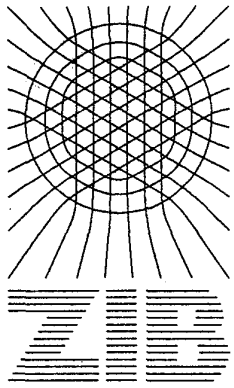
Konrad-Zuse-Zentrum für Informationstechnik Berlin



MICHAEL WULKOW JÖRG ACKERMANN

The Treatment of Macromolecular Processes with Chain-Length-Dependent Reaction Coefficients – An Example from Soot Formation

Preprint SC 91-18 (December 1991)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Heilbronner Str. 10
1000 Berlin 31
Verantwortlich: Dr. Klaus André
Umschlagsatz und Druck: Rabe KG Buch-und Offsetdruck Berlin

ISSN 0933-7911

The Treatment of Macromolecular Processes with Chain-Length-Dependent Reaction Coefficients – An Example from Soot Formation

MICHAEL WULKOW JÖRG ACKERMANN

Konrad–Zuse–Zentrum für Informationstechnik Berlin, Heilbronner Strasse 10, D-1000 Berlin 31, Germany

December 1991

ABSTRACT

The description of chain-length distributions in macromolecular reaction kinetics leads to so-called countable systems of differential equations. In particular, when the appearing reaction rate coefficients depend on the chain-length of the reacting macromolecules itself, an efficient numerical treatment of these systems is very difficult. Then even the evaluation of the right-hand side of the system can become prohibitively expensive with respect to computing time. In this paper we show, how the *discrete Galerkin method* can be applied to such problems. The existing algorithm CODEX is improved by use of a *multiplicative error correction scheme* for time discretization and a new type of *numerical preprocessing* by means of a *Gauss summation.* Both ideas are exemplary for a wide class of approximation types and are described very briefly here. The new numerical techniques are tested on an example from soot formation, where the coagulation of molecules is modeled in terms of reaction coefficients depending on the radii of the particles and their collision frequency.

Contents

1 INTRODUCTION

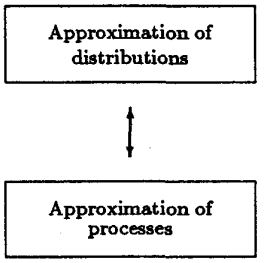
The mathematical description of a chain-length distribution (CLD) in a macromolecular reaction leads to a *countable system of ordinary differential equations* (CODE), which consists of single differential equations of the form

$$u_s'(t) = f_s(u_1(t), u_2(t), \dots) \ , \quad s = 1, 2, \dots, s_{\max} \ . \tag{1.1}$$

In the present context, the variables $u_s(t)$ may denote the concentrations of macromolecules with chain-length s at time t. The whole vector (sequence) of the $u_s(t)$ is called $u(t)$. The right-hand side functions $f_s(u) = f_s(u_1, u_2, \dots)$ (time dependency omitted) arise from the modeling of a macromolecular process, e.g.

$$f_1(u) = -u_1 \ , \ f_s(u) = u_{s-1} - u_s \ , \ s = 2, 3, \dots, s_{\max} \ ,$$

in a chain addition polymerization. In contrary to an ordinary differential equation, the upper index $s_{\max}$ is usually very large ($10^4 - 10^6$ in practical examples) or even *infinite*. If an infinite system has to be truncated at chain-length $s_{\max}$ for computational reasons, this value is rarely known a priori and may vary with time t. Thus a system (1.1) cannot be treated by standard numerical methods for ordinary differential equations and ideas for the following two tasks have to be developed:



An approximation of *distributions* is necessary, when a chain-length distribution has to be represented in a computer (data compression). Examples are the use of statistical moments, selection of fractions (lumping) or approximations by special functions. The approximation of *processes* implies the discretization of (1.1) with respect to time as well as the numerical evaluation of the right-hand side, which can take a lot of time (Section 4). The two tasks in the above diagram are connected by the requirement, that it must be possible to perform both with *one* approximation scheme. For example, if we characterize a CLD by its statistical moments, the representation of $f_s(u)$ in terms of the moments of u_s is necessary to derive differential equations for the moments. Similarly,

if we select certain fractions u_{s_i}, only these fractions may be used for an approximation of $f_s(u)$. For both methods this is not always possible. Moreover, the whole algorithm must be *efficient*, implying that the number of degrees of freedom of an approximation has to be as small as possible. This restricts the generality of an approach on the one or on the other side (i.e. treatment only of certain types of CLD's and/or certain processes) and has to be kept in mind, when a numerical method is discussed.

