1991 Mathematics Subject Classification. 65Y05, 65L05, 65H10, 65F50; Secondary 68Q35.
Key words and phrases. Algebraic–differential equations, systems of structured nonlinear equations, systems of linear sparse equations, parallelization of numerical methods.

Herausgegeben vom
Institut für Angewandte Analysis und Stochastik
Mohrenstraße 39
D — 10117 Berlin

Fax: + 49 30 2004975
e–mail (X.400): c=de;a=d400;p=iaas-berlin;s=preprint
e–mail (Internet): preprint@iaas-berlin.d400.de

Contents

Abstract. We consider hierarchically structured systems of algebraic–differential equations. Numerical methods for their solution are described. Parallel methods are discussed.

1. Introduction

Numerical simulation of electric networks has become a mighty tool in the design of electronic circuits. The arising systems of algebraic–differential equations are very large. A subcircuit of a 4M–bit–dram, which has been simulated by the circuit simulation program MAGNUS [1], included more than 30 000 transistors.
Let F denote a vector–valued mapping

$$F : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n.$$

We consider the initial–value problem for the system of algebraic–differential equations

$$F(x(t), \dot{x}(t), t) = 0 \quad \text{on} \quad t \geq t_0 \tag{1.1}$$

and assume that for each initial vector

$$x(t_0) = x^0 \tag{1.2}$$

this problem has a solution on a finite interval $[t_0, t_e]$.
Our aim is to describe, how the solution can be calculated in an effective way using the hierarchical structure of the system.
In section 2 hierarchically structured systems of algebraic–differential equations are introduced. Methods for their solution are described in section 3 and section 4. Since sequential methods are rather time expensive, parallel methods are considered in section 5. They are discussed in connection with algebraic–differential equations, nonlinear equations, and linear equations. Results are reported for the linear case only.

2. Hierarchically structured systems of algebraic–differential equations

Let us assume that the system of algebraic–differential equations (1.1) has the following structure:

$$F(u_1(t), \dot{u}_1(t), y_1(t), \dot{y}_1(t), \ldots, u_r(t), \dot{u}_r(t), y_r(t), \dot{y}_r(t), w(t), \dot{w}(t), t) = 0 \tag{2.1}$$
$$\text{on} \quad t \geq t_0$$
$$F : \mathbb{R}^{\sigma_1} \times \mathbb{R}^{\sigma_1} \times \mathbb{R}^{\sigma_1} \times \mathbb{R}^{\sigma_1} \times \cdots \times \mathbb{R}^{\sigma_r} \times \mathbb{R}^{\sigma_r} \times \mathbb{R}^{\sigma_r} \times \mathbb{R}^{\sigma_r} \times \mathbb{R}^{\rho} \times \mathbb{R}^{\rho} \times \mathbb{R}$$
$$\to \mathbb{R}^{\sigma_1 + \cdots + \sigma_r + \rho}$$

and

$$H_i(u_i(t), \dot{u}_i(t), y_i(t), \dot{y}_i(t), x_i(t), \dot{x}_i(t), t) = 0 \quad \text{on} \quad t \geq t_0 \tag{2.2}$$
$$H_i : \mathbb{R}^{\sigma_i} \times \mathbb{R}^{\sigma_i} \times \mathbb{R}^{\sigma_i} \times \mathbb{R}^{\sigma_i} \times \mathbb{R}^{\pi_i} \times \mathbb{R}^{\pi_i} \times \mathbb{R} \to \mathbb{R}^{\sigma_i + \pi_i}$$
$$\text{for} \quad i = 1, 2, \ldots, r$$

with the initial conditions

$$
\begin{gathered}
w(t_0) = w^0, \\
u_i(t_0) = u_i^0, \quad y_i(t_0) = y_i^0, \quad x_i(t_0) = x_i^0 \quad . \\
\text{for} \quad i = 1, 2, \ldots, r.
\end{gathered}
\tag{2.3}
$$

The number of equations n is equal to the number of functions. It is

$$n = \rho + \sum_{i=1}^{r} (2\sigma_i + \pi_i).$$

Because the equations are implicit and nonlinear and the number of equations is large we solve (1.1), (1.2), or the structured system (2.1), (2.2), (2.3), respectively, with numerical methods (cf. [2], [3], [6], [11], [12]). The principal steps are by the numerical solution of the systems (2.1) – (2.3):

1. discretization of (2.1) and (2.2) with integration formulas,
2. numerical solution of the nonlinear equations with Newton–methods or with other iteration methods,
3. numerical solution of the linear systems with sparse–matrix–techniques.

The single steps are explained in the following sections.

