

Konrad-Zuse-Zentrum für Informationstechnik Berlin



Jörg Ackermann

Michael Wulkow

MACRON – A Program Package for
Macromolecular Kinetics

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Str. 10
D-10711 Berlin-Wilmersdorf
Telefon: 030-89604-0
Telefax: 030-89604-125
e-mail : bibliothek@sc.zib-berlin.de

Verantwortlich: Dr. Klaus André

ISSN 0933-7911

MACRON – A Program Package for Macromolecular Kinetics

JÖRG ACKERMANN MICHAEL WULKOW

*Konrad-Zuse-Zentrum für Informationstechnik Berlin, Heilbronner Straße 10,
W-1000 Berlin 31, Federal Republic of Germany*

December 1990

ABSTRACT

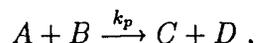
This paper introduces the new program package MACRON for the simulation of macromolecular and standard chemical reactions. Such problems lead to very large systems of ordinary differential equations, which can generally not be solved directly. An efficient approach to these problems is the so-called discrete Galerkin method. The analytical and numerical preparations for this method are performed in MACRON by a chemical compiler. The complete reaction system, standard kinetics as well as macromolecular reactions, can be entered by the user in a chemical formalism. In order to ensure efficiency and reliability, sophisticated numerical routines are built within the package. MACRON can be used without a detailed knowledge of the used numerical methods. Some illustrative examples are added.

CONTENTS

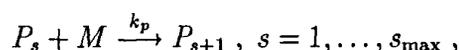
INTRODUCTION	1
1 NUMERICAL CONCEPTS	3
2 USING MACRON	6
2.1 Input File	6
2.2 Simulation	9
3 EXAMPLE: QUASI LIVING RADICAL POLYMERIZATION	15
4 APPENDIX : MACRON-INPUT	23
4.1 General description of the input file CHEMIN	23
4.2 Description of the input blocks	23
4.3 Macromolecular Processes	33
REFERENCES	36

INTRODUCTION

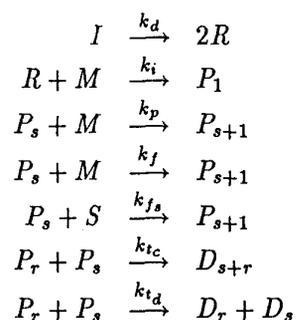
Macromolecular reaction systems generally involve different types of kinetic steps. The first kind are elemental reactions between chemical species A, B, C, D of the form



with a reaction rate coefficient k_p . The simulation of large reaction systems of this kind can be performed by a program package as LARKIN [7]. It is started by the user via a *chemical input*, which is entered in terms of chemical reactions. Macromolecular reactions are described by certain basic steps. For example the notation



models a chain addition of polymer consisting of s_{\max} elemental reactions. The associated set of differential equations is called a *countable (infinite) system of ordinary differential equations* (CODE), because each interesting chain length (degree, index) has to be treated separately. The size of s_{\max} is a priori not known. Thus the computational task is to solve an infinite system in principle. A mixing of the above reaction types arises in many reaction systems of interest. A complete chemical process may contain many reactions and invoke some types of macromolecular species. An example is the following free radical polymerization system (numerically treated in [2]):



Generally, the associated differential equations describe the concentration of the species involved in the elemental (standard) reactions and e.g. the chain length distribution (CLD) of the macromolecules. Such systems turn out to be of very high or even infinite dimension, thus an efficient numerical solution by standard ODE software seems hopeless in general (see Section 3).

In the last years the *discrete Galerkin method* (DEUFLHARD/WULKOW [8]) has been developed as an efficient numerical approach to CODE's. By this method the solution of a CODE is approximated by an error-controlled expansion into orthogonal polynomials of a discrete variable.

The discrete Galerkin method allows the numerical solution of complete macromolecular reaction systems (see e.g. [2] MMA polymerization, and [14] degradation of dextrane).

However, the *realization* of the method requires a good knowledge of analytical and numerical details, which cannot be expected from any user (in specific fields of application). On the other hand, the software standard of the used programs was only on the level of research codes up to now.

The program package MACRON (MACROmolecular reaction kiNetics) presented in this paper is a first step towards a user-friendly implementation. This package combines the discrete Galerkin techniques for the simulation of macromolecular reactions with the software environment of LARKIN. In particular, the *chemical compiler* of LARKIN has been extended. Now, the reaction equations, standard kinetics as well as macromolecular reaction steps, can be entered by the user in a familiar form. Then the necessary preparations of the Galerkin method (called *analytical preprocessing*) are performed. For this purpose a list of typical macromolecular reaction steps (e.g. chain addition, transfer reactions, termination processes) has been implemented and will be continued. The size of a chemical system is restricted by the available computer memory.

The numerical devices of MACRON (time-integration, error estimation) are chosen to give a most reliable standard.

The scope of this first version of MACRON are chemical reactions, which are in some sense connected with the Schulz-Flory distribution. Limitations of the present approach are discussed in Section 2.2. The package will serve as a basis of further developments, in particular in view of ideas, which are realized in the algorithm CODEX [13].

Most of the algorithmical details of the implementation are described elsewhere (see [8], [2], [13]), thus in Section 1 only a brief summary of the basic concepts and the new ingredients of MACRON is given. Section 2 explains in short form the usage of MACRON – including possible difficulties and limitations. To illustrate the efficiency of the code, Section 3 covers a recent example from polymer chemistry. The Appendix includes a detailed input description of MACRON and a list of the macromolecular reaction types implemented at the moment.

1 NUMERICAL CONCEPTS

The numerical heart of MACRON is the *discrete Galerkin method*. Details and examples of this approach can be found in [8], [2] or [13] and will not be repeated here. Only the following items will be sketched below:

- Galerkin approximation
- Analytical preprocessing
- Moving weight function
- Error estimation

which are all related to the discrete Galerkin method itself. Further, some partly new

- Numerical devices

will be mentioned.

Galerkin approximation. Let $u_s(t)$ be the concentration of a molecule of chain length (degree, index) s at time t . The sequence $u_1(t), u_2(t), \dots$, can be considered as a *chain length distribution* (CLD). This CLD is approximated by a truncated expansion $u_s^n(t)$ of certain *orthogonal polynomials* of a discrete variable $l_j(s; \rho)$ multiplied by a parameter dependent weight function $\Psi_\rho(s)$:

$$(1.1) \quad u_s^n(t) = \Psi_\rho(s) \sum_{j=0}^n a_j(t) l_j(s; \rho) .$$

Note, that this is a *global* representation, i.e. the range of s is not restricted. In the present version of MACRON the weight function Ψ_ρ is set to be the *Schulz-Flory distribution*

$$(1.2) \quad \Psi_\rho(s) = (1 - \rho) \rho^{s-1} , \quad 0 < \rho < 1 .$$

The associated polynomials are the *discrete Laguerre polynomials*. Obviously we are interested in obtaining good approximations for small truncation index n . This depends in a crucial way on the parameter ρ . Hence, an adaptive procedure, referred to as *moving weight function concept*, is introduced in [8]. Based on this idea, in MACRON a *single-step* parameter selection can be performed. In this case, after each time step a possibly new value of the parameter is adapted. (For an alternative see the Appendix)

One extension of (1.2) is the modified weight function [13]:

$$(1.3) \quad \Psi_{\rho,\alpha}(s) = (1 - \rho)^{1+\alpha} \rho^{s-1} \binom{s-1+\alpha}{s-1},$$

with an additional parameter $\alpha > -1$. Approximations based on this weight function have some nice features. Depending on ρ and α , a Poisson distribution can be efficiently represented as well as hyperbolas and Schulz-Flory distributions multiplied by arbitrary polynomials. In the present first version of MACRON the capacity of this amelioration has not been used, because the analytical properties of the associated polynomials are more difficult to implement. However, an extended moving weight function concept can be derived and will be used to improve the error estimation (see below). Besides, it is possible to transform a given representation (1.1) to an expansion associated with (1.3) after a simulation.

Analytical preprocessing. For complete reaction systems, many properties of the discrete Laguerre polynomials have to be employed to derive differential equations for the respective expansion coefficients. This sometimes lengthy procedure is called *analytical preprocessing*. The preprocessing can be done step-by-step and is well suited to be implemented in a program. In MACRON a lot of basic kinetic steps are prepared. Especially the treatment of systems with some macromolecular species becomes much easier, because all transformations between the several polynomial systems are automatically performed.

Error estimation. It has been shown in [8] and [13], that the error of approximation (1.1) can be estimated well. The resulting estimate gives valuable hints to discuss obtained results (see Section 2.2).

In MACRON an additional device (called α -check) has been installed to increase the reliability of the method. Optimal parameters $\bar{\rho}$ and $\bar{\alpha}$ for the weight function (1.3) are computed. By comparison of the parameter ρ adaptively chosen for (1.2) and the parameters $\bar{\rho}$ and $\bar{\alpha}$, useful hints can be obtained to estimate the quality of an approximation. Tests showed, that with the standard parameter adaptation for (1.2) from [8], distributions of the extended weight function type up to $\alpha = 5 - 10$ can be efficiently (i.e. with less than 10-20 expansion coefficients) approximated. Whenever for a given CLD the two-parameter adaptation leads to $a > 10$, a warning message will be generated.