In the discrete Galerkin method, presented in several versions in [5], [1] and [12], a chain-length distribution is expanded into a series of certain special functions. This expansion is truncated after $n+1$ coefficients leading to a Galerkin approximation u_s^n

$$u_s = \Psi(s) \sum_{k=0}^{\infty} a_k \, l_j(s) \;\longrightarrow\; u_s^n = \Psi(s) \sum_{k=0}^{n} a_k \, l_j(s) \,,$$

with a weight function Ψ and polynomials $l_j(s)$. The polynomials $l_j(s)$ are connected with the weight function by the *orthogonality* relation

$$\sum_{s=1}^{\infty} l_j(s) \, l_k(s) \, \Psi(s) = \begin{cases} \gamma_j \,, & \text{for } j = k \\ 0 \,, & \text{for } j \neq k \end{cases}$$

(for details see [5] or [12]). The above approach is very efficient, whenever there are certain similarities between the CLD u_s and the weight function Ψ. On the other hand, the choice of a weight function restricts the class of distributions to be approximated (for a discussion see [1]). Based on the above idea, with the program package MACRON [1] the treatment of those processes from polymer chemistry is possible, which allow a so-called *analytical preprocessing*. Differential equations for the expansion coefficients a_k can be *explicitly* derived in this case. This restricts the class of valid processes, but for the remaining set of problems (which is not empty!) MACRON is a quite efficient tool.

In this paper, we go a step further and present the crucial points of an approximation scheme for very general processes. This is documented on an example from soot formation, where the coagulation of molecules is modeled in terms of complicated chain-length-dependent reaction rate coefficients (Section 2). Up to now, this example could only be treated by a direct integration of a large scale ODE system – leading to tremendous computing times on a supercomputer. Recently developed numerical techniques, which will be sketched below, reduce such computational effort to CPU times of about one minute on a workstation to obtain results within a relative accuracy of 5 - 10 %. At the moment, there are some restrictions concerning the approximation of general distributions, but the new techniques can easily be applied to extended schemes.

The present version of the research code CODEX [12] uses a *multiplicative error correction scheme* for time discretization(Section 3) and a new type of *numerical* preprocessing by means of a *Gauss summation* (Section 4). The weight function is specified to be

$$\Psi_{\rho,\alpha}(s) = (1-\rho)^{1+\alpha}\,\rho^{s-1}\binom{s-1+\alpha}{s-1}\ ,\ 0<\rho<1\ ,\ \alpha>-1\ ,$$

which has various approximation properties and leads to the *modified discrete Laguerre polynomials*. We explain the difficulties and motivations briefly, details of the used concepts can be found in [12]. For a discussion of the different time discretizations used in MACRON and CODEX see [11].

2 A Model from Soot Formation

Coagulation (combination) processes can be described in the chemical notation by

$$P_r + P_s \xrightarrow{k_{sr}} P_{r+s}\ ,$$

where P_s may denote a polymer molecule or a soot (smog) particle of size s. This reaction module appears frequently in applications - distinguished by different modelings of the reaction rate coefficients k_{sr}. In *polymer chemistry* often moment dependent rate coefficients are in use (e.g. to model the gel effect), whereas the modeling of surface effects for the combination of *soot particles* leads to coefficients dependent on the size of the reacting molecules. The countable system of a coagulation process reads in general ($u_s(t)$: number chain-length distribution)

$$(2.1)\quad u_s'(t) = f(u(t))(s) := \frac{1}{2}\sum_{r=1}^{s-1} k_{r,s-r}\,u_r(t)\,u_{s-r}(t) - u_s(t)\sum_{r=1}^{\infty} k_{sr}\,u_r(t)\ ,$$

for $s = 1, 2, \ldots$. In our example, the following reaction coefficients are suggested:

$$(2.2)\qquad k_{s,r} := k_p\left(\frac{1}{r}+\frac{1}{s}\right)^{-1/2}\left(r^{1/3}+s^{1/3}\right)^2\ ,\ k_p \text{ constant.}$$

