Peter Deuflhard  Jörg Ackermann

Adaptive Discrete Galerkin–Methods for Macromolecular Processes

Preprint SC 93–22 (September 1993)
Abstract. In this paper, a rather recent algorithmic approach to the numerical simulation of macromolecular processes is surveyed. It avoids the numerical stiff integration of thousands up to millions of ODE’s by constructing a scale of discrete Hilbert spaces, especially weighted sequence spaces, and establishing a corresponding Galerkin method. Examples including polyreactions of industrial relevance and ecological waste management by biochemical recycling illustrate the importance and efficiency of the algorithm.
Chapter 1

Introduction

Macromolecular processes play an important role both in science and engineering including problems of industrial relevance such as polyreaction kinetics. In spite of their importance, their mathematical and especially computational treatment has been unsatisfactory for quite a time. Typically, the mathematical modelling ends up at a huge system of ordinary differential equations consisting of thousands up to millions of differential equations corresponding to the chain length of the macromolecule under consideration. Direct stiff integration of such systems is prohibitive due to both storage and CPU time limitations. The traditional statistical moment treatment is often unsatisfactory due to the unsolvable problem of how to choose the truncation index so that at least the first few moments are sufficiently accurate. A further approach, the generation of a PDE by making a continuous variable out of the chain length, which is a discrete variable by its very nature, suffers from the fact that it induces ill-posedness for reaction mechanisms of practical interest. In this situation, Deuflhard and Wulkow in [13] had suggested a new method that they called discrete Galerkin method. This method is adaptive in the sense that it either controls or at least monitors the truncation index occurring in the Galerkin approximation.

In Section 2 below, a first idea of the class of problems to be treated is given, in order to exemplify the modelling structure. Next, two variants of the discrete Galerkin method are discussed in some detail. First, in Section 3, a method of lines type approach comparable to the well-known MOL for PDE’s is presented. It is quite easy to understand and allows monitoring the number of degrees of freedom to be used in the Galerkin approximation. Second, in Section 4, a more sophisticated variant is described that has similarities with a Rothe method for PDE’s. This variant permits an efficient time dependent variation of the truncation index according to the required accuracy of the Galerkin approximation. Both variants have been implemented in software that has solved an impressive number of realistic problems from science and engineering. Examples to illustrate the efficiency of the approach are inserted.
Chapter 2

Mathematical Polyreaction Models

The efficient numerical simulation of chemical reactions between polymers is certainly a challenging task in the field of chemical engineering. Polyreaction mechanisms include

- chain addition polymerization, e.g. anionic polymerization or free radical polymerization
- reversible polymerization, which is just the above mechanism but now including the reverse direction as well
- polymer degradation
- coagulation or irreversible polycondensation.

In all of these cases the mathematical modelling on the basis of chemical kinetics laws leads to a system of (countably) infinitely many ordinary differential equations — see e.g. [13]. Only in extremely simplified situations such differential systems can be solved analytically in closed form. Moreover, these processes typically do not arise in an isolated form but in a larger context of further reaction rate equations or thermodynamic equations. Therefore the standard situation is that a numerical solution is inevitable. In order to give a vague first impression of the complexity of such problems in a real life setting, we start with a rather recent example.

Example 1: Biopolymerization. This problem arises in the attempt to recycle waste of synthetic materials in an ecologically satisfactory way — which is certainly an important problem of modern industrial societies. An attractive idea in this context would be to look out for a synthetic material, which is both produced and eaten by bacteria — under different environmental conditions, of course. An example of such a biological recycling is the use of the Alcalignes eutrophus H16–bacteria which use fructose as a chemical basis to produce polyester (PHB) — compare [6]. The different chemical macromolecular reaction steps of production and degradation of PHB can be summarized in the following
Figure 2.1: Example biopolymerization. Schematic cut through bacteria cells (white areas: polyester granules).

combined reaction model

\[
\begin{align*}
E & \xrightarrow{k_a} A \\
A + M & \xrightarrow{k_t} P_1 \\
P_s + M & \xrightarrow{k_p} P_{s+1} \\
P_s & \xrightarrow{k_t} D_s + E \\
D_{s+r} & \xrightarrow{k_d} D_s + D_r
\end{align*}
\]

with \( s, r = 1, 2, \ldots \). Herein \( M \) denotes the monomer (fructose), \( E \) the enzyme, \( A \) the activated enzyme, \( P_s \) the produced or so-called “living” polymer and \( D_s \) the “dead” PHB–polymer of length \( s \). A schematic illustration of the process is given in Fig. 2.1.