Numerical Devices. The computational part of MACRON includes (among others)

-
- numerical integration with the stiff extrapolation code EULSIM [5]. This program has been supplied with an appropriate scaling of the several variables arising in the process and a weighted error norm.
 - control of the initial phase. In the case, that the simulation is started with a zero distribution for a macromolecule, an adaptive iteration procedure to estimate the weight function parameter has been developed. An *initiation time* is computed, for which a stable calculation of the parameter ρ is possible.
 - approximation of given CLD's. The expansion coefficients can be obtained directly by a numerical evaluation. This is useful, whenever a simulation has to be started with an initial (for example measured) distribution. If such a distribution is only given at some mesh points, these values are linearly and quadratically interpolated. Then *discrete summation rules* as developed in [13] are used in form of a *two-level summation*. This allows an error estimation for the appearing sums and makes a control of perturbations of the expansion coefficients possible. A reasonable number of expansion coefficients suited for the actual input is derived. The first statistical moments and mean values are computed. The routine can also be used for *data compression* of measured CLD's.

2 USING MACRON

The structure of the program package MACRON can be schematically described by the following diagram, which roughly prescribes the organization of this section:

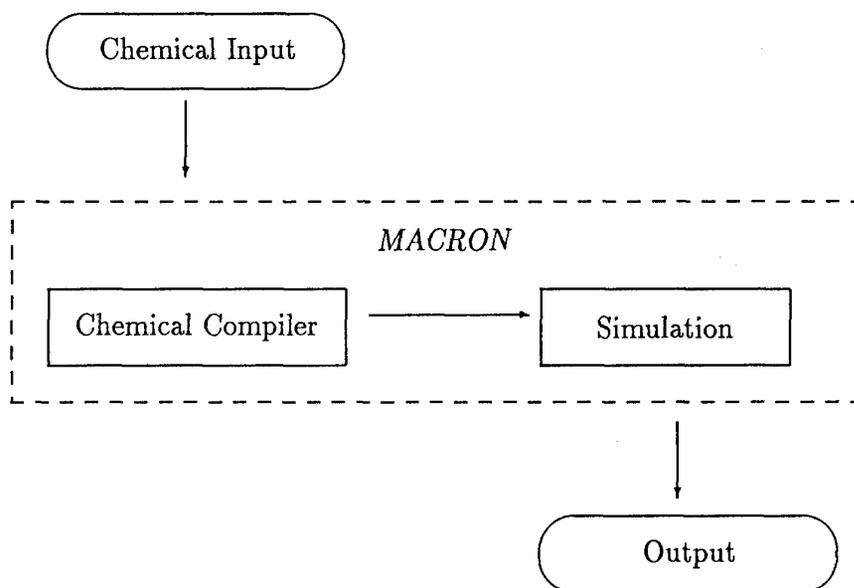


Figure 1: Schematic diagram of MACRON.

In Section 2.1, some hints are given in addition to the syntax description of the input file CHEMIN (see Appendix). CHEMIN contains in particular the chemical reaction system and will be analyzed by the chemical compiler. In case of successful compilation, the user can start the simulation of his model. After a simulation run, the user may modify the chemical input (this is possible without interruption of the program, if a window system is used). Since working with the numerical algorithm requires some experience, in Section 2.2 we try to give an idea of how to proceed with MACRON in view of the approximation of chain length distributions.

Different possibilities are given in order to produce an appropriate output. A choice can be made by the input parameter IPRINT or interactively.

2.1 INPUT FILE

In [2], the discrete Galerkin method is applied to the *free radical polymerization* system shown in the introduction of this paper. A lot of preparations concerning

the analytical preprocessing and the numerical integration are necessary to use the discrete Galerkin method for such a specific reaction system. But the effort pays off. The efficiency of the method has been impressively demonstrated for this model (and similar ones). MACRON performs all preparations automatically for an arbitrary reaction system and we exemplify the MACRON input file for this well discussed model. This should give only an idea of the usage of MACRON, especially if the user is confused by the detailed informations collected in the Appendix.

The input file shown in Table 1 must have the name CHEMIN and contains in general all information to define a reaction system. It is divided in several input blocks, every input block is opened by a specific keyword and ends when a new input block is opened. Detailed information about every keyword can be found in the Appendix.

Here the first input block is opened by the keyword *HEAD. The following lines will be interpreted as text to identify the reaction system. The block is closed after 3 text lines by the keyword *MODEL PARAMETER opening the next block. The 1 in the first input line of this block determines a model parameter (constant temperature). The following blocks *ELEMENTS and *SPECIES may be omitted, because they contain no further input lines. The unit system 3 is chosen in the next block.

The reaction equations in the block *REACTION SYSTEM are most important. The internal system of differential equations will be constructed according to an analysis of standard and macromolecular kinetics. Macromolecular reactions are characterized by involved macromolecular species (here P[N] and D[N]). This is familiar to every chemist. The chemical compiler works only on the level of *chemical* equations. The resulting *differential* equations are not available for the user. This is not as restrictive as it seems to be, actually. The language of chemical reaction kinetics can be used to construct differential equations, which do not result from the analysis of standard kinetics.

For example, in [2] a quasi stationary state approximation (QSSA) for the initiation reaction leads to the contributions

$$(2.1) \quad I' = -k_d I + \dots$$

$$(2.2) \quad M' = -2fk_d I + \dots$$

with the efficiency factor $f = 0.3$ and reaction constant $k_d = 1.5 \cdot 10^{-5}$. The reaction



just produces (2.1), whereas



```

*HEAD
*****
-----      MMA      -----
*****
*MODEL PARAMETER
  1
*ELEMENTS ( EL-NAME - ATOMIC WEIGHT IN G/MOL )
*SPECIES ( SP-NAME - SPECIES COMPOSITION )
*UNIT SYSTEM
  3
*REACTION SYSTEM
I          =>          (1.5D-5)
I          =>  I + M    (-9.0D-6)
I          =>  P[1] + I (9.0D-6)
P[N] + M   =>  P[N+1]  (759.4D0)
P[N] + M   =>  D[N] + P[1] (0.0178D00)
P[N] + S   =>  D[N] + P[1] (0.0331D00)
P[N] + P[M] =>  D[N+M]  (24193872.5D00)
P[N] + P[M] =>  D[N] + D[M] (10368802.5D00)
*GALERKIN PROJECTION
P[N]  5
D[N]  5
*INITIAL CONCENTRATIONS ( 0 ) 0=MOLAR CONC. , 1=MOLE FRACTIONS
M  4.32D0
I  0.01508D0
S  4.91D0
*ACCURACY
  1.D-3
*INTEGRATION BOUNDS
  0.0
  16200.0
*PRINT PARAMETER
  0
*DISTRIBUTION OUTPUT
  1  10000  100

```

Table 1: Input file CHEMIN for free radical polymerization.

leads to (2.2). Both reactions seem senseless to a chemist, but are necessary to overcome the lack of reaction constant k_i in [2] by a QSSA assumption. Such artificial reactions are seldom, but appear necessarily in case of simplifications of a model.

The next input block, *GALERKIN PROJECTION, determines the number of polynomials used for the Galerkin approximation of each macromolecular species. In this case, five polynomials are sufficient to get a projection error less than $7 \cdot 10^{-3}$. More information concerning this point can be found in Section 2.2.

Initial values for the species concentrations M, I, S are given in the input block *INITIAL CONCENTRATIONS. The process time and the accuracy of the numerical integration of the resulting systems of ODE's are regulated in *ACCURACY and *INTEGRATION BOUNDS, respectively. By *PRINT PARAMETER additional information given on separated files can be requested. Finally *DISTRIBUTION OUTPUT determines the range of indices and the increment for the output of distributions.

Note that this is only one specific example of a (short) input. Several possibilities of MACRON are not demonstrated here. We therefore refer to another example treated in Section 3 and to the Appendix.

2.2 SIMULATION

First steps with a new reaction system. When the user begins the work with a complete new reaction system, we recommend to use the minimum allowed number of two expansion coefficients, NPROJ=2, for each macromolecular species first. NPROJ determines the quality and accuracy of the Galerkin approximation, but will not be chosen adaptively by MACRON. A choice of NPROJ=2 will lead to correct results for the first statistical moments of each chain length distribution. The results of such a simulation — like mean values, polydispersities and concentrations — may be helpful for a first rough adjustment of model parameters. If *open* reaction steps occur, NPROJ+ 3 expansion coefficients will be used automatically by the program to ensure this. About the attributes *closed* and *open* the user will be informed in the macromolecular reaction list in Section 4.3.

In the second step, the resulting CLD's can be considered. In this version of MACRON all approximations are based on the Schulz-Flory distribution. Obviously, such a distribution can be very well represented with NPROJ=2 expansion coefficients. The more different an actual CLD is from the Schulz-Flory distribution, the higher NPROJ must be. Actually, the approximation of a wide range of CLD's is possible within technical accuracy ($10^{-1} - 10^{-2}$) and $NPROJ \leq 20$. The maximum number of coefficients is restricted by $NPROJ \leq 98$. We recommend

to increase NPROJ by steps of about five using the error prediction formula described below.

Starting the simulation with an initial distribution. To start the simulation with an initial chain length distribution, the user has several possibilities. If there exists a (for example measured) distribution on a separate file, it can be related to a macromolecular species using the input block *GALERKIN PROJECTION. The program approximates the data by a polynomial expansion used as initial value of a simulation. The truncation index can be prescribed by the user, otherwise it will be chosen by the program. Moreover, some standard distributions can be defined. A skillful user may also enter initial values for every weight function parameter and the associated expansion coefficients separately.