Such reaction constants are typical for the modeling of coagulation processes of (spherical) particles in the gas phase. We can write $k_{s,r}$ in terms of the relative velocity $v_{s,r}$ of particles of size s and r (in a gas at constant temperature) and their reaction cross section $\sigma_{s,r}$

$$k_{s,r} = v_{s,r}\,\sigma_{s,r}\ .$$

The velocity $v_{s,r}$ is proportional to $\mu_{s,r}^{-1/2} \sim (s+r)^{1/2}/(sr)^{1/2}$, μ the reduced mass, whereas $\sigma_{s,r}$ (in the noninteracting particle approximation) is proportional to the square of the sum of their radii. Assuming further spherical particles we get

$$k_{s,r} \sim \frac{(r+s)^{1/2}}{(rs)^{1/2}}(r^{1/3}+s^{1/3})^2 \quad ,$$

which is just the above form. For more details concerning the constant k_p, which clearly depends on the gas temperature and the particle mass density, we refer to [7] and references quoted there. Because the constant k_p only determines the time scale of the process, for simplicity we set in our model problem $k_p = 1$ and the reaction time large enough to reach the physically relevant regime.

Note, that the applicability of the algorithm CODEX does not depend on this special choice of the coefficients k_{sr}.

The general problem (2.1) has been attacked by different authors. In [7] an attempt is made to get a special approximation of the moments just for the coefficients (2.2). In [9] a discrete Fourier transform is applied, but this requires a certain separation of r and s in the expression for the k_{rs}. A continuous modeling as in [8] leads to theoretical difficulties.

In order to obtain a reference solution of (2.1), we performed direct time integration of a truncated system as an ODE (replace ∞ by $s_{\max}$ in (2.1), $s_{\max}$ large enough). Such an integration up to an interesting t_{end} took more than 3.5h (CPU) on a Cray-YMP. This time would be even larger, if the truncation index $s_{\max}$ was not known from the simulations with CODEX a priori ! A realistic number of size-classes is given in [7] to be about $s_{\max} = 10000$. By the way we note, that the whole simulation with CODEX is independent of the parameter $s_{\max}$.

3 TIME DISCRETIZATION

We write the system (1.1) in a closed form

$$u'(t) = f(u(t)) \ , \ u(0) = \varphi \ ,$$

where $u(t) = (u_s(t))_s$ may be a chain-length distribution at time t and $f(u(t))$ a process as in (2.1). The idea of the *Rothe method* [2] is to discretize this equation as if it was an ODE. The arising stationary subproblem is then solved by the discrete Galerkin method within a certain accuracy supplied by a *time-step control.* In order to obtain an approximation $u^1 = u_{t+\tau}$ of $u(t+\tau)$ in a time step of size τ, we apply the *semi-implicit Euler* scheme [4]:

$$
\begin{aligned}
(I - \tau A)\, \Delta u &= \tau f(u(t))\,, \\
u^1 &= u(t) + \Delta u\,,
\end{aligned}
\tag{3.1}
$$

with A the (Frechét) derivative $f_u(u(t))$ of the right-hand side $f(u)$.

The task is then to get an estimation η^1 of the time error $\|u^1 - u(t+\tau)\|$ ($\|\cdot\|$ an appropriate norm) for predicting a new reasonable step-size $\bar{\tau}$. This is usually be done by computing a 'better' approximation u^2 and then taking the difference $\|u^2 - u^1\|$ as an estimate of the time error. However, in the case of countable systems, u^1 and u^2 can only be *approximated* and it turned out [2], that the respective approximation error has to be comparatively small for not perturbing the time error estimation. In order to avoid this disadvantage, BORNEMANN developed a so-called *multiplicative error correction scheme* [3], which allows the *direct* computation of the time error estimate. The accuracy requirements for the u^1 and u^2 are then less restrictive than in the case described above. Actually, if a tolerance tol is prescribed for the the solution $u(t)$, the approximation u^1 has to be computed within an accuracy