3. Numerical methods

3.1. Systems of algebraic–differential equations. We consider the system (2.1) and (2.2) at the time $t = t_{n+1}$. Following [4] the time discretization is realized by a backward differentiation formula

$$-h_n \dot{x}_{n+1} = \sum_{i=0}^{k} \alpha_i x_{n+1-i},$$

where $\dot{x}_{n+1}$ is an approximation of $\dot{x}(t)$ at time $t = t_{n+1}$ and $x_n = x(t_n)$ and $h_n = t_{n+1} - t_n$. The discretization provides a structured system of nonlinear equations

$$
\begin{gathered}
F(u_1, y_1, \ldots, u_r, y_r, w) = 0 \\
F : \mathbb{R}^{\sigma_1} \times \mathbb{R}^{\sigma_1} \times \ldots \times \mathbb{R}^{\sigma_r} \times \mathbb{R}^{\sigma_r} \times \mathbb{R}^{\rho} \to \mathbb{R}^{\sigma_1 + \cdots + \sigma_r + \rho}
\end{gathered}
\tag{3.1}
$$

and

$$
\begin{gathered}
H_i(u_i, y_i, x_i) = 0 \\
H_i : \mathbb{R}^{\sigma_i} \times \mathbb{R}^{\sigma_i} \times \mathbb{R}^{\pi_i} \to \mathbb{R}^{\sigma_i + \pi_i} \\
\text{for} \quad i = 1, 2, \ldots, r,
\end{gathered}
\tag{3.2}
$$

respectively, in the form

$$
\widehat{F}(x_1, u_1, \ldots, x_r, u_r, y_1, \ldots, y_r, w) = \begin{bmatrix} H_1(u_1, y_1, x_1) \\ \vdots \\ H_r(u_r, y_r, x_r) \\ F(u_1, y_1, \ldots, u_r, y_r, w) \end{bmatrix} = 0. \tag{3.3}
$$

For the integration of the differential equations we work with control of the step size width and the order of the integration formulas.
Let us use Newton and Gauss Seidel methods for the numerical solution of (3.1) and (3.2) where we regard the structure of the system.

3.2. Systems of nonlinear equations.

Two–level Newton method (cf. [6], [9], [12]). Let us assume that the subsystems (3.2) can be solved for y_i, i.e.

$$y_i = G_i(u_i). \tag{3.4}$$

Then the system (3.1) and (3.2) is equivalent to the system

$$F(u_1, G_1(u_1), \ldots, u_r, G_r(u_r), w) = 0.$$

Let denote $u = (u_1, \ldots, u_r)^T$. The correction of the Newton method on the upper level

$$\begin{bmatrix} u^{(1)} \\ w^{(1)} \end{bmatrix} = \begin{bmatrix} u^{(0)} \\ w^{(0)} \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta w \end{bmatrix} \tag{3.5}$$

is obtained from

$$\left[\frac{\partial F}{\partial u} + \left(\frac{\partial F}{\partial G_1} \cdot \frac{\partial G_1}{\partial u_1}, \ldots, \frac{\partial F}{\partial G_r} \cdot \frac{\partial G_r}{\partial u_r}\right) \frac{\partial F}{\partial w}\right] \begin{bmatrix} \Delta u \\ \Delta w \end{bmatrix} = -F \tag{3.6}$$

and the correction of the Newton method on the lower level

$$\begin{bmatrix} x_i^{(1)} \\ y_i^{(1)} \end{bmatrix} = \begin{bmatrix} x_i^{(0)} \\ y_i^{(0)} \end{bmatrix} + \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix}, \qquad i = 1, 2, \ldots, r \tag{3.7}$$

is obtained from

$$\begin{bmatrix} \frac{\partial H_i}{\partial x_i} & \frac{\partial H_i}{\partial y_i} \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta y_i \end{bmatrix} = -H_i \qquad i = 1, \ldots, r. \tag{3.8}$$

The computation of the Jacobian for the method on the upper level need the Jacobian

$$\frac{\partial G_i}{\partial u_i}$$

which is determined as follows. Let us assume that the subsystems (3.2) can also be solved for x_i, i.e.

$$x_i = G_i^*(u_i).$$

Then we have with (3.4) and (3.2) the equation

$$\begin{bmatrix} \frac{\partial H_i}{\partial x_i} & \frac{\partial H_i}{\partial y_i} \end{bmatrix} \begin{bmatrix} \frac{\partial G_i^*}{\partial u_i} \\ \frac{\partial G_i}{\partial u_i} \end{bmatrix} = -\frac{\partial H_i}{\partial u_i}. \tag{3.9}$$

From the last equation we find the Jacobian. We see that the coefficient matrices of (3.8) and of (3.9) are one another.

An another possibility for the computation of the Jacobian $\frac{\partial G_i}{\partial u_i}$ is given in [12]. There the coefficient matrix of (3.9) is multiplied by the transposed M^T of a matrix M which is determined in such a way that

$$M^T \cdot \left[\frac{\partial H_i}{\partial x_i}\ \frac{\partial H_i}{\partial y_i}\right] = (0, I)$$

where 0 and I denote the null matrix and the unity matrix, respectively.
We write the last equation in the form

$$\left[\frac{\partial H_i}{\partial x_i}\ \frac{\partial H_i}{\partial y_i}\right]^T \cdot M = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

from which

$$\frac{\partial G_i}{\partial u_i} = -M^T \cdot \frac{\partial H_i}{\partial u_i} \tag{3.10}$$

follows. In [12] a convergence theorem is given for the two–level Newton method. The convergence is quadratic if the inner method is stopped as soon as

$$\left\|(\Delta x_i, \Delta y_i)\right\| \leq \left\|(\Delta u, \Delta w)\right\|^2 \tag{3.11}$$