Use of the Error Estimation. The most important tool to discuss the quality of a Galerkin approximation is the error estimate given at the end of each simulation. In the following, we point out some possibilities to work with this device. We use three simple but illustrative test examples and approximate only given distributions (as if they had been obtained by a simulation). Possible limitations and difficulties are emphasized. In 'good' examples, the approximation will often show a similar behavior to Test 1.

Theorem 2.14 in [13] implies, that under some assumptions the (relative) error $\bar{\epsilon}_n$ of a Galerkin approximation u_s^n behaves like

$$\bar{\epsilon}_n \approx C n^\gamma$$

where C and γ are at least locally independent of n . Assume now, that two expansions (of the same distribution) have the errors $\bar{\epsilon}_{n_1}$ and $\bar{\epsilon}_{n_2}$ with $n_1 < n_2$. If the user is interested in reaching a given tolerance, a plausible prediction for $\bar{\epsilon}_{n_3}$ ($n_3 > n_2$) is

$$(2.3) \quad \bar{\epsilon}_{n_3} \approx \left(\frac{(\bar{\epsilon}_{n_2})^{n_3 - n_1}}{(\bar{\epsilon}_{n_1})^{n_3 - n_2}} \right)^{1/(n_2 - n_1)}$$

This can be verified by using the binomial theorem and by omitting $O(1/n)$ terms. For $n_2 = n_1 + 1$, $n_3 = n_2 + 1$ formula (2.3) reduces to

$$\bar{\epsilon}_{n+2} = \frac{\bar{\epsilon}_{n+1}^2}{\bar{\epsilon}_n}$$

In practical examples, the *true* error $\bar{\epsilon}_n$ of an approximation is not available, therefore it is replaced by a well-trying error estimate ϵ_n .

• *Test 1.* In Table 2, estimated errors ϵ_n and true errors $\bar{\epsilon}_n$ are listed, which arise from the approximation of a (typical) distribution

$$u_s = \rho^s \cdot s, \quad \rho = 0.95.$$

Here the moving weight function condition leads to the choice $\bar{\rho} = 2\rho/(1 + \rho)$. For example, from ϵ_5 and ϵ_8 , we can expect with (2.3), that

$$\epsilon_{15} \approx \left(\frac{\epsilon_8^{10}}{\epsilon_5^7} \right)^{1/3} = 2.8 \cdot 10^{-4}$$

— a value which indeed is nearly obtained. In Table 2, the predictions are based on the respective last two error estimations.

n	estimated error ϵ_n	true error $\bar{\epsilon}_n$	prediction of ϵ_n
2	$2.3 \cdot 10^{-1}$	$3.1 \cdot 10^{-1}$	
5	$6.7 \cdot 10^{-2}$	$7.6 \cdot 10^{-2}$	
8	$1.3 \cdot 10^{-2}$	$1.4 \cdot 10^{-2}$	$2.0 \cdot 10^{-2}$
10	$3.9 \cdot 10^{-3}$	$4.0 \cdot 10^{-3}$	$4.4 \cdot 10^{-3}$
15	$1.7 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$	$1.9 \cdot 10^{-4}$
20	$6.7 \cdot 10^{-6}$	$3.1 \cdot 10^{-6}$	$7.4 \cdot 10^{-6}$

Table 2: Behavior of error, Test 1

As a rule of thumb we can say, that whenever the error (estimates) behaves that regular, the user may expect to get reliable approximations. The α -check (based on the extended weight function (1.3)) leads in this case to

$$\bar{\rho} = 0.95 \quad \text{and} \quad \bar{\alpha} = 1,$$

corresponding to the fact, that the distribution u_s can be written as $C \cdot \Psi_{0.95,1}(s)$.

• *Test 2.* We try to approximate a *Poisson*-distribution having a maximum at index $s = 50$:

$$u_s = e^{-\lambda} \frac{\lambda^{s-1}}{(s-1)!}, \quad \lambda = 50$$

Table 3 shows that the true error $\bar{\epsilon}_n$ decreases very slowly, the error estimates are in the right scale, but predictions on the bases of the estimates ϵ_n are not reasonable. There is especially no *monotony* of the ϵ_n . In a real simulation we would not know a priori the form of the true distribution and in order to find

n	estimated error ϵ_n	true error $\bar{\epsilon}_n$	prediction of ϵ_n
2	0.50	0.87	
5	0.16	0.76	
10	0.26	0.72	0.02
15	0.11	0.62	no monotony
20	0.11	0.61	0.04
30	0.08	0.49	no monotony
40	0.10	0.46	0.05

Table 3: Behavior of error, Test 2

the reason for such an error behavior we could use the α -check of MACRON. In this case we would obtain

$$\bar{\rho} = 0.01 \quad , \quad \bar{\alpha} = 5000 \quad .$$

According to Section 2.3 in [13], this result indicates that u_s is a Poisson distribution indeed. Note, that a Poisson distribution reaches its maximum at the index $s = \bar{\rho} \cdot \bar{\alpha}$.

The attempt to approximate u_s with 10 and 20 discrete Laguerre polynomials is shown in Figure 2. Even the approximation with 20 discrete Laguerre polynomials is not very good.

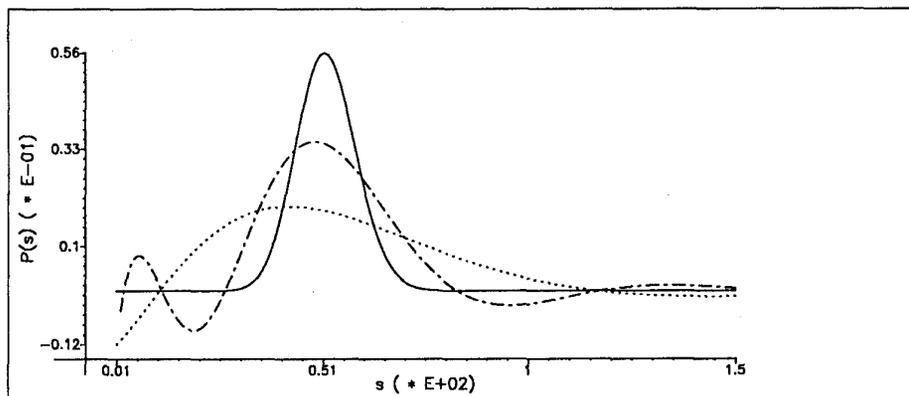
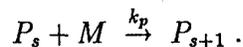


Figure 2: Approximations of a Poisson distribution (—) with with 10 (···) and 20 (---) discrete Laguerre polynomials.

Anyhow, it is possible to compute the statistical moments of u_s exactly. A *post-transformation* of an approximation u_s^n to an expansion in terms of the

modified weight function $\Psi_{\bar{\rho}, \bar{\alpha}}$ (an option of MACRON) will often lead to a more reasonable result in case of closed systems. The user may try this by simulating a pure chain addition process of the form



• *Test 3.* By examining the free radical polymerization process of Table 1, we discovered, that after about 0.02 seconds of process time the chain length distribution of the radicals looks like a step function (with a smooth transition) for a short phase. This effect was detected because of some irregularities of the error estimates. In order to model the situation, we approximate the distribution

$$u_s = \begin{cases} 1, & s \leq 50 \\ 0, & s > 50 \end{cases}$$

and can study the errors and approximations in Table 4 and Figure 3.

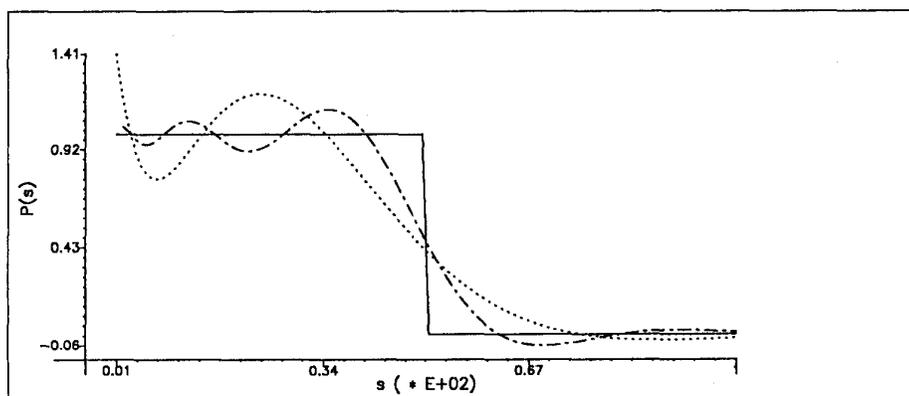


Figure 3: Approximations of a step function with NPROJ=10 (...) and NPROJ=20 (---) polynomials.

It is obvious, that such a CLD cannot be approximated well by an expansion with only few polynomials, but note, that in Table 4 the *true* errors do well agree with the prediction formula (2.3).

The estimate ϵ_n have a similar behavior as in the last example, but the α -check gives no additional information ($\bar{\rho} = 0.88$, $\bar{\alpha} = 2.27$) here. This indicates, that the distribution to be approximated is badly shaped for polynomial approximation and the user may try to increase the truncation index NPROJ (if the machine allows it). Similar effects can arise for some bimodal CLD's and

n	estimated error ϵ_n	true error $\bar{\epsilon}_n$	prediction of $\bar{\epsilon}_n$
2	0.30	0.53	
5	0.07	0.46	
10	0.05	0.38	0.36
15	0.09	0.36	0.31
20	0.09	0.33	0.34
30	0.03	0.30	0.28
40	0.05	0.28	0.27

Table 4: Behavior of error, Test 3

for measurements with a lot of oscillations. However, in this case a smoothing may be sometimes desirable. In Figure 4, a measured CLD from a biological polymerization [1], given only at 76 mesh points, is sampled by 30 polynomials with error $\bar{\epsilon}_{30} = 7 \cdot 10^{-2}$ (α -check: $\alpha = 5.2$). This is remarkable, because the orthogonal projection used for the approximation is perturbed by the range of the measured data (from about $s=2000$ to about $s=60000$), which also has oscillations.