$$
\mathsf{eps} = \frac{1}{8}\,\mathsf{tol}
$$

to obtain a reliable estimate η^1, which can also be used to improve the approximation. In the present case, η^1 and the resulting *second order* approximation u^2 can be computed by

$$
\begin{aligned}
(I - \tau A)\,\eta^1 &= -\frac{1}{2}\tau^2 A f(u(t))\,, \\
u^2 &= u^1 + \eta^1\,.
\end{aligned}
\tag{3.2}
$$

Another important feature of such a discretization (3.1) and correction (3.2) scheme is, that only *one* type of stationary subproblem has to be solved in each global time step, since the equations in (3.1) and (3.2) have the same left-hand side $(I - \tau A)$. If (3.1) is the discretization of a standard ODE system, this implies, that only one matrix decomposition has to be done. Finally we mention, that when a time step with size τ has been performed, a new step size $\bar{\tau}$ can be computed by

$$
\bar{\tau} = \tau \sqrt{\frac{\mathsf{tol}}{\|\eta^1\|}}\,.
$$

4 Numerical Preprocessing

The treatment of countable systems arising from macromolecular reaction kinetics always leads to the computation of infinite sums as a subproblem. For

example, if we want to analyze the m-th statistical moment of the right-hand side in (2.1) (i.e. the time derivative of the m-th moment of the CLD), the evaluation of the following double sum is required (see also [7]):

$$\sum_{s=1}^{\infty} s^m \left(\frac{1}{2} \sum_{r=1}^{s-1} k_{r,s-r}\, u_r\, u_{s-r} - u_s \sum_{r=1}^{\infty} k_{sr}\, u_r \right) \tag{4.1}$$

This term can not be expressed analytically in terms of the moments μ_k for arbitrary coefficients k_{rs}, which prevents the application of a method of moments (for an attempt see [7]). Similar problems seem to appear also in the context of the discrete Galerkin method. However, if a time discretization as described in Section 3 is applied there, the arising sums must not be computed exactly, but may be perturbed in some way fitting to the chosen approximation scheme. This is just fulfilled by a *summation of Gaussian type* [10]. The idea is to replace a sum

$$S = \sum_{s=1}^{\infty} g(s)$$

by an approximation

$$\tilde{S} = \sum_{j=1}^{k} \omega_j\, g(s_j)$$

with *weights* ω_j and *nodes* s_j chosen, such that S can be computed exactly, whenever $g(s)$ is the product of a polynomial of degree $2k-1$ and the weight function $\Psi(s)$ of the used discrete Galerkin method. It is well known from the theory of quadrature, that then the nodes are just the zeros of the associated orthogonal polynomials (which are the modified discrete Laguerre polynomials here). The nodes and weights can be computed easily for a given k by applying the QR-algorithm to a triangular eigenvalue problem, which contains terms from the three-term-recurrence formula of the modified discrete Laguerre polynomials (see [6], Chapter 9.3.). This makes a Gauss summation very efficient, even when the nodes have to be updated very often. In CODEX, $n+1$ nodes are used, when the present Galerkin approximation has n expansion coefficients. The Gauss summation captures exactly the structure of the approach and does not require any truncation of the sums. In the case, where $g(s)$ can be expanded into the associated orthogonal polynomials (times weight function), the summation leads to the numerical values of the analytical terms.

Example. We rewrite (4.1) in the form

$$\sum_{r=1}^{\infty} u_r \sum_{s=1}^{\infty} k_{rs}\, u_s \left(\frac{1}{2} (s+r)^m - r^m \right) \tag{4.2}$$

and apply a double Gauss summation assuming that u_s can be approximated by a Galerkin method with weight function $\Psi_{\rho,\alpha}$ (this does not mean, that u_s must look like $\Psi_{\rho,\alpha}$!). Thus we have to evaluate