satisfied is.
<u>One–level Newton method</u> (cf. [6])
Let denote

$$z = (x_1, u_1, \ldots, x_r, u_r, y_1, \ldots, y_r, w).$$

Then we obtain with (3.3) the correction of the one–level Newton method

$$z^{(1)} = z^{(0)} + \Delta z \tag{3.12}$$

from

$$\begin{bmatrix} \frac{\partial H_1}{\partial x_1} & \frac{\partial H_1}{\partial u_1} & & & & \frac{\partial H_1}{\partial y_1} & & & \\ & & \ddots & & & & \ddots & & \\ & & & \frac{\partial H_r}{\partial x_r} & \frac{\partial H_r}{\partial u_r} & & & \frac{\partial H_r}{\partial y_r} & \\ & \frac{\partial F}{\partial u_1} & \cdots & & \frac{\partial F}{\partial u_r} & \frac{\partial F}{\partial y_1} & \cdots & \frac{\partial F}{\partial y_r} & \frac{\partial F}{\partial w} \end{bmatrix} \Delta z = -\widehat{F}. \tag{3.13}$$

For the solution of the linear systems (3.13) we use the Gaussian block elimination method. Moreover we write (3.13) in the following form

$$\begin{bmatrix} A_1 & & & B_1 & & & \\ & \ddots & & & \ddots & & \\ & & A_r & & & B_r & \\ C_1 & \ldots & C_r & D_1 & \ldots & D_r & E \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \vdots \\ \Delta z_r \\ \Delta z_{r+1} \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_r \\ b_{r+1} \end{bmatrix} \tag{3.14}$$

The Gaussian block elimination method for (3.14) needs the following steps:

1. Factorization of $A_i = L_i \cdot U_i, \quad i = 1, 2, \ldots, r$

2. Computation of F_i^T with the factorization of 1.

$$\begin{aligned} U_i^T \cdot Z_i &= C_i^T \\ L_i^T \cdot F_i^T &= Z_i \\ i &= 1,2,\dots,r \end{aligned}$$

3. Computation

$$\widehat{D}_i = D_i - F_i B_i$$

and

$$\begin{aligned} \hat{b}_i &= F_i b_i \\ i &= 1,2,\dots,r \end{aligned}$$

4. Solution of the linear system

$$(\widehat{D}_1 \cdots \widehat{D}_r E)\Delta z_{r+1} = b_{r+1} - \sum_{i=1}^{r} \hat{b}_i$$

5. Computation of Δz_i, $i = 1,\dots,r$ with

$$\begin{aligned} L_i u_i &= b_i - B_i \Delta z_{r+1} \\ U_i \Delta z_i &= u_i \\ i &= 1,2,\dots,r \end{aligned}$$

Gauss Seidel method

We consider again (3.3) and apply the block Gauss Seidel method to this system. The iteration method is reads as follows

$$\begin{aligned} H_1(u_1^{(0)}, y_1^{(1)}, x_1^{(1)}) &= 0 \\ \cdots\cdots & \\ H_r(u_r^{(0)}, y_r^{(1)}, x_r^{(1)}) &= 0 \\ F(u_1^{(1)}, y_1^{(1)}, \dots, u_r^{(1)}, y_r^{(1)}, w^{(1)}) &= 0 \end{aligned} \tag{3.15}$$

The iteration method (3.15) means the decoupling in blocks of the nonlinear system (3.3).

The nonlinear systems

$$H_i(u_i^{(0)}, x_i^{(1)}, y_i^{(1)}) = 0, \quad i = 1,2,\dots,r \tag{3.16}$$

are treated in the same way as with the two–level Newton method. We write (3.16) as

$$H_i(\tilde{u}_i, \tilde{x}_i, \tilde{y}_i) = 0,$$

wherever the iteration method is

$$\begin{bmatrix} \tilde{x}_i^{(1)} \\ \tilde{y}_i^{(1)} \end{bmatrix} = \begin{bmatrix} \tilde{x}_i^{(0)} \\ \tilde{y}_i^{(0)} \end{bmatrix} + \begin{bmatrix} \Delta\tilde{x}_i \\ \Delta\tilde{y}_i \end{bmatrix}, \quad i = 1,2,\dots,r, \tag{3.17}$$

and find the Newton correction as the solution of the linear system

$$\begin{bmatrix} \frac{\partial H_i}{\partial \tilde{x}_i} & \frac{\partial H_i}{\partial \tilde{y}_i} \end{bmatrix} \begin{bmatrix} \Delta \tilde{x}_i \\ \Delta \tilde{y}_i \end{bmatrix} = -H_i(\tilde{u}_i, \tilde{x}_i^{(0)}, \tilde{y}_i^{(0)}), \quad i = 1, 2, \ldots, r. \tag{3.18}$$