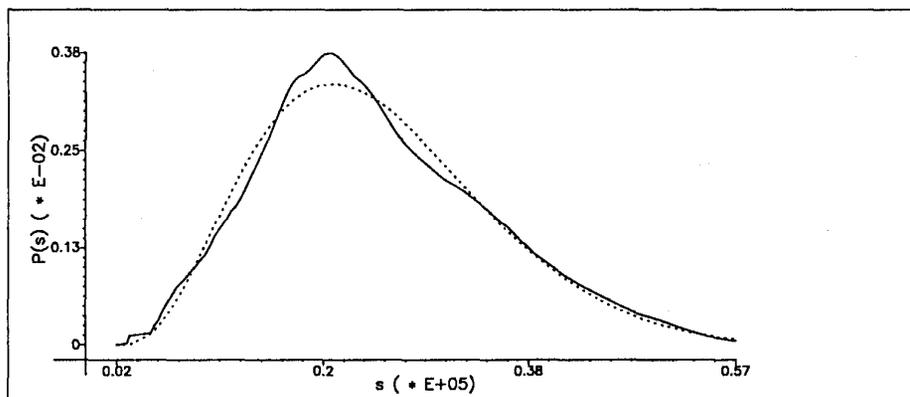


Figure 4: Approximations of measured distribution with NPROJ=30

In brief, whereas statistical moments, mean length, mean weights, dispersion and concentrations can be computed for every reaction system, distributions will not always satisfactorily be obtained by the actual version of MACRON. However, the built-in devices enable the user to decide, how reliable the informations really are.

3 EXAMPLE: QUASI LIVING RADICAL POLYMERIZATION

During the development of MACRON, the package has been tested on realistic problems :

- the workhorse example 'Free Radical Polymerization', which has been treated in [2] with the discrete Galerkin method. As could be seen in that publication, a lot of preparations have been necessary to start with the simulation. Now, the program can be started with the input file shown in Figure 1.
- the formation of soot in flames. This is a difficult task, because a lot of (> 300) chemical reactions have to be considered before the macromolecular steps come into play. The combination of both has been tested with MACRON.
- a recent model for biological polymerization. The model has been simulated using the concepts described in [1]. As a result, first principle considerations and model assumptions could be checked.
- a 'Quasi Living Radical Polymerization' model published in [9]. Despite very extensive computations on supercomputers and several approximations, up to now the simulation of this model was only possible by a direct integration of a large-scale ODE system. Thereby restrictions on the reaction constants and the reaction times had to be accepted. By MACRON the reaction system can be treated very fast on a workstation without these restrictions.

We do not want to feature all examples in detail, since it would be out of the scope of this paper. But we choose the last example to show that an enormous reduction of computing time and an increase of quality can be obtained by MACRON.

Quasi Living Radical Polymerization. We compare an application of MACRON with the results of CH. H. J. JOHNSON et al. in [9].

In [9] a program has been developed to solve directly the complete set of coupled differential equations resulting from an analysis of polymerization kinetics. The program was written to make full use of the speed and power of modern

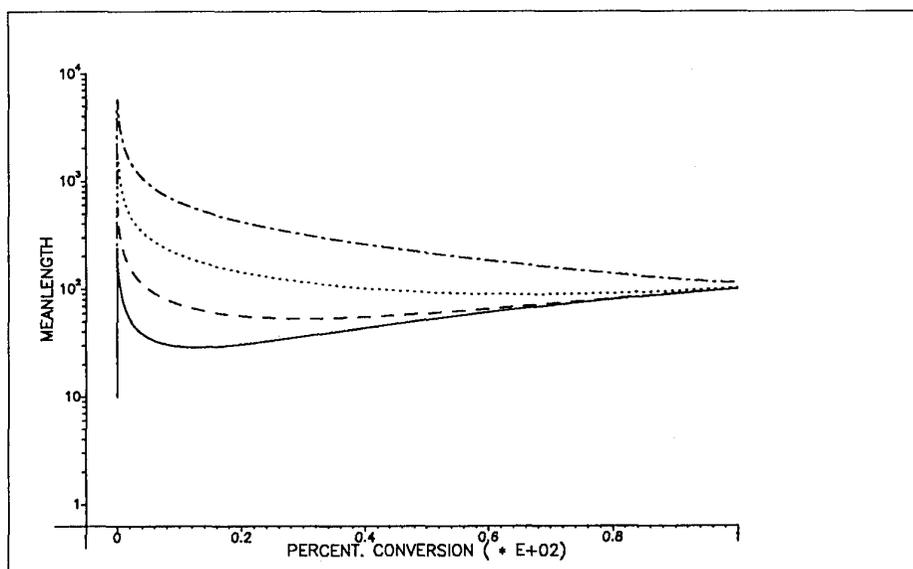
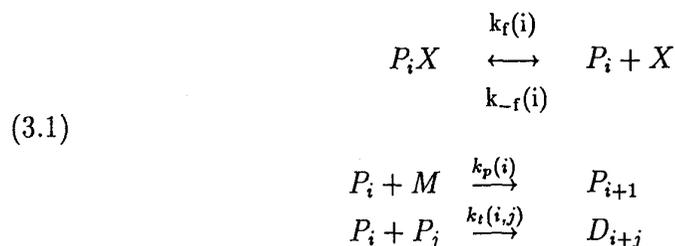


Figure 5: Mean values of the chain length of all polymer species for reaction constants $k_f = 10^{-2}$ (—), 10^{-3} (---), 10^{-4} (···) and 10^{-5} (- · -) versus percentage conversion of the monomer M

supercomputers and was applied to the following reaction system.



Involved chemical species are the monomer M , the radical X , and three different polymer species $P_i X$, P_i , D_i . For more details we refer to [9].

The maximum number of species that could be handled in [9] was limited by the maximum vector length available on the used supercomputer (CYBER 205) as well as by the demand on the central processing units (CPU). Therefore the concentrations D_i , $i = 1, 2, \dots$, were not treated separately for each index i

```

*HEAD
--- Quasi Living Radical Polymerization ---
CH. H. J. Johnson et al., Aust. J. Chem. 43, (1990), 1215
*MODEL PARAMETER
1
*UNIT SYSTEM
5
*REACTION SYSTEM
P + X      => PX      (1.D9)
PX         => P + X   (1.D-5)
P[N]X     => P[N] + X (1.D-5)
P[N] + X  => P[N]X   (1.D9)
P + M     => P[1]    (2.D3)
P[N] + M  => P[N+1]  (2.D4)
P + P     => D       (1.D7)
P + P[N]  => D + P[N] (2.D4)
P[N] + P[M] => D[N+M] (1.D7)
*GALERKIN PROJECTION
P[] 2
D[] 2
P[]X 2
*INITIAL CONCENTRATIONS (0)
M 10.0D0
X 0.0D0
PX 0.1D0
*ACCURACY
1.0D-4
*INTEGRATION BOUNDS
0.0
750000.0
*PRINT PARAMETER
0
*DISTRIBUTION OUTPUT
1 10000 100

```

Table 5: Input file CHEMIN for Quasi Living Radical Polymerization.

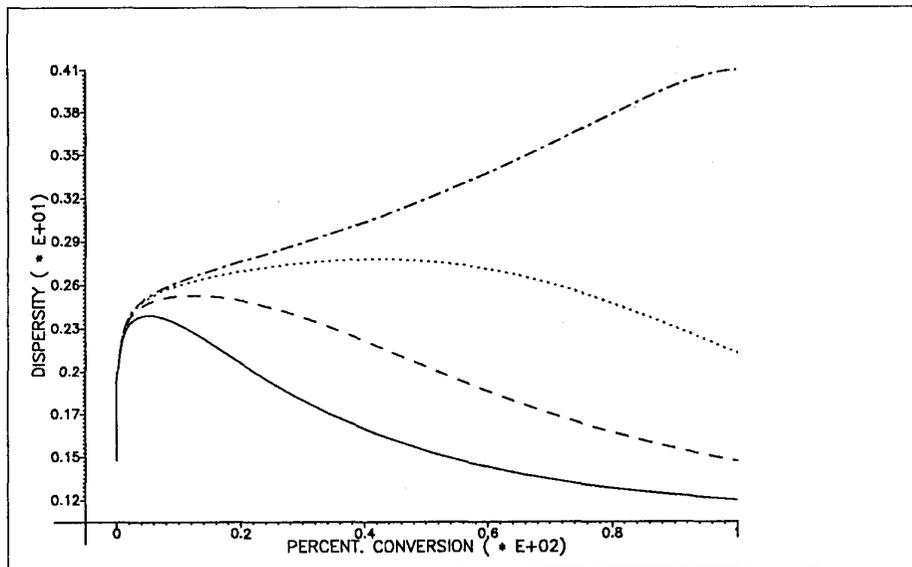


Figure 6: Dispersity of $P_i X$ for reaction constants $k_f = 10^{-2}$ (—), 10^{-3} (---), 10^{-4} (···) and 10^{-5} (- · -) versus percentage conversion of the monomer M

and the remaining two polymer species $P_i X$, P_i were considered only for short time calculations ($t \leq 10s$) with maximal chain length i_{\max} up to $i_{\max} = 500$. For long time calculations (10 – 5000s) the restriction $i_{\max} \leq 200$ was used, because the computing time is proportional to i_{\max}^2 .