$$(4.3) \qquad \sum_{j=1}^{k} \omega_j \frac{u_{s_j}}{\Psi_{\rho,\alpha}(s_j)} \sum_{i=1}^{k} \omega_i \, k_{s_j,s_i} \frac{u_{s_i}}{\Psi_{\rho,\alpha}(s_i)} \left(\frac{1}{2} (s_j + s_i)^m - s_j^m \right)$$

where k is the number of nodes s_j and weights ω_j. For a numerical test we set

$$u_s = \frac{s}{r_{\max}} \bar{\rho}^s \ , \ \bar{\rho} = e^{-1/r_{\max}} \ ,$$

which is a distribution having a maximum at $s = r_{\max}$=100. Then in CODEX the parameters $\rho = 2\bar{\rho}/(1+\bar{\rho})$, $\alpha = 0$ of the weight function $\Psi_{\rho,\alpha}$ are chosen due to a fitting condition. We know, that for $m = 1$ (4.1) is zero and this is exactly revealed by the Gauss summation. The following Table 1 shows the convergence of the summation for the second statistical moment. A reference result has been computed with 200 nodes. A direct summation of a truncated series takes too long (e.g. $s_{\max} = 1500$, 1150 sec. (CPU) on a SPARC 1+, leads to accuracy $8 \cdot 10^{-3}$). For comparison, the Gauss summation with 10 nodes takes about 0.07

nodes k	error
3	$1 \cdot 10^{-1}$
4	$3 \cdot 10^{-2}$
5	$5 \cdot 10^{-3}$
10	$6 \cdot 10^{-4}$
20	$1 \cdot 10^{-4}$
50	$7 \cdot 10^{-6}$

Table 1:
Convergence of Gauss summation, relative error.

sec. (CPU). In the present context, we do not use the above method to compute moments (these can be directly obtained from the Galerkin approximation), but to built up the so-called Galerkin equations, which determine the expansion coefficients of a solution. In the test runs presented in Section 5, a maximum number of $k = 14$ coefficients was necessary. In view of this result, it is easily seen, that the above summation combined with the low accuracy requirements of the multiplicative error correction are the crucial steps to a fast simulation of models as treated here.

5 Numerical Results

The application of the semi-implicit Euler scheme (3.1) in CODEX requires the Frechét derivative $Df(\varphi)(u)$ of $f(u)$ with respect to u at φ, which can be computed pointwise (the time dependency is omitted) by

$$(5.1) \qquad Df(\varphi)(u)(s) = \sum_{r=1}^{s-1} k_{r,s-r}\,\varphi_{s-r}\,u_r - \varphi_s \sum_{r=1}^{\infty} k_{sr}\,u_r - u_s \sum_{r=1}^{\infty} k_{sr}\,\varphi_r \,.$$

In order to perform the Galerkin method, (double) sums of the form

$$\sum_{s=1}^{\infty} l_j(s)\, Df(\varphi)(l_k)(s) \quad , \; j,\, k = 0,\, 1,\, \ldots,\, n\,,$$

have to be evaluated by the summation algorithm as described in Section 4 (with $g(s) = l_j(s) Df(\varphi)(l_k)(s)$). Further details of the implementation are omitted here, we will only discuss some results. It turns out, that the solution $u_s(t)$ (number distribution) at $t = 100$ sec. has a narrow peak for small chain-lengths ($s < 100$, if $s_{\max} \approx 10^4$). This peak is obviously hard to approximate (i.e. time consuming) by a polynomial expansion as used here (see Figure 1). Nevertheless, a relative accuracy of 8-10% can be obtained in moderate computing times (about 50 sec. CPU on a SPARC 1+), which increase strongly for higher accuracies. This is an effect of properties of the basis functions, not a consequence of the used time and operator discretizations. For a better