Now we must solve the last equation of (3.15). The Newton method gives where we write the equation of (3.15) as

$$F(\tilde{u}_1, y_1^{(1)}, \ldots, \tilde{u}_r, y_r^{(1)}, \tilde{w}) = 0,$$

the iteration method

$$\begin{bmatrix} \tilde{u}_1^{(1)} \\ \vdots \\ \tilde{u}_r^{(1)} \\ \tilde{w}^{(1)} \end{bmatrix} = \begin{bmatrix} \tilde{u}_1^{(0)} \\ \vdots \\ \tilde{u}_r^{(0)} \\ \tilde{w}^{(0)} \end{bmatrix} + \begin{bmatrix} \Delta \tilde{u}_1 \\ \vdots \\ \Delta \tilde{u}_r \\ \Delta \tilde{w} \end{bmatrix}$$

with

$$\begin{bmatrix} \frac{\partial F}{\partial \tilde{u}_1} \cdots \frac{\partial F}{\partial \tilde{u}_r} \frac{\partial F}{\partial \tilde{w}} \end{bmatrix} \begin{bmatrix} \Delta \tilde{u}_1 \\ \vdots \\ \Delta \tilde{u}_r \\ \Delta \tilde{w} \end{bmatrix} = -F(\tilde{u}_1^{(0)}, y_1^{(1)}, \ldots, \tilde{u}_r^{(0)}, y_r^{(1)}, \tilde{w}^{(0)}).$$

The Gauss Seidel method can be applied to (3.3) without any difficulty, but the convergence is only linear. The Gauss Seidel method can be advantageously applied to more deeply structured systems in which the subsystems have again such a structure like the total system. This means for the circuit simulation that the subcircuits can be decomposed into subcircuits. In the case of more deeply structured systems the discussed methods can be combined.

4. Linear equations

In the preceding section it has been shown that several numerical methods for systems of nonlinear algebraic–differential equations leads us to linear system

$$\tilde{A}x = b \tag{4.1}$$

of a large number n of equations (and unknowns) with a non–symmetric sparse $n \times n$ matrix $\tilde{A}$. In the applications which we have in mind a lot such systems have to be solved, but they all have a fixed sparse matrix structure in common. After a formalization of the sparse matrix structure the Gauss elimination method will effectively be adapted to this situation.
We solve also the equation (4.1) with

$$\begin{aligned} P\tilde{A}Q &= L \cdot U \\ Ly &= Pb \\ UQ^{-1}x &= y \end{aligned} \tag{4.2}$$

where P and Q are permutation matrices and L and U are lower and upper triangular matrices. The diagonal entries of L are one.

Because the matrices of (4.1) are sparse only the nonzero elements (nne) are stored. We use the following scheme. The nonzero elements of $\tilde{A}$ and their column indices are stored row by row in the vectors A and JA and the indices of the first row elements of $\tilde{A}$ in A in the vector IA. It can be verified that (m equal number of the nne)

$$A \in \mathbb{R}^m, \quad JA \in \mathbb{N}^m \quad \text{and} \quad IA \in \mathbb{N}^{n+1}$$

is and let us set $IA(n+1) = m+1$. Thus the matrix

$$\tilde{A} = \begin{bmatrix} 9 & & & 2 & 1 \\ 1 & & 3 & & 5 \\ & 2 & & 4 & \\ 1 & & 7 & & 8 \\ & 5 & & 7 & 9 \end{bmatrix} \tag{4.3}$$

is stored as follows:

$$\begin{aligned} A &= [9,2,1,\ 1,3,5,\ 2,4,\ 1,7,8,\ 5,7,9]^T \\ JA &= [1,4,5,\ 1,3,5,\ 2,4,\ 1,3,5,\ 2,4,5]^T \\ IA &= [1,4,7,9,12,15]^T \end{aligned}$$

By the computation of the permutation matrices P and Q also the determination of a pivot strategies we must fulfilled several conditions. The strategies must be so that the elimination method is numerical stable and the fill in the matrices is minimal. It is interesting to note that the condition are contradictory. The pivots in each elimination step are computed as follows where we consider only the first step for simplicity.
Let

$$\tilde{A} = (a_{i,j}).$$

Pivot elements can be only such elements which fulfill the so–called β–condition. Let $\mathbb{I}$ denote the set $\{1, 2, \dots, n\}$ and let us denote

$$\hat{a}_j = \max_{i \in \mathbb{I}} |a_{i,j}|, \quad \forall\, j \in \mathbb{I}$$

An nonzero element $a_{i,j}$, $i, j \in \mathbb{I}$ is said to satisfy a β–condition for some $\beta \in [0, 1]$ if

$$\hat{a}_j \cdot \beta \leq |a_{i,j}|.$$

It is obvious that at least one element $a_{i,j}$ in each column j satisfies the β–condition for a given β. For $\beta = 0$ the factorization may be numerically instable. The case $\beta = 1$ is called column pivoting. It is stable, but the fill in can become rather large. Thus we recommend a choice $\beta = 0.01$ or $\beta = 0.001$.
Let r_i be the number of the nne in the i-th row and c_j in the j-th column. Then the costs for each element $a_{i,j} \neq 0$ are defined following Markowitz [10] as

$$(r_i - 1)(c_j - 1).$$

For the determination of a pivot we consider four strategies. The pivot elements are chosen in the set of elements satisfying the β–condition with a fixed β. If several elements fulfill the condition then the element with the largest absolute value is pivot.

Strategies	
1	Pivot is the element with the minimal costs.
2	We determine the row with minimal number of elements by several the first respectively. The pivot is chosen in this columns in such a way that the costs are minimal.
3	The pivot is chosen in the first column j with minimal c_j in such a way that the costs are minimal.
4	The pivot is chosen in all columns j with minimal c_j in such a way that the costs are minimal.