With the constants of Table 1 in [9] the MACRON input for the reaction system (3.1) has the form presented in Table 5. Note that the first member of each polymer species is not handled as part of a distribution, but as ordinary chemical species. In this manner different values for $k_p(1)$ and $k_p(i) = k_p$ for $i \geq 2$ can be chosen. The D_i are considered separately for every index i .

First we discuss the properties of main chemical interest, i.e. the average molecular weight (defined in [9] as conversion/number of all polymer chains) and the polydispersity of the mainly built polymer species $P_i X$. With regard to that, it is sufficient to use only two expansion coefficients for each polymer species (see Section 2.2). It is especially interesting to study these quantities in dependency to the rate constant $k_f(i) = k_f$, which in turn depends on the

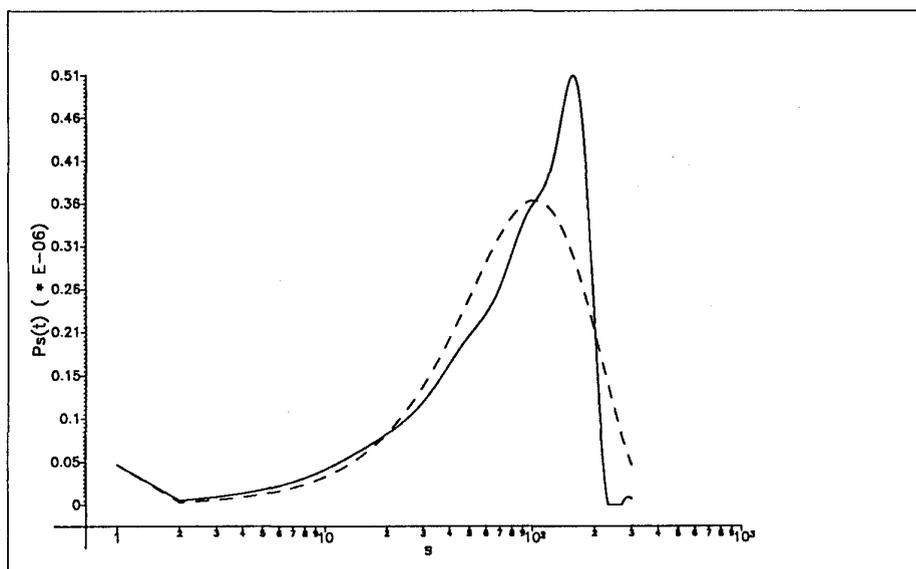


Figure 7: Distribution P_i at $T = 0.001$ s approximated with 2 (---) and 80 (—) polynomials

radical X (see ref. [11]). Following [9] we made experiments with the values $k_f = 10^{-2}$, 10^{-3} , 10^{-4} and 10^{-5} .

In all cases the calculations were performed up to nearly 100 % conversion of the monomer M . This corresponds to reaction times up to 750000 s (for $k_f = 10^{-5}$). One feature of the reaction system (3.1) is, that depending on the k_f -value very long chains are formed in the initial stages of the reaction (see Figure 5). By this, the direct computation of JOHNSON et al. can only state something about the polydispersity for $k_f = 10^{-2}$, where the molecular weights for chain length > 500 is nearly negligible. Even in this case in [9] only extrapolated values are given and the costs to obtain them are very high. For example, about 4.5 h computation time are required on a CYBER 205 to calculate the polydispersity $d = 2.20$ (extrapolated value $d = 2.34$) for the reaction time 10 s and $k_f = 10^{-2}$.

The exact numerical value $d = 2.404$ for $k_f = 10^{-2}$ and reaction time 10 s is obtained by MACRON within 18.1 s CPU time on a workstation (SUN SPARC 1). This is a fairly better use of time, in particular if one is interested in the op-

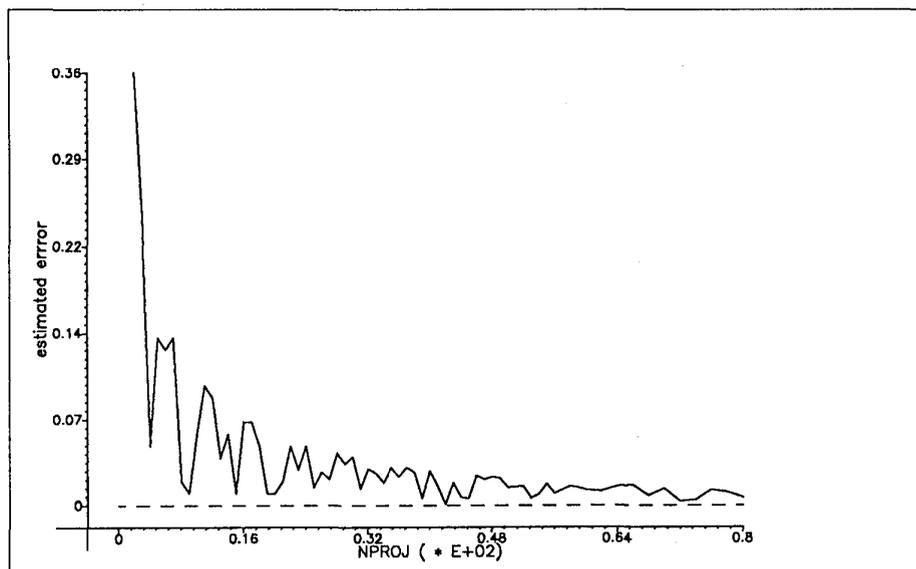


Figure 8: *Estimation of the Galerkin projection error versus the number of polynomials*

timalization of parameters or extension of the simulated reaction times up to 750000 s (100% conversion). Note that our treatment is not restricted by the formation of very long chains. Thus we also can calculate the polydispersity for the k_f -values $k_f = 10^{-3}$, 10^{-4} , and 10^{-5} . The results are plotted in Figure 6. For the simulation of the reaction time 750000 s in case $k_f = 10^{-5}$ we need 44 s CPU time on the SPARC 1. Figure 6 makes clear that the choice of polymers $P_j X$ with high rate coefficient k_f is crucial to get narrow polymer distributions.

Dealing now with the computation of the CLD's we choose the worst case, i.e. the high structured distribution of P_i at $t=0.001$ s. This CLD — plotted in Figure 1 of ref. [9] — has a very narrow peak at chain length ≈ 120 and decreases steeply for higher chain lengths. By a simulation with $NPROJ=2$, which was sufficient for the quantities discussed before, we get the estimate for the Galerkin projection error $\epsilon_2 = 0.36$. The corresponding CLD is plotted in Figure 7.

Increasing $n=NPROJ$ lowers the estimate ϵ_n , but the descent of ϵ_n is superposed by a rather oscillatory behavior (Figure 8). This is very similar to the

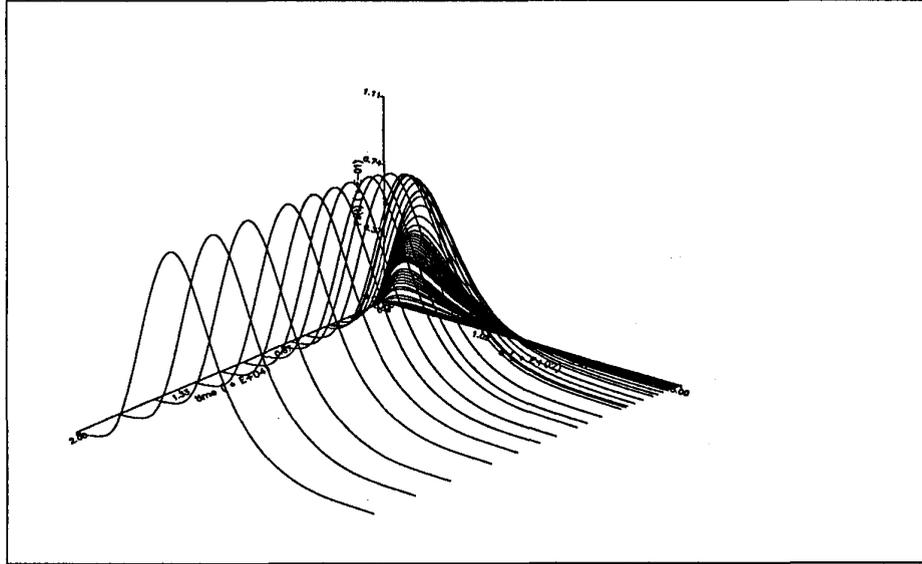


Figure 9: Time evolution of $P_i X$ up to $t = 200000 s$, $k_f = 10^{-2}$.

situation of the step function of Sect. 2.2. In principle a polynomial approximation of the CLD's at $t=0.001$ is possible, but describing the fine structure of the distribution P_i is a difficult task and polynomials of higher degree are needed. Compared to the expensive direct solution, plotted in [9], the approximation with $NPROJ=80$ polynomials shown in Figure 7 can be regarded as the exact solution (with the exception of slight oscillations). MACRON needed 45 s CPU time (in the mode $\rho = \text{constant} = 0.99$) to obtain this result.

Note that this distribution is just the worst case of a CLD we found for this example. At processing times of technical interest (in order of hours), the distributions have not these narrow peaks and steps, which lead to this naughty error behavior. A user of MACRON, interested in the results of a reaction system after a specific reaction time, should know the grade of precision he needs. If he is really interested in $P[s]$ at $t=0.001$ with accuracy 10 %, $NPROJ \approx 12$ would be sufficient. In Figure 9 the time evolution of the species $P_i X$ is presented for $k_f = 10^{-2}$ from $t=0$ up to $t=20000s$, $NPROJ=80$.