In *mathematical* terms the above process can be represented by the following system of ordinary differential equations

\[
\begin{align*}
E' &= -k_a E + k_t \sum_{r=1}^{s_{\text{max}}} P_r \\
A' &= +k_a E - k_t A M \\
M' &= -k_p M \sum_{r=1}^{s_{\text{max}}} P_r - k_t A M \\
P'_1 &= -k_p M P_1 + k_t A M - k_t P_1 \\
P'_s &= -k_p M (P_s - P_{s-1}) - k_d P_s \\
D'_s &= +k_t P_s - k_d (s - 1) D_s + 2k_d \sum_{r=s+1}^{s_{\text{max}}} D_r \\
E' &= -k_a E + k_t \sum_{r=1}^{s_{\text{max}}} P_r \\
A' &= +k_a E - k_t A M \\
M' &= -k_p M \sum_{r=1}^{s_{\text{max}}} P_r - k_t A M \\
P'_1 &= -k_p M P_1 + k_t A M - k_t P_1 \\
P'_s &= -k_p M (P_s - P_{s-1}) - k_d P_s \\
D'_s &= +k_t P_s - k_d (s - 1) D_s + 2k_d \sum_{r=s+1}^{s_{\text{max}}} D_r \\
\end{align*}
\]
The truncation index $s_{\text{max}}$ is not known a priori, but practical considerations lead to the order of magnitude $s_{\text{max}} = 50,000$, which means that the above system consists of 100,000 differential equations!

Standard computational approaches to solve such systems would include

- **Large scale stiff integration** by the sparse mode of any of the more efficient stiff integrators such as the BDF method (Gear, Hindmarsh) or a semi-implicit extrapolation integrator (Deuflhard et al.). However, a prohibitive feature of these systems often is that the Jacobian, which is used in all of these integrators, is not really sparse — which drives the storage and computation time beyond reasonable bounds.

- **Lumping techniques.** In this approach knowledge about details of the chemical process (in isolated form!) is applied to derive a rule for arranging appropriate compartments of polymers to obtain smaller differential systems. This method works nicely in linear models only. Its extension to nonlinear models, however, is more than dubious.

- **Statistical moment treatment.** This rather popular method suffers from the serious drawback that it is not really clear when to truncate the number of moments to be computed — at least in the most frequent case of so-called open loops.

- **Continuous modelling** by partial integro-differential equations of neutron transport type. However, this type of modelling makes the discrete variable $s$ artificially a continuous variable — a mathematical crime which is punished by the occurrence of ill-posed problems in realistic models.

In this situation, Deuflhard and Wulkow had suggested a new approach, which they called discrete Galerkin method. This approach combines the advantages of all of the other approaches and avoids the disadvantages — as described in the initializing paper [13]. Two variants of this method are now given in the subsequent survey.
Chapter 3

Method of Lines Approach

In this section we start from the mathematical model directly. Let $u_s(t)$ denote the concentration of macromolecules of chain length $s$ at time $t$. The sequence $u_1(t), u_2(t), \ldots$ can be written in short hand notation as distribution $u(t) = (u_s(t), s = 1, 2, \ldots)$. As illustrated above, the kinetics of a macromolecular reaction process can be represented by a countable system of ordinary differential equations (abbreviated as CODE) of the form

$$u'_s(t) = (Au(t))_s$$

with given initial distribution $u_s(0)$. For simplicity, the linear special case is treated here with $A$ denoting a linear, possibly unbounded operator. The changes to the general nonlinear case, which in fact is treated in the below mentioned software, are marginal. Starting point for the construction of the discrete Galerkin method in [13] was the introduction of a discrete inner product to take care of the discrete nature of the variable $s$. Formally, it can be written as

$$(f, g) := \sum_{s=1}^{\infty} f(s)g(s)\Psi(s)$$

where $f, g$ are functions of the discrete variable $s = 1, 2, \ldots$ and $\Psi$ is a given (positive) weight function. This product induces an associated norm

$$||f|| := (f, f)^{1/2}$$

and an associated orthogonal basis $\{l_j(s), j = 0, 1, 2, \ldots\}$ of polynomials of the discrete variable $s$ with

$$(l_j, l_k) = \gamma_j \delta_{jk}, \quad \gamma_j > 0 \quad j, k = 0, 1, 2, \ldots. \quad (3.4)$$