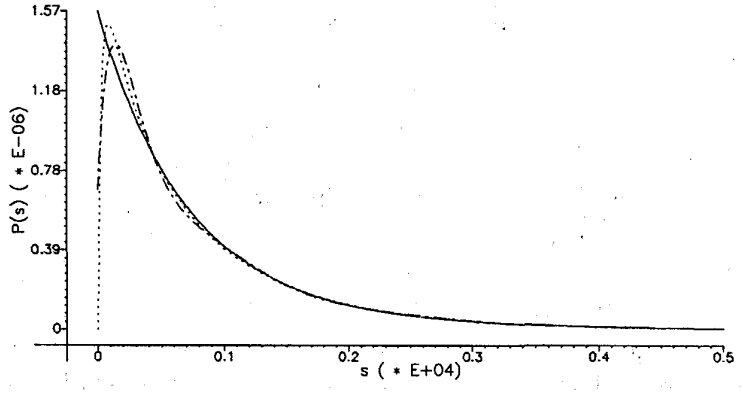


Figure 1: Comparison between direct solution (···) and Galerkin approximations with $n = 4$ (–) and $n = 20$ (-·-) of a heterogeneous coagulation process at t=100.

study of the algorithm for higher accuracies, we compute directly the weight

distribution $u_s(t) \cdot s$ from a transformed equation (2.1). Table 2 shows the performance of CODEX for a simulation up to $t = 100$ sec. in this case. For tol=10^{-2} the computing time can be explained by the fact that more than 10 expansion coefficients are necessary and the respective Gauss summation takes more than 90 % of the total time. If the direct solution at $t = 100$ sec. is directly

tol	time-steps	$n_{\max}$	true error in $H_{\rho,\alpha}$	CPU
10^{-1}	50	5	$1.4 \cdot 10^{-1}$	16
$5 \cdot 10^{-2}$	67	7	$8.5 \cdot 10^{-2}$	49
10^{-2}	135	14	$3.1 \cdot 10^{-2}$	1386

Table 2:
CODEX: performance for several tolerances.

represented by a basis expansion with the parameters obtained by CODEX (i.e. ρ, α and n), the behavior of the time error estimation can be studied. In Table 3 it can be seen, that this device works very accurately. Figure 2

tol	time-error (true)	time-error (estimation)
10^{-1}	$4 \cdot 10^{-2}$	$5 \cdot 10^{-2}$
$5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$
10^{-2}	$9 \cdot 10^{-3}$	$5 \cdot 10^{-3}$

Table 3:
CODEX: Comparison of the time error and its estimation.

shows the time evolution (in logarithmic scale) of the weight distribution up to $t = 100$ sec., showing how fast the interesting range of the chain-length s increases with time. Table 4 compares the computing times of CODEX and a direct integration with the non-stiff ODE-solver DIFEX1 [4] on SPARC 1+ and CRAY-YMP, respectively. As can be seen, the time for the direct integration increases quadratically with the value of $s_{\max}$. Therefore the treatment of such a CODE as a large ODE must be done on a supercomputer in general – with tremendous effort. The computing time of CODEX increases for tolerances up to $5 \cdot 10^{-2}$ in principle with the number of time steps – independent of $s_{\max}$. Figure 3 illustrates the adaptivity of the discrete Galerkin method with respect to the truncation index n for the tolerances 10^{-1}, $5 \cdot 10^{-2}$, 10^{-2}. Because the coagulation process roughens the distribution with time, an increasing number of expansion coefficients is necessary, in particular at the beginning.

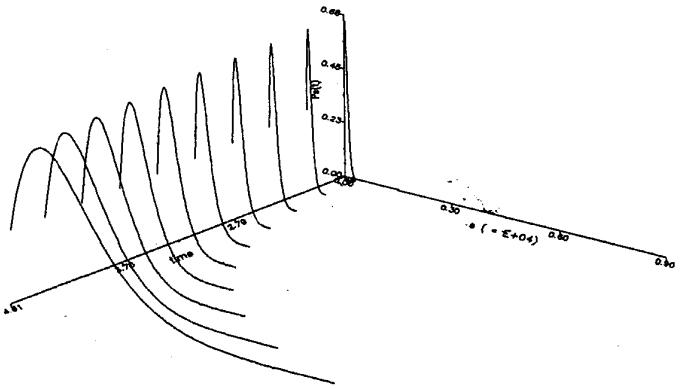


Figure 2: Time evolution of the weight distribution in a soot formation

t	$s_{\max}$	CPU (SPARC 1+) CODEX	CPU (CRAY-YMP) DIFEX1
1.0	50	13	5
2.0	90	19	10
5.0	250	27	25
10.0	600	34	92
50.0	1200	51	441
100.0	8500	59	> 14000

Table 4:
Computing times (sec.) for direct (non-stiff) integration (DIFEX1) on SPARC 1+ and CRAY-YMP and CODEX (tol $= 5 \cdot 10^{-2}$).