As a conclusion we can say that compared to present methods used in the

application, MACRON will save a lot of computing time. A user, most probably a chemist or physicist, can formulate the problem in the appropriate and well-known terms of chemical reactions. Without knowing anything about the resulting system of differential equations and without being an expert in numerical methods and supercomputers he will get reasonable results. In standard cases also for the CLD's. In non-standard cases the implemented features prevent a misinterpretation of the results. Thus, reaction systems like this example can be tested in an easy and economical way by every chemist on his local computer.

4 APPENDIX : MACRON-INPUT

This input description of MACRON is based on a not published manual of LARKIN [7]. Since the development of the program is in a state of flux, many details of the implementation will be subject to change. Updates will be announced in the respective version of MACRON. The possibility to consider thermodynamic effects is restricted to some extent if macromolecular reactions are involved. Necessary information like molecular weights or enthalpy coefficients are not defined for a macromolecular species in general. At best mean values are available. The same is true for a stoichiometric balance check. But we recommend to make use of these features for large pre-reaction systems as far as possible and applicable.

The chemical input is contained in a file called CHEMIN. Depending on the special situation, a second input file THERMO (described under the keyword *ENTHALPY COEFFICIENTS) or an input file describing an initial chain length distributions (see under *GALERKIN PROJECTION) can be used.

In Sections 4.1-2 the format directions of the input are described, in Section 4.3 the available macromolecular reactions are listed.

4.1 GENERAL DESCRIPTION OF THE INPUT FILE CHEMIN

The input is organized in *blocks*, which are opened by a *keyword*-line. A block is closed by opening a next one. Inside the block the user has to supply information according to a special format described below. The order of appearance of the blocks is prescribed in Table 6, but blocks, which are not necessary for the actual model, may be omitted. As far as possible a default option will be used in this case.

The following general rules should be observed by creating an input file:

- i. A keyword line must start in column 1. The space after the keyword may be filled with characters according to the corresponding syntax form.
- ii. A keyword may be abbreviated by its first letters, i.e *E for *ELEMENTS, *EN for *ENTHALPY and so on. For the abbreviations see also Table 6.
- iii. Comment lines must start with "*C" in column 1 and may contain any text after that.
- iv. Only keyword and comment lines may start with a "*" in column 1.
- v. No blank lines are allowed in the input file.
- vi. The first 72 columns of each input line are read. Further information is ignored. There is an exception for the block *ENTHALPY COEFFICIENTS: in this case the first 75 columns are read with a fixed format.

4.2 DESCRIPTION OF THE INPUT BLOCKS

1 Keyword *HEAD

This block offers the possibility to identify the actual input file during the whole MACRON session by any text. Maximum number of lines for this block: 5 lines (Comment lines not counted).

No.	Keyword	Short form	Contents
1	*HEAD	(*H)	Text to identify actual model
2	*MODEL PARAMETER	(*M)	Integer value to select the type of thermodynamical modeling
3	*ELEMENTS	(*E)	Names and weights of elements which compose the species
4	*SPECIES	(*S)	Names and element composition of species
5	*NAME OF THIRD BODIES	(*N)	Names and definition of third bodies
6	*UNIT SYSTEM	(*U)	Integer value to select unit system
7	*TEMPERATURE	(*T)	Initial temperature of system
8	*PRESSURE	(*PRE)	Initial pressure of system
9	*DENSITY	(*DE)	Initial density of system
10	*REACTION SYSTEM	(*R)	Reaction equations and associated kinetic parameters
11	*GALERKIN APPROXIMATION	(*G)	Number of expansion coefficients of the Galerkin approximations and potential input of distributions
12	*INITIAL CONCENTRATION	(*INI)	Initial concentrations of chemical species
13	*RHO VALUES	(*RHO)	Set of constant rho values
14	*ENTHALPY COEFFICIENTS	(*EN)	Coefficients for NASA-fits of thermodynamical data
15	*ACCURACY	(*A)	Relative accuracy for the integrator
16	*INTEGRATION BOUNDS	(*INT)	Lower and upper integration bound
17	*OUTPUT POINTS	(*O)	Additional output points
18	*PRINT PARAMETER	(*PRI)	Parameter for desired additional output files
19	*DISTRIBUTION OUTPUT	(*DI)	Range of indices for Output of CLD's

Table 6: Name, function and order of blocks.

Example:

```
*HEAD
HERE YOU CAN WRITE UP TO 5 LINES OF ARBITRARY TEXT
TO IDENTIFY THE REACTION SYSTEM
```

2 Keyword *MODEL PARAMETER

*MODEL PARAMETER specifies the type of thermodynamic modeling used for the actual chemical problem by setting the internal variable MODEL. See Table 7 for the possible choices and [12] for detailed description of the thermodynamical modeling. The cases MODEL=2,4,6 require user supplied subroutines (see [12]).

Syntax rules:

- i. The Block must consist of exactly one line besides the keyword line containing the integer number MODEL anywhere in the line.

MODEL	thermodynamical modeling
1	constant temperature
2	prescribed time-dependent temperature
3	constant density
4	prescribed time-dependent density
5	constant pressure
6	prescribed time-dependent pressure

Table 7: Possibilities of thermodynamical modeling.

Example: (Choice of constant temperature model)

```
*MODEL PARAMETER
1
```

3 Keyword *ELEMENTS

In this block the elements, which compose a chemical species (an element must not necessarily be a chemical element), are defined. The definition of elements (and the associated species composition in block *SPECIES) allows to make a stoichiometric balance check for each chemical equation defined in block *REACTION SYSTEM. If, in addition, the atomic weight of each element is given, species molecular weights are computed and can be used to compute the initial density of the actual system internally.

Stoichiometric analysis does not work whenever macromolecular species are involved. However, it is recommended to check large pre-reaction systems separately.

Syntax rules:

- i. A maximum number of MAXEL=10 elements can be declared.
- ii. Element names may be composed of up to ELEML=5 alphanumeric digits (but no blank, "[" or "]"), but may not start with "*" or "(".
- iii. Each line of this block may contain only one element declaration.
- iv. Each element name may be followed by its atomic weight (as a real number) separated by at least one blank.
- v. Each element may be declared only once.
- vi. The element declaration may appear anywhere in the line.

Example: (Declaration of elements H and O)

```
*ELEMENTS
H          1.0
O          16.0
```

4 Keyword *SPECIES

This block can be used to define a list of species names which will appear in the reaction mechanism.

If a stoichiometric balance check is desired, all chemical species of the reaction mechanism must be declared line by line in this block, followed by their element composition. Otherwise the actual list of declared species can be a subset of all species appearing in the reaction mechanism. The information of the blocks *ELEMENT and *SPECIES can also be used for the internal computation of the initial density.

Syntax rules:

- i. A maximum number of MAXSP=999 species can be declared.
- ii. A species name may be composed of up to NAMEL =10 arbitrary alphanumeric digits (but no blank, "[" or "]"), but may not start with a "*" or "(".
- iii. Each line of this block may contain only one species declaration.
- iv. Each species name may be followed by its element composition.
- v. An element composition consists of a sequence of element names, each of them followed by an integer number indicating how many elements of this type compose the species.
- vi. Each species may be declared only once.
- vii. The species declaration may appear anywhere in the line.

Example: (Declaration of Species WATER, OH*, H2 and H*)

```
*SPECIES
WATER           H 2 0 1
OH*             H 1 0 1
H2              H 2
H*              H 1
```

5 Keyword *NAME OF THIRD BODIES

This block can be used to define a list of third body species, which will appear in the reaction mechanism. See [12] for general information on third bodies.

Syntax rules:

- i. A maximum number of 5 third bodies can be declared.
- ii. A third body name may be composed of up to NAMEL= 10 alphanumeric digits (but no blank, "[" or "]"), but may not start with a "*" or "(".
- iii. Each line of this block may contain only one third body declaration.
- iv. Each third body name can be followed by its species composition
- v. A species composition starts with "=" followed by a sequence of species names each of them preceded by a real number which indicates the weight and followed by a "+" (except the last species). If the line ends with a real number or a "+", then the next line of input will be used also for the definition of the composition of the actual third

body. "=", "real number", "species name" and "+" must be separated by at least one blank. If no species composition of the third body is given, this third body will be regarded as the unweighted sum of all species.

- vi. Each third body may be declared only once.
- vii. The third body declaration may appear anywhere on the line.

Example: (Declaration of M)

```
*THIRD BODIES
M = 0.35 H* + 6.5 WATER + 0.4 OH* + 1.5 H2
```

6 Keyword *UNIT SYSTEM

This block specifies the unit system for the actual system. All input values in CHEMIN are assumed to be given in these units. Possible choices are listed in Table 8.

Syntax rules:

- i. This block must consist of exactly one line besides the keyword line containing the value of the integer number IUNIT. IUNIT specifies the unit system for the actual system.