The numerical order for the strategies is

Strategies	Operations
1	$O(nm)$
2	$O(nn)$
3	$O(nn)$
4	$O(nn)$

In the following table the fill–in and the computing time in seconds are given for an example. The example has been a system of 2904 equations with 58 142 nonzero elements which was treated on a VAX 4000–300.

Strategies	Fill in	Time
1	10 100	533,56
2	16 246	24,08
3	18 870	9,12
4	16 738	26,71

It is obvious that by the numerical solution of structured systems of algebraic-differential equations we must solve many linear systems with the same nonzero pattern. We can generate a machine program [7] for the solution of the linear equations. The execution of the machine program gives their solution with a special Gaussian elimination method. An other possibility is the generation of a pseudo–code. The interpretation of the pseudo–code gives the solution of the systems. The first method is very fast, but depends upon the machine code. The second method can be formulated independent of a computer. A general description will be given elsewhere. Here we describe the method for an example.

Let $\tilde{A}$ be the matrix (4.3). Determining the permutation matrices we obtain

$$P\tilde{A}Q = \begin{bmatrix} 2 & 4 & & & \\ 5 & 7 & & & 9 \\ & 2 & 9 & & 1 \\ & & 1 & 7 & 8 \\ & & 1 & 3 & 5 \end{bmatrix} \tag{4.4}$$

The matrix $P\tilde{A}Q$ is stored in the vector A in the following way where $\textcircled{i}$ denoted the i–the element in A.

$$\begin{bmatrix} \textcircled{7} & \textcircled{8} & & & \\ \textcircled{12} & \textcircled{13} & & & \textcircled{14} \\ & \textcircled{2} & \textcircled{1} & & \textcircled{3} \\ & & \textcircled{9} & \textcircled{10} & \textcircled{11} \\ & & \textcircled{4} & \textcircled{5} & \textcircled{6} \end{bmatrix}$$

The following operations are necessary to factorize $\tilde{A}$:

$$\begin{aligned} \mathrm{A}(12) &= \mathrm{A}(12)/\mathrm{A}(7) \\ \mathrm{A}(13) &= \mathrm{A}(13)\text{–}\mathrm{A}(12)*\mathrm{A}(8) \\ \mathrm{A}(2) &= \mathrm{A}(2)/\mathrm{A}(13) \\ \mathrm{A}(3) &= \mathrm{A}(3)\text{–}\mathrm{A}(2)*\mathrm{A}(14) \\ \mathrm{A}(9) &= \mathrm{A}(9)/\mathrm{A}(1) \\ \mathrm{A}(11) &= \mathrm{A}(11)\text{–}\mathrm{A}(9)*\mathrm{A}(3) \\ \mathrm{A}(4) &= \mathrm{A}(4)/\mathrm{A}(1) \\ \mathrm{A}(5) &= \mathrm{A}(5)/\mathrm{A}(10) \\ \mathrm{A}(6) &= \mathrm{A}(6)\text{–}\mathrm{A}(4)*\mathrm{A}(3)\text{–}\mathrm{A}(5)*\mathrm{A}(11) \end{aligned}$$

In the pseudo–code the arising operations are classified. For each assignment statement the type of the operation (6 are sufficiently), the indices and the length of the scalar product are stored.

In the forward and backward elimination we use a pseudo–code quite analogously. Compared with the generated machine programs the pseudo–code needs less storage, but the interpretation of the pseudo–code takes more time than the execution of the machine program.

The example system of 2904 equations mentioned above was also solved with another pivot strategy. The solution process on a MicroVAX 3600 (double precision FORTRAN; $\beta = 10^{-6}$) provides the following results.

Fill–in	10 232	
Operations factorization	271 133	
Pseudo–Code factorization	2 259 272	Byte
Pseudo–Code fb–elimination	581 840	Byte
Pivot strategy	725,77	Sec.
Generation pseudo–Code	35,74	Sec.
Factorization	4,17	Sec.
FB–elimination	0,48	Sec.
estimation of the condition	$1{,}14 \cdot 10^{12}$	

5. Parallelization of the numerical methods

We sketch the use of parallel or vector computer in the realization of the methods introduced in the preceding sections.

The parallelization can advantageously carried out on each level, i.e. in the systems of

- algebraic–differential equations,
- nonlinear equations,
- linear equations.

5.1. Level algebraic–differential equations. We consider the problem (2.1) and (2.2). The subsystems (2.2) can be solved independently one from another. Let us suppose that just r processors are available. Then the following algorithm can be carried out for the solution in the interval $[t_0, t_1]$:

0. Let $k = 0$, $t_0 < t_1$, and choose an error bound $\epsilon > 0$
1. Choose an approximation

   $$u_i^{(k)}(t), \quad i = 1, \dots, r, \quad t \in [t_0, t_1]$$

2. On each processor $i = 1, \dots, r$ solve a system

   $$H_i(u_i^{(k)}, \dot{u}_1^{(k)}, y_i^{(k)}, \dot{y}_i^{(k)}, x_i^{(k)}, \dot{x}_i^{(k)}, t) = 0 \quad on \quad [t_0, t_1]$$

   with a numerical method. The result is $x_i^{(k)}(t)$ and $y_i^{(k)}(t)$
3. Solve

   $$F(u_1^{(k+1)}, \dot{u}_1^{(k+1)}, y_1^{(k)}, \dot{y}_1^{(k)}, \dots, w^{(k)}, \dot{w}^{(k)}, t) = 0, \qquad on \quad [t_0, t_1];$$

   the result is $u_i^{(k+1)}(t), w^{(k)}(t)$
4. If $\|u_i^{(k+1)}(t) - u_i^{(k)}(t)\| \leq \epsilon \ \ \forall i$, then set new time interval and go to 0 otherwise set $k = k + 1$ and go to 2

In general the step 2 represents a rather large subprogram such that its parallel execution is considerable gain of time.