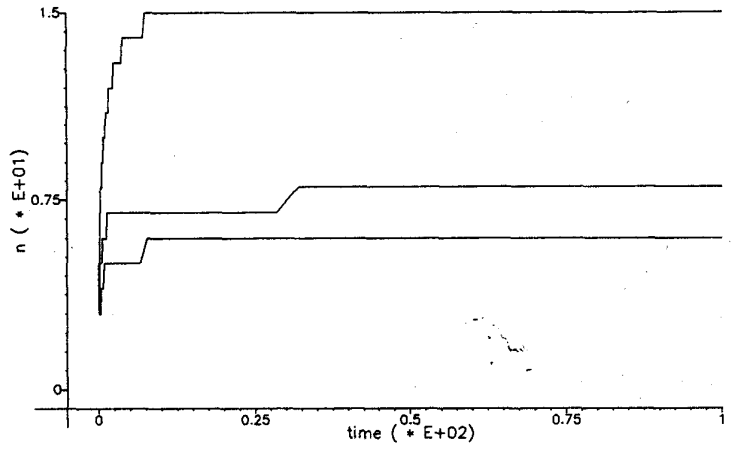


Figure 3: Time evolution of number of expansion coefficients for different tolerances.

We conclude with the remark, that the techniques implemented in CODEX can be extended obviously to problems with combinations of operators as well as to systems of CODE's.

REFERENCES

[1] J. Ackermann, M. Wulkow: MACRON – A Program Package for Makromolekular Reaction Kinetics. Konrad-Zuse-Zentrum SC-90-14 (1990).

[2] F. A. Bornemann: *An Adaptive Multilevel Approach to Parabolic Equations I. General Theory and 1D-Implementation.* IMPACT Comput. Sci. Engrg. 2, 279–317 (1990).

[3] F. A. Bornemann: *An Adaptive Multilevel Approach to Parabolic Equations II. Variable-Order Time Discretization Based on a Multiplicative Error Correction.* IMPACT Comput. Sci. Engrg. 3, 93–122 (1991).

[4] P. Deuflhard: *Recent Progress in Extrapolation Methods for Ordinary Differential Equations.* SIAM Rev. 27 (1985), pp. 505-535.

[5] P. Deuflhard, M. Wulkow: *Computational Treatment of Polyreaction Kinetics by Orthogonal Polynomials of a Discrete Variable.* IMPACT Comput. Sci. Eng., 1 (1989) 269–301.

[6] P. Deuflhard, A. Hohmann: *Numerische Mathematik.*, W. de Gruyter (1991).

[7] M. Frenklach, S. J. Harris: *Aerosol Dynamics Modeling Using the Method of Moments.* J. Colloid Interface Sci. Vol. 118, No. 1 (1987).

[8] H. Gajewski, K. Zacharias: *On an Initial-Value Problem for a Coagulation Equation with Growth Term.* Math. Nachr. 109 (1982), 135–156.

[9] F. D. Magalhaes, M. R. N. Costa : *An Efficient Method for Computing Chain Length Distributions for Chain Length Dependent Reactivity.* in K.-H. Reichert, W. Geiseler: *Polymer Reaction Engineering.* VCH Weinheim (1989).

[10] A. F. Nikiforov, V. B. Uvarov: *Special Functions of Mathematical Physics.* Birkhäuser (1988).

[11] M. Wulkow, J. Ackermann: *Numerical Simulation of Macromolecular Kinetics – Recent Developments.* IUPAC Working Party "Modelling of Polymerization Kinetics and Processes", Macro Group, Frankfurt (1990).

[12] M. Wulkow: *Adaptive Treatment of Polyreactions in Weighted Sequence Spaces.* Konrad–Zuse–Zentrum Berlin, Preprint 91-17 (1991), submitted to IMPACT Comput. Sci. Engng.