Example: (Choice of unit system 1, see Table 8.)

```
*UNIT SYSTEM
1
```

IUNIT	unit system					
	C	T	P	S	A	E
1	(mole, m^3)	(K)	(PA)	(g,m^3)	(mole, m^3,sec)	(J)
2	(KJ)
3	(mole, cm^3)	...	(atm)	(g,cm^3)	(mole, cm^3,sec)	(J)
4	(KJ)
5	(mole,l)	...	(cal,l)	(g,l)	(mole,l,sec)	(cal)
6	(kcal)

Table 8: Unit systems. (C: concentration, T: temperature, P: pressure, S: density, A and E: parameters of the Arrhenius law)

7 Keyword *TEMPERATURE

The block consists of one line with a positive real number setting the initial value of temperature. No initial temperature is required if MODEL= 1 or MODEL= 2, no Arrhenius law is used and the initial concentrations are not given in mole fractions.

Example: (Choice of (initial) temperature, for units see *UNIT SYSTEM.)

```
*TEMPERATURE  
1200.D0
```

8 Keyword *PRESSURE

The block consist of one line with a positive real number setting the initial value of pressure. No initial pressure is required, if MODEL= 1 or MODEL= 2. If initial concentrations are not given in mole fractions and temperature is positive then the initial pressure is internally computed by means of the ideal gas law. In cases where the initial pressure can be computed internally *and* is given in this block, the latter value overwrites the computed one. Both values will be compared and different values will cause a warning message.

Example: (Choice of (initial) pressure , for units see *UNIT SYSTEM.)

```
*PRESSURE  
4.36506
```

9 Keyword *DENSITY

The block consists of one line with a positive real number setting the initial value of density. No initial density is required for MODEL \leq 4. A value for density is computed internally using the information from blocks *SPECIES and *ELEMENTS (to get species weights). Therefore this information should be complete and correct. If a value for density is given in this block, it will be used for the simulation.

Example: (Choice of (initial) pressure , for units see *UNIT SYSTEM.)

```
*DENSITY  
1.4
```

10 Keyword *REACTION SYSTEM

In this block the whole reaction mechanism has to be defined. The reactions may be reversible or irreversible (internally counted as two reactions). The number of reactants (and products) of any equation is optional to allow the modeling of special effects – except third bodies for which only one appearance per equation is possible. Note, that the full declaration of species in block *SPECIES permits also the detection of typing errors.

Syntax rules:

- i. A correct *reaction equation* is composed of the following components (all separated by blanks):
 - a. species symbols
 - b. delimiters
 - c. kinetic parameter fields
- ii. A reaction may be written on several consecutiye lines.

- iii. The *species symbol* is a unique sequence of (up to 10) alphanumeric digits (but no blank space) and may not start with a "*" or "(" . A species symbol is either a species name, a third body name or a macromolecular species. The latter is identified by containing the pair "[]" in the name, "[" and "]" may stay anywhere in a species symbol. The string enclosed in "[" and "]" will be interpreted as index (e.g. chain length). More than one pair is not allowed. The name of such a species is the species symbol without the index. For example the species symbol "P[M]" leads to the species name P[] with index M and "N[I+2]O" to the species name N[]O with index I+2. In general the index of every macromolecular species will be ignored for its identification. Note that "[" and "]" are parts of the name. The syntax for the species will be checked. No stoichiometric coefficients can be given.
- iv. There are 3 types of valid delimiters:
- + delimiter between two species symbols (reactants, products)
 - => delimiter between reactants and products for an irreversible equation
 - <=> delimiter between reactants and products for a reversible equation
- v. The *kinetic parameter field* consists of 1 up to 3 numbers (separated by commas and enclosed in parenthesis). If only one number is given, this value is assumed to be the rate reaction coefficient. If 2 (or 3) numbers are given, the rate constant is computed internally according to the (modified) Arrhenius law (RC: reaction constant, TEMP: temperature) :

$$RC := A * \exp(-E/(R*TEMP)) * TEMP**ALPHA$$

where the first kinetic parameter is assumed to be A, the second to be E and the possible third to be ALPHA. To define the rate constants for a reversible equation there are two possibilities:

- a) A *reversible* reaction is followed by two kinetic parameter fields, where the first is assigned to the direction "=" and the second to "<=" .
 - b) If there is only one kinetic parameter field after a reversible equation, then the reverse rate constant is computed via the equilibrium constant (this is only possible if thermodynamical data is available for all species appearing in this equation).
- vi. The use of *reaction inequalities* is possible, i.e. an equation may have no reactant or no product, to allow the modeling of addition (subtraction) of substances during the reaction. Note that this is in contradiction with the assumptions made for the modeling of thermodynamics.
- vii. A reaction equation may consist of an arbitrary number of lines, i.e. the input is regarded as a stream and the separation of the equations is not indicated by the end of the line, but by the end of the associated kinetic parameter field.
- viii. The syntax of reactions containing a macromolecular species is checked to match one of the implemented macromolecular processes (see Appendix 4.3).

Example: (Three reactions, see also Figure 1 and 5 at page 8 and 17, respectively.)

```
*REACTION SYSTEM
H2 + O2      =>  H* + HO2          (.6D2)
O3 + M      <=>  O2 + O + M      (0.38D09, 0.950D05)
C2H2 + O    <=>  CO + CH2       (0.41D9,0.7D4,1.5) (.2D8,1.D3,1.)
```

11 Keyword *GALERKIN APPROXIMATION

The internal approximation of CLD's is characterized by the number of expansion coefficients NPROJ and the weight function parameter ρ . This block enables to determine the values of NPROJ as well as to start the simulation with an initial chain length distribution. For each macromolecular species one line may be supplied. The species name must be given completely, i.e. with "[]" and "]". A given index (enclosed in "[]") will be ignored. A species name may appear only once. The following particularities have to be regarded.

- i. Range of NPROJ: $2 \leq \text{NPROJ} \leq 100$.
Default value: NPROJ = 10.
- ii. If the simulation has to be started with an initial distribution, there is a possibility to enter measured distributions as well as standard types. A filename with input data can be entered after the keyword *F in the same line. This file must consist of lines containing an index (e.g. chain length) and the *concentration* of the associated macromolecules. The index must appear in increasing order, the differences between the indices (≥ 1) are arbitrary. Comment lines are allowed and have to begin with "*C" in the first column. NPROJ is automatically chosen, except when the user prescribes another value. If the computed value is higher than the given value, this will lead to a warning.

Be sure, that the initial distribution is a *number chain length distribution* or a *number molecular weight distribution*. That means, that the quotient of the 1st and 0th moments must be the number mean value of the distribution (possibly multiplied by the molecular weight of the monomer) !

To start the simulation by a Schulz-Flory distribution with parameter ρ , enter *SF followed by a value for ρ . The initial concentration of the macromolecular species must be given in *INITIAL CONCENTRATION.

An initial distribution of the form $u_1 = c, u_s = 0$ (only monomer at the beginning) can be declared by *D followed by a the value for c.

Example: (Input data for species P[] in the given filename, 7 expansion coefficients prescribed.)

```
*GALERKIN APPROXIMATION
P[ ] 7 *F 'todays.data'
```

Example: (Species D[] started as Schulz-Flory distribution with parameter $\rho = 0.99$ and default value NPROJ=10.)

```
*GALERKIN APPROXIMATION
D[ ] *SF 0.99
```

12 Keyword *INITIAL CONCENTRATIONS (I) (I=0,1)

This block is used to define the initial concentrations of the declared species (involved in the reaction mechanism) at the very beginning of the simulation (i.e. concentrations of species at starting point of simulation). Only non-zero values are required because the default concentration of all species appearing in the mechanism is zero.

To indicate that the unit for the initial concentrations is the molar concentration (according to the value of IUNIT) the keyword ***INITIAL CONCENTRATION (0)** is required. To indicate that the unit for the initial concentrations is the mole fraction the keyword ***INITIAL CONCENTRATION (1)** must be used. Each line of this block may contain only one initial concentration definition.

- i. An initial concentration definition is composed of one species name and one number for the associated initial concentration.
- ii. A species may appear only once.
- iii. Species symbol and associated value may appear anywhere in the line, separated by at least one blank space.

Example: (Species A with molar concentration 0.003.)

```
*INITIAL CONCENTRATIONS (0)
A 0.003
```

13 Keyword ***RHO VALUES**

In special cases it may be advantageous to use a constant ρ -value for the approximation of a macromolecular distribution. This can be done in this block. According to the syntax of the previous block ***INITIAL CONCENTRATIONS** an input line containing the name of a macromolecular species followed by a ρ -value between 0.0 and 1.0 may occur for every macromolecular species. No adaptation of these ρ -values during the whole processing time will be done. In the best case, this will increase the numerical stability and decrease the cpu time, but if wrong ρ -values are chosen a Galerkin approximation will not be possible. Thus we recommend to use this facility only after a simulation with **NPROJ=2**. By the maximum of the mean chain length (s) during the whole processing time you can suppose a good ρ -value to be around $\rho = 1.0 - 1/(s)$. But notice, that this value will be used during the whole processing time, whereas the standard ρ -adaptation leads to the optimized ρ -values at every time step.

Example: (Set of constant ρ -values for P[] and D[] to 0.99 and 0.998, respectively)

```
*RHO VALUES (0)
P[] 0.99
D[] 0.998
```

14 Keyword ***ENTHALPY COEFFICIENTS**

This block is used to define the coefficients for the NASA-fits of thermodynamical data (see [12]) and is not required in two cases:

- a. **MODEL** ≤ 2 and no equilibrium constant is needed (to compute reverse kinetic parameters)
- b. all required thermodynamical data can be found in file **THERMO**

Data given here overwrite data from file **THERMO**. If **MODEL** ≥ 3 , thermodynamical data should be available for all species in the reaction system, missing coefficients are internally set to zero. Data for species not appearing in the reaction system are ignored.