5.2. Level nonlinear equations. The different methods which has been described in the section 3 can all be parallelized in simular ways. Thus we describe the parallelized algorithm for the two–level Newton method only.
The algorithm for a fixed time step can be formulated as follows

0. Set $l = 0$ and choose an error bound $\eta > 0$
1. Choose an approximation $u_i^{(l)}, w^{(l)}, \quad i = 1, \dots, r$
2. Set $k = 0$
3. Choose an approximation $x_i^{(k)}, y_i^{(k)}, \quad i = 1, \dots, r$
4. On each processor $i = 1, \dots, r$ execute a Newton step for the linear system (3.7). The result is $x_i^{(k+1)}, y_i^{(k+1)}$
5. If

   $$\|(\Delta x_i^{(k+1)}, \Delta y_i^{(k+1)})\| > \|(\Delta u^{(k)}, \Delta w^{(k)})\|^2$$

   then set $k = k + 1$ and go to 4. Otherwise go to 6
6. Compute the Jacobian $\frac{\partial G_i}{\partial u_i}$ with (3.9) or (3.10) respectively.
7. Solve on a processor the linear system (3.6) and compute a Newton step to (3.5) with the result $u_i^{(l+1)}$ and $w^{(l+1)}$.
8. If $\|(\Delta u_i^{(l+1)}, \Delta w^{(l+1)})\| > \eta$, then set $l = l + 1$ and go to 2, else determine new time step.

We see that the steps 4. – 6. can be used parallel on the processors.

On–level Newton method

The computation of Δz by a Newton step to (3.12) can be formed by use of the parallel processor.

The steps 1., 2., 3. and 5. can be applied by the Gaussian block elimination method parallel on r processors while the step 4. must be computed on a processor.

The parallelization of the Gaussian–Seidel method can also be made. We obtain this immediately from (3.17) and (3.18).

5.3. Level linear equations. Here we describe the pseudo–code of section 4 for vector computer. The pseudo–code can also advantageously adapted for parallel computer.

We must find out the statements of the pseudo–code which are independent. We extract these statements and gather up vector statements.

We investigate the independence with the algorithm of Yamamoto and Takahashi [13]. The matrix

$$LU = P\tilde{A}Q$$

from (4.2) can be connected with the matrix

$$M = (m_{i,j}) \qquad m_{i,j} \in \mathbb{N} \cup \{0\}$$

where $m_{i,j}$ denoted the level of the independence. If a matrix element is connected with the level zero then are by the factorization no operations necessary. All matrix elements with the same level can be computed independently. By the factorization we compute at first all matrix element with level 1 then 2 and so on.

The algorithm for the determination of the elements $m_{i,j}$ from the matrix M is given as follows.

```
M = 0
for I = 1,N–1 do
     for all {J: A(J,I )≠ 0 & J>I} do
          M(J,I) = 1+max(M(J,I),M(I,I))
          for all {K: A(I,K) ≠ 0 & K>I} do
               M(J,K) = 1+max(M(J,K),M(J,I),M(I,K))
          end
     end
end
```

For example we consider the matrix (4.3). We obtain

$$M = \begin{bmatrix} 0 & 0 & & & \\ 1 & 2 & & & 0 \\ & 3 & 0 & & 4 \\ & & 1 & 0 & 5 \\ & & 1 & 1 & 6 \end{bmatrix}$$

The factorization is accomplish in the order

Level	Statement

1	A(12) = A(12)/A(7)
	A(9) = A(9)/A(1)
	A(4) = A(4)/A(1)
	A(5) = A(5)/A(10)

2	A(13) = A(13) - A(12) *A(8)
3	A(2) = A(2)/A(13)
4	A(3) = A(3) - A(2) * A(14)
5	A(11) = A(11) - A(5) * A(3)
6	A(6) = A(6) - A(4) * A(3) - A(5) * A(11)

By the use of a vector computer we must find vector statements in the several levels. The following statements have prove a success.

No. vector operation	operation
1	Scalarproduct
2	A(K) = 1/A(K)
3	A(K) = A(K) * A(L)
4	A(K) = (A(I) * A(J) + A(L) * A(M)) * A(M)

The difficulty consists there that the array elements are addressed indirect. For the vector computer exist adequate statements.
For the linear system on page 8 we have fund 171 levels. The results are:

Level	Number statements
0	20 565
1	1 952
2	18 399
3	1 362
4	4 238
5	5 124
⋮	⋮

The largest vector lengths by the vector operations were:

Vector operation	vector length
2	1 952
3	18 399
3	357
3	639

By the linear system were 91,2 % of the operations vectorizable. We have numbered only operations with a vector length 3 and greater.