With these preparations a natural ansatz for the unknown distribution $u_s(t)$ will be

$$u_s(t) = \Psi(s) \sum_{k=0}^{\infty} a_k(t)l_k(s). \quad (3.5)$$

Truncation of this expansion after $n+1$ terms leads to the Galerkin approximation

$$u_s^{(n)}(t) := \Psi(s) \sum_{k=0}^{n} a_k^{(n)}(t)l_k(s) \quad (3.6)$$

in terms of the truncation index $n$. Note that whenever the system has the already mentioned open loop property, then the upper index $(n)$ in the expansion coefficients $a_k$ cannot be ignored.
The thus characterized approximation has the structure of the method of lines approach in the treatment of PDE’s: first, the space discretization is performed, which leads to a system of ODE’s of fixed dimension. These differential equations are typically stiff, too, and need to be solved by an efficient stiff integrator. The hopeful benefit of this approach, however, can be that the newly arising stiff system has drastically smaller dimension than the original one. In order to achieve this goal, at least intuitively, the Galerkin method suggested by Deuflhard and Wulkow involves two further features, which are crucial for the success of the method. First, the weight function $\Psi$, which will depend on at least one parameter, is normalized such that its zero and first order statistical moments $\nu_0, \nu_1$ coincide with the corresponding moments $\mu_0, \mu_1$ of the unknown distribution $u$ (up to some common scaling factor). As a consequence, the weight function $\Psi$ is then time dependent just as the wanted distribution $u$ — which motivates the name moving weight function in view of the term moving grid within the method of lines for PDE’s. (There also exists the analogue of static regridding, to adapt the free parameters.) Second, in order to monitor the truncation index $n$, the truncation error is estimated. The principle of the truncation error estimator is the same, which is used to monitor the discretization error of an ODE integrator: one just estimates the first neglected term in the expansion. For details the reader may refer to the paper [13]. For the purpose of the present survey, the thus designed adaptive discrete Galerkin procedure will now just be illustrated. For this reason, let us specify the weight function to be the geometric weight function, also called Schulz–Flory distribution in the chemical literature. If we normalize it as described above, it arises as

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

so that

$$\nu_0 = \sum_{s=1}^{\infty} \Psi_\rho(s) = 1 \quad \text{and} \quad \nu_1 = \sum_{s=1}^{\infty} s\Psi_\rho(s) = (1 - \rho)^{-1} = \frac{\mu_1}{\mu_0}$$

in terms of the parameter $\rho$. The associated orthogonal polynomials are the discrete Laguerre polynomials

$$l_k(s) = \rho^k \sum_{\nu=0}^{k} \binom{k}{\nu} \left( \frac{\rho - 1}{\rho} \right)^\nu \left( s - 1 \right)^\nu.$$  

(3.9)

Of course, the above representation is not used for actual computation. Rather, the three term recurrence is exploited. By inserting expansion (3.5) into (3.1), multiplying with the test function $l_j(s)$ and summing over the discrete variable $s$ we obtain

$$\sum_{s=1}^{\infty} \Psi(s) \sum_{k=0}^{\infty} a_k'(t) l_j(s) l_k(s) = \sum_{s=1}^{\infty} \Psi(s) \sum_{k=0}^{\infty} a_k(t) l_j(s)(Al_k(s)).$$  

(3.10)
Changing the order of summations then leads to

\[ \sum_{k=0}^{\infty} a_k'(t)(l_j, l_k) = \Psi(s) \sum_{k=0}^{\infty} a_k(t)(l_j, A_l_k) \quad . \] (3.11)

So the above rather lengthy expression simplifies drastically: by application of the orthogonality relation (3.4) only one term remains on the left side, thus yielding an infinite system of ordinary differential equations for the sequence of expansion coefficients \( a_j(t), j = 0, 1, \ldots \)

\[ \gamma_j a_j'(t) = \Psi(s) \sum_{k=0}^{\infty} a_k(t)(l_j, A_l_k) \quad . \] (3.12)