Syntax rules (same as for file THERMO):

- i. The total block consists of its keyword line and an arbitrary number of subblocks.
- ii. A subblock is defined by one line containing the species name followed by 3 lines containing the coefficients for this species. The data must be in the following format:

5E15.8 (also be effective for the file THERMO)

Example: (Enthalpy coefficients for species O2)

```
*ENTHALPY COEFFICIENTS
O2 MILLER J12/820 2 0 0 OG 300.000 5000.000      1
  0.36811543E+01 0.61174575E-03-0.10959707E-06 0.10993072E-10-0.38448244E-15 2
-0.12226443E+04 0.32938614E+01 0.33175726E+01 0.34704781E-03 0.14182033E-05 3
-0.78923112E-09-0.99682740E-13-0.10160647E+04 0.55977631E+01      4
```

15 *ACCURACY

The block consists of one line with the required relative accuracy for the integrator.

Remark: The accuracy should be chosen in the range 1.E-2 – 1.E-6.

Example: (Relative accuracy for the integrator set to 10^{-4})

```
*ACCURACY
1.D-4
```

Remark: Difficulties for the integration and approximation may arise in case of too low accuracy demands. The user can test this by increasing the required accuracy (for example by a factor 10).

16 *INTEGRATION BOUNDS

The block consists of two lines, containing the lower and upper integration bound in seconds.

Example: (Simulation from 0 s to 1200 s)

```
*INTEGRATION BOUNDS
0.0
1200.0
```

17 *OUTPUT POINTS

In the first line of this block the number of output points is defined. In the following lines the values of the output points have to be given line by line. The upper integration bound is a default output point.

Example: (Three additional output points at 300 s, 600 s and 900 s)

```
*OUTPUT POINTS
3
300
600
900
```

18 *PRINT PARAMETER

This block consists of one line to define the integer print-parameter *IPRINT* for additional simulation output on several files. *IPRINT* may be chosen in the range $0 \leq \text{IPRINT} \leq 6$. The effect of *IPRINT* depends on the special program version (graphic adaption). See the information menu in an interactive session. As a rule *IPRINT*=0 and *IPRINT*=6 correspond respectively to minimal and maximal output.

Example: (Maximal output)

```
*PRINT PARAMETER
6
```

19 *DISTRIBUTION OUTPUT

For the output of a *CLD* between chain length *SMIN* and *SMAX* with increment *INC*, these three integer values have to be given in one line.

Example: (*CLD* output in the chain length range 1 - 10000 with increment 100, e.g. 100 points)

```
*DISTRIBUTION OUTPUT
1 10000 100
```

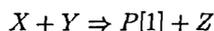
4.3 MACROMOLECULAR PROCESSES

A number of typical macromolecular reaction steps is implemented up to now. The modular character of the method allows an easy continuation of this list, if the reactions are analytically preprocessable. Whenever a reaction in block **REACTION SYSTEM* contains at least one pair "[]" (sign of a macromolecular species), the reaction input line is interpreted as a macromolecular reaction step. The reaction is then examined on whether it matches one of the implemented reaction steps below. Therefore the user has to regard strictly the syntax of a macromolecular reaction. The reactions are related to the reaction numbers below, a control output is given. In case of syntax errors or reactions not to analyze, suggestions will be made concerning input line.

In the following we use the names *P*[], *Q*[] for macromolecules, *X*, *Y* and *Z* for standard chemical species and *N*, *M* for integer values to characterize the index of a macromolecule. In case of no other specification, the integers *N*, *M* are running variables $N, M = 1, 2, 3, \dots$. Thus most of the reactions below are synonyms for infinite reaction systems (see Introduction).

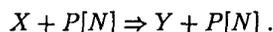
The macromolecular reaction steps also have an effect on the standard chemical species expressed in terms of their total concentration (the 0th moment). The number of involved chemical species in the macromolecular reaction is arbitrary. The degradation processes (*REAC*=9, 10) are an exception, since the situation is more complicated and no chemical species should occur. We only present standard forms for the reaction steps. The letter *k* always denotes the reaction constant. The attributes *closed* and *open* indicate whether the expansion coefficients generally depend on higher indices. This is important for the choice of *NPROJ*, if the user only wants to compute statistical moments.

- i. Initiation (*REAC* = 0, closed). Initiation of macromolecular species by chemical species has the form

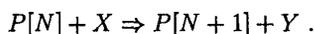


In particular the index of $P[\]$ must be 1. This is not really a restriction for the modeling of reaction systems. The choice of a first chemical species to be the first member of a macromolecular distribution is optional. A finite number of species, which are members of a macromolecular distribution, can be handled as a usual chemical species (see Table 5 at page 17).

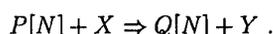
- ii. Macromolecular Species as Reaction Partner ($REAC = 1$, closed). A macromolecular species may initiate a chemical reaction, where it only occurs as a reaction partner.



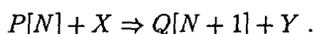
- iii. Chain Addition ($REAC = 2$, closed). The main process describing the growth of macromolecular species is the chain addition step.



- iv. Direct Transfer ($REAC = 3$, closed). A macromolecular species can react directly to another macromolecular species $Q[\]$.

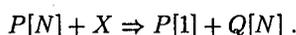


- v. Chain Addition Transfer ($REAC = 4$, closed). A transfer reaction may also have the form



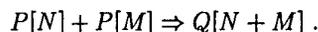
Such a reaction class is e.g. necessary to describe the growth of macromolecular species by a cyclic reaction system.

- vi. Chain Transfer ($REAC = 5$, closed). A typical reaction in polymer chemistry is the chain transfer



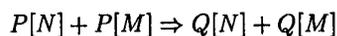
It can be considered as a combination of the direct transfer and the initiation process.

- vii. Combination Transfer ($REAC = 6$, closed). The coagulation transfer is often used as a termination process

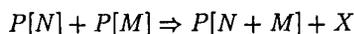


(No blank space in the index of $Q[\]$!)

- viii. Double Transfer ($REAC = 7$, closed).



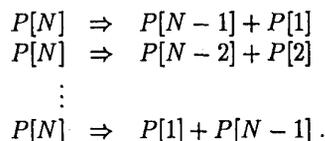
- ix. Combination ($REAC = 8$, closed).



- x. Degradation ($REAC = 9$, open). The degradation (cracking) of macromolecules is formally written as

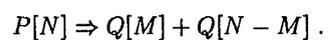


This is a short notation for:

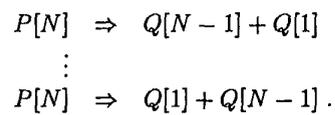


The input must have the condensed form (*).

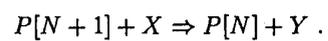
xi. Degradation Transfer ($REAC = 10$, open). As before the formal reaction



symbolizes the reaction sequence



xii. Reverse Chain Addition ($REAC = 11$, open).



REFERENCES

- [1] R. Bradel, A. Kleinke, M. Wulkow: *Auf- und Abbau von Polyhydroxybuttersäure in Bakterien*. Poster No.10, Makromolekulares Colloquium, Freiburg (1991).
- [2] U. Budde, M. Wulkow: *Computation of Molecular Weight Distributions for Free Radical Polymerization Systems*. Chem. Ing. Sci., Vol. 46, No. 2 (1991), pp. 497-508.
- [3] G. Bader, U. Nowak, P. Deuffhard: *An Advanced Simulation Package for Large Chemical Reaction Systems*. Univ. Heidelberg, SFB 123: Techn. Rep. 149 (1982).
- [4] P. Deuffhard: *On Algorithms for the Summation of Certain Special Functions*. Computing 17, (1976) pp. 35-48.
- [5] P. Deuffhard: *Recent Progress in Extrapolation Methods for Ordinary Differential Equations*. SIAM Rev. 27, pp. 505-535 (1985).
- [6] P. Deuffhard, G. Bader, U. Nowak: *LARKIN - A Software Package for the Simulation of Large Systems Arising in Chemical Reaction Kinetics*. IN: K.H. Ebert, P. Deuffhard, W. Jaeger (Eds.): *Modelling of Chemical Reaction Systems*. Springer Series Chem. Phys. 18 (1981).
- [7] P. Deuffhard, U. Nowak : *Efficient Numerical Simulation and Identification of Large Chemical Reaction Systems*. Ber. Bunsenges. Phys. Chem. 90 (1986).
- [8] P. Deuffhard, M. Wulkow: *Computational Treatment of Polyreaction Kinetics*. IMPACT Comput. Sci. Eng., 1 (1989).
- [9] Ch.H.J. Johnson et al.: *Quasi Living Radical Polymerization*. Aust. J. Chem. 43, p. 1215 (1990).
- [10] W.H. Ray: *On the Mathematical Modeling of Polymerization Reactors*. Macromol. Sci.-Revs. Macromol. Chem. C8 1, pp. 1-56 (1972).
- [11] E. Rizzardo, Chem. Aust. 43, p. 32 (1987)
- [12] D. Walkowiak: *Numerische Behandlung großer, adiabater chemischer Reaktionssysteme*. Universität Heidelberg, Diploma Thesis (1986).
- [13] M. Wulkow: *Numerical Treatment of Countable Systems of Ordinary Differential Equations*. Konrad-Zuse-Zentrum Berlin, Technical Report TR-90-8 (1990).
- [14] M. Wulkow, P. Deuffhard: *Towards an Efficient Computational Treatment of Heterogeneous Polymer Reactions*. Konrad-Zuse-Zentrum Berlin, Preprint SC-90-1 (1990).