References

1. J. Borchardt, I. Bremer, F. Grund, D. Horn, M. Uhle, *MAGNUS - Mehrstufige Analyse großer Netzwerke auf der Basis von Untersystemen*, Proceedings, 3. Tagung Schaltkreisentwurf, Dresden, 1989, S. 148–152.
2. L.O. Chua, P.-M. Lin, *Computer-Aided Analysis of Electronic Circuits*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1975.
3. H. Elschner, A. Möschwitzer, A. Reibiger, *Rechnergestützte Analyse in der Elektronik*, Verlag Technik, Berlin, 1977.

4. C.W. Gear, *Numerical initial value problems in ordinary differential equations*, Prentice–Hall, Inc., Englewood Cliffs, New Jersey, 1971.

5. F. Grund, *Modellierung nichtlinearer elektrischer Netzwerke*, Forschungsbericht, 26 S., Institut für Mathematik, Mohrenstraße 39, D - 10117 Berlin, 1981.

6. ______, *Nichtlineare Gleichungssysteme bei elektrischen Netzwerken mit Unternetzwerken*, Forschungsbericht, 24 S., Institut für Mathematik, Mohrenstraße 39, D – 10117 Berlin, 1981.

7. ______, *Numerische Lösung von Gleichungssystemen mit sparsam besetzten Matrizen*, Report R–Math–01/83, Tagung Numerische Lösung von Differentialgleichungen, Matzlow/Garwitz, 1982, S. 33–44.

8. C.W. Ho, A.E. Ruehli, P.C. Brennan, *The modified nodal approach to network analysis*, IEEE Trans. Circuits and Systems **CAS–22** (1975), 504–509.

9. W. Hoyer, J.W. Schmidt, *Newton–type decomposition methods for equations arising in network analysis*, Z. Angew. Math. Mech. **64** (1984), 397–405.

10. H.M. Markowitz, *The elimination form of the inverse and its application to linear programming*, Management Sci. **3** (1957), 255–269.

11. N.B. Rabbat, H.Y. Hsieh, *A latent macromodular approach to large–scale sparse networks*, IEEE Trans. Circuits and Systems **CAS–23** (1976), 745–752.

12. N.B.G. Rabbat, A.L. Sangiovanni-Vincentelli, H.Y. Hsieh, *A multilevel Newton algorithm with macromodelling and latency for the analysis of large–scale nonlinear circuits in the time domain*, IEEE Trans. Circuits and Systems **CAS–26** (1979), 733–741.

13. F. Yamamoto, S. Takahashi, *Vectorized LU decomposition algorithms for large–scale circuit simulation*, IEEE Trans. on Computer–Aided Design **CAD–4** (1985), 232–239.

Veröffentlichungen des Instituts für Angewandte Analysis und Stochastik

Preprints 1992

1. D.A. Dawson, J. Gärtner: Multilevel large deviations.

2. H. Gajewski: On uniqueness of solutions to the drift–diffusion–model of semiconductor devices.

3. J. Fuhrmann: On the convergence of algebraically defined multigrid methods.

4. A. Bovier, J.-M. Ghez: Spectral properties of one–dimensional Schrödinger operators with potentials generated by substitutions.

5. D.A. Dawson, K. Fleischmann: A super–Brownian motion with a single point catalyst.

6. A. Bovier, V. Gayrard: The thermodynamics of the Curie–Weiss model with random couplings.

7. W. Dahmen, S. Prößdorf, R. Schneider: Wavelet approximation methods for pseudodifferential equations I: stability and convergence.

8. A. Rathsfeld: Piecewise polynomial collocation for the double layer potential equation over polyhedral boundaries. Part I: The wedge, Part II: The cube.

9. G. Schmidt: Boundary element discretization of Poincaré–Steklov operators.

10. K. Fleischmann, I. Kaj: Large deviation probability for some rescaled superprocesses.

11. P. Mathé: Random approximation of finite sums.

12. C.J. van Duijn, P. Knabner: Flow and reactive transport in porous media induced by well injection: similarity solution.

13. G.B. Di Masi, E. Platen, W.J. Runggaldier: Hedging of options under discrete observation on assets with stochastic volatility.

14. J. Schmeling, R. Siegmund–Schultze: The singularity spectrum of self-affine fractals with a Bernoulli measure.

15. A. Koshelev: About some coercive inequalities for elementary elliptic and parabolic operators.

16. P.E. Kloeden, E. Platen, H. Schurz: Higher order approximate Markov chain filters.

17. H.M. Dietz, Y. Kutoyants: A minimum–distance estimator for diffusion processes with ergodic properties.

18. I. Schmelzer: Quantization and measurability in gauge theory and gravity.

19. A. Bovier, V. Gayrard: Rigorous results on the thermodynamics of the dilute Hopfield model.

20. K. Gröger: Free energy estimates and asymptotic behaviour of reaction–diffusion processes.

21. E. Platen (ed.): Proceedings of the 1st workshop on stochastic numerics.

22. S. Prößdorf (ed.): International Symposium "Operator Equations and Numerical Analysis" September 28 – October 2, 1992 Gosen (nearby Berlin).

23. K. Fleischmann, A. Greven: Diffusive clustering in an infinite system of hierarchically interacting diffusions.

24. P. Knabner, I. Kögel–Knabner, K.U. Totsche: The modeling of reactive solute transport with sorption to mobile and immobile sorbents.

25. S. Seifarth: The discrete spectrum of the Dirac operators on certain symmetric spaces.

26. J. Schmeling: Hölder continuity of the holonomy maps for hyperbolic basic sets II.

27. P. Mathé: On optimal random nets.

28. W. Wagner: Stochastic systems of particles with weights and approximation of the Boltzmann equation. The Markov process in the spatially homogeneous case.

29. A. Glitzky, K. Gröger, R. Hünlich: Existence and uniqueness results for equations modelling transport of dopants in semiconductors.

30. J. Elschner: The *h-p*–version of spline approximation methods for Mellin convolution equations.

31. R. Schlundt: Iterative Verfahren für lineare Gleichungssysteme mit schwach besetzten Koeffizientenmatrizen.

32. G. Hebermehl: Zur direkten Lösung linearer Gleichungssysteme auf Shared und Distributed Memory Systemen.

33. G.N. Milstein, E. Platen, H. Schurz: Balanced implicit methods for stiff stochastic systems: An introduction and numerical experiments.

34. M.H. Neumann: Pointwise confidence intervals in nonparametric regression with heteroscedastic error structure.

35. M. Nussbaum: Asymptotic equivalence of density estimation and white noise.

Preprints 1993

36. B. Kleemann, A. Rathsfeld: Nyström's method and iterative solvers for the solution of the double layer potential equation over polyhedral boundaries.

37. W. Dahmen, S. Prössdorf, R. Schneider: Wavelet approximation methods for pseudodifferential equations II: matrix compression and fast solution.

38. N. Hofmann, E. Platen, M. Schweizer: Option pricing under incompleteness and stochastic volatility.

39. N. Hofmann: Stability of numerical schemes for stochastic differential equations with multiplicative noise.

40. E. Platen, R. Rebolledo: On bond price dynamics.

41. E. Platen: An approach to bond pricing.

42. E. Platen, R. Rebolledo: Pricing via anticipative stochastic calculus.

43. P.E. Kloeden, E. Platen: Numerical methods for stochastic differential equations.

44. L. Brehmer, A. Liemant, I. Müller: Ladungstransport und Oberflächenpotentialkinetik in ungeordneten dünnen Schichten.

45. A. Bovier, C. Külske: A rigorous renormalization group method for interfaces in random media.

46. G. Bruckner: On the regularization of the ill–posed logarithmic kernel integral equation of the first kind.

47. H. Schurz: Asymptotical mean stability of numerical solutions with multiplicative noise.

48. J.W. Barrett, P. Knabner: Finite element approximation of transport of reactive solutes in porous media. Part I: Error estimates for non–equilibrium adsorption processes.

49. M. Pulvirenti, W. Wagner, M.B. Zavelani Rossi: Convergence of particle schemes for the Boltzmann equation.

50. J. Schmeling: Most β shifts have bad ergodic properties.

51. J. Schmeling: Self normal numbers.

52. D.A. Dawson, K. Fleischmann: Super–Brownian motions in higher dimensions with absolutely continuous measure states.

53. A. Koshelev: Regularity of solutions for some problems of mathematical physics.

54. J. Elschner, I.G. Graham: An optimal order collocation method for first kind boundary integral equations on polygons.

55. R. Schlundt: Iterative Verfahren für lineare Gleichungssysteme auf Distributed Memory Systemen.

56. D.A. Dawson, K. Fleischmann, Y. Li, C. Müller: Singularity of super–Brownian local time at a point catalyst.

57. N. Hofmann, E. Platen: Stability of weak numerical schemes for stochastic differential equations.

58. H.G. Bothe: The Hausdorff dimension of certain attractors.

59. I.P. Ivanova, G.A. Kamenskij: On the smoothness of the solution to a boundary value problem for a differential–difference equation.

60. A. Bovier, V. Gayrard: Rigorous results on the Hopfield model of neural networks.

61. M.H. Neumann: Automatic bandwidth choice and confidence intervals in nonparametric regression.

62. C.J. van Duijn, P. Knabner: Travelling wave behaviour of crystal dissolution in porous media flow.

63. J. Förste: Zur mathematischen Modellierung eines Halbleiterinjektionslasers mit Hilfe der Maxwellschen Gleichungen bei gegebener Stromverteilung.

64. A. Juhl: On the functional equations of dynamical theta functions I.

65. J. Borchardt, I. Bremer: Zur Analyse großer strukturierter chemischer Reaktionssysteme mit Waveform–Iterationsverfahren.

66. G. Albinus, H.–Ch. Kaiser, J. Rehberg: On stationary Schrödinger–Poisson equations.

67. J. Schmeling, R. Winkler: Typical dimension of the graph of certain functions.

68. A.J. Homburg: On the computation of hyperbolic sets and their invariant manifolds.

69. J.W. Barrett, P. Knabner: Finite element approximation of transport of reactive solutes in porous media. Part 2: Error estimates for equilibrium adsorption processes.

70. H. Gajewski, W. Jäger, A. Koshelev: About loss of regularity and "blow up" of solutions for quasilinear parabolic systems.