In the Galerkin method this infinite system is replaced by the appropriate finite system thus defining the coefficients \( a_j^{(n)}(t), j = 0, 1, \ldots, n \). As for the choice of the truncation index \( n \), the truncation error is estimated so that at least the order of magnitude of the error is known, once a decision for a specific \( n \) has been made. In emergency cases, the computation can be just repeated with higher truncation index. Typical values in applications are in the range \( 5 - 10 \). The remaining task is to compute the \( O(n^2) \) terms \( (l_j, A_l_k) \) either analytically or, in complicated mechanisms, numerically by means of an adaptive multigrid summation or a Gauss–Christoffel summation formula based on the weight function \( \Psi \). For a number of interesting mechanisms analytic formulae are known and can be exploited.

For the special case of the Schulz–Flory weight function this sometimes rather lengthy preprocessing can be automated. This is done in the software package MACRON [1]. The package MACRON combines the discrete Galerkin techniques with the formerly developed software package LARKIN due to [12, 2] for the treatment of standard though large scale chemical reaction systems. Only the chemical formalism is needed as input to start the simulation. It includes sophisticated stiff ODE solvers of extrapolation type [11], pointwise evaluation of the Galerkin approximation by a fast summation algorithm, appropriate internal scaling and monitoring by truncation error norms. In order to document the efficiency of the algorithm, a comparative run of the large scale stiff integration of the original ODE system and the small scale ODE system for the Galerkin coefficients is given in Table 1. The underlying chemical problem is a “quasi–living” free radical polymerization problem of technical interest — see [15]. The authors of [15] applied a modern sparse version of Gear’s method. The discrete Galerkin method was used within the package MACRON.

Upon ignoring the fact that the direct stiff integration did not give a comparable accuracy for the desired dispersity, the speed–up factor can be estimated. To do this, the CPU–time of about 16,000 sec with maximum
Table 1: Computation times and computed polydispersity as a function of maximum chain length in direct stiff integration of original ODE system ($t = 10$ s). Reference: Johnson et al. [15].

<table>
<thead>
<tr>
<th>chain length</th>
<th>CPU–time$^a$</th>
<th>dispersity$^b$</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>43</td>
<td>1.98</td>
</tr>
<tr>
<td>100</td>
<td>220</td>
<td>2.07</td>
</tr>
<tr>
<td>200</td>
<td>1.320</td>
<td>2.21</td>
</tr>
<tr>
<td>300</td>
<td>4.048</td>
<td>2.30</td>
</tr>
<tr>
<td>400</td>
<td>7.987</td>
<td>2.33</td>
</tr>
<tr>
<td>500</td>
<td>16.042</td>
<td>2.34</td>
</tr>
</tbody>
</table>

a) CPU seconds on a Cyber 205

b) exact dispersity 2.404 computed in 18.1 s CPU time on a workstation with the discrete Galerkin method ($n = 5$).

chain length 500 on the supercomputer Cyber 205 must be compared with the 18.1 sec (CPU) on the workstation SUN SPARC 1. Based on the LINPACK benchmark results for the matrix order 100x100 and 1000x1000, the MFLOPS rates of a Cyber 205 (2–pipe) vary in the range 17 – 113, whereas the SUN workstation has roughly 1.4. Therefore we obtain an estimated speed-up factor of 10.762–71.537 ! Speed-up factors in the range of $10^3$ up to $10^4$ appeared to be typical also in other applications. With this progress in the algorithmic development, much more complicated models come into the range of tractability. We finally want to present here a rather recent one.

Example 2: Soot formation in flames. This example arises from modelling chemical combustion in flames as studied by Warnatz [17]. Because of the large variety of isomeric structures of the occurring species, thousands of reactions between hundreds of species are easily reached. Therefore the reaction systems need to be generated automatically by a symbolic expert system [10] based on LISP. Part of these reaction systems are macromolecular reaction steps describing the cyclic growth of soot. A detailed description of the soot formation mechanism would be beyond the scope of the present paper — we just refer to the work of Chevalier and Warnatz [9]. The macromolecular reaction steps suggested in [17] for the production of stable aromatic hydrocarbon (symbolized by $P_s$) via intermediates $A_s$ and $B_s$ during ring
growth are characterized by:

\[
\begin{align*}
C_3H_3 & + C_3H_3 \longrightarrow P_s \\
P_s & + H \leftrightarrow A_s + H_2 \\
A_s & + C_2H_2 \leftrightarrow B_s \\
B_s & + C_2H_2 \longrightarrow P_{s+1} + H \\
P_s & + P_t \longrightarrow P_{s+t}
\end{align*}
\]

with \(s, t = 1, 2, \ldots\). The program MACRON enables a further automatic processing of this reaction system (including the large number of standard chemical reactions) and is used to study the soot formation in flat isothermal flames. The results are in satisfactory agreement [17] with the experimental data of Böhm et al. [3]. Studies of a constant pressure flat flame are in progress and different processes of soot formation can be easily modeled, e.g. oxidation of soot.
Chapter 4

Adaptive Rothe Method

The method of lines (MOL) approach to the discrete Galerkin method as described in the preceding section works efficiently in quite a number of realistic applications including those of industrial relevance [7]. Further studies on possible improvements of this variant have been made by Canu and Ray [8]. However, in view of its applicability to a larger class of practical problems, the MOL variant has nevertheless several disadvantages, the most important of which are:

- **Choice of weight function.** The MOL requires an a priori decision about the choice of \( \Psi \), which in turn determines the choice of the basis for the Galerkin approximation. Apart from the Schulz–Flory distribution, which generates the discrete Laguerre polynomials, the paper [13] also mentions the Poisson distribution, which generates the Charlier polynomials. A “wrong” choice of weight function may result in an undesirably large necessary truncation index \( n \).

- **Adaptive control of the truncation index.** In the MOL approach the truncation error is only monitored rather than controlled. In fact, if the truncation error estimate indicates an intolerable error, then the whole run must be repeated with a larger truncation index to be chosen ad hoc.

In order to overcome these drawbacks, Wulkow [18, 19] presented a much improved variant of the discrete Galerkin method, which avoids the above difficulties.

First, he designed a two parameter weight function, which covers both the Schulz–Flory distribution and the Poisson distribution under its roof. It reads

\[
\Psi_{\rho, \alpha}(s) = (1 - \rho)^{1-\alpha} \left( \frac{s - 1 + \alpha}{s - 1} \right)^{\rho^{s-1}} \quad 0 < \rho < 1 \quad \alpha > -1 .
\]

Such a weight function had been constructed earlier by Wulkow and Deuflhard in [20], but there for a different reason with \( |\alpha| < 1 \) only. The associated polynomials are the so-called modified discrete Laguerre polynomials

\[
l_k(s; \rho, \alpha) = \sum_{j=0}^{k} \rho^{k-j} (\rho - 1)^j \binom{k - \alpha}{k - j} \left( \frac{s - 1}{j} \right) .
\]
Of course, in this two parameter case, the three statistical moments $\nu_0, \nu_1, \nu_2$ of the weight function $\Psi_{\rho,\alpha}$ can be forced to be linearly dependent on the three moments $\mu_0, \mu_1, \mu_2$ of the unknown distribution $u$. As a consequence, the parameters $\rho, \alpha$ appear to be time dependent.

Wulkow recognized the basic importance and usefulness of $\Psi_{\rho,\alpha}$ for general $\alpha > -1$, which led him to employ it for introducing a corresponding scale of discrete Hilbert spaces $H_{\rho,\alpha}$. To start with, we slightly redefine the above inner product (3.2) in the form

$$(u,v)_{\rho,\alpha} = \sum_{s=1}^{\infty} u(s)v(s)\Psi_{\rho,\alpha}^{-1}.$$  \hspace{1cm} (4.3)

This induces an associated norm $\| \cdot \|_{\rho,\alpha}$ and the scale of weighted sequence spaces

$$H_{\rho,\alpha} := \{ u \in \mathbb{R}^N \mid \|u\|_{\rho,\alpha} = \sum_{s=1}^{\infty} u(s)^2\Psi_{\rho,\alpha}^{-1} < \infty \}. \hspace{1cm} (4.4)$$

In this notation the solution sequence $u$ to be approximated must satisfy the condition

$$u \in H_{\rho,\alpha}. \hspace{1cm} (4.5)$$

The corresponding embeddings for the scale of spaces are

$$H_{\rho,\alpha} \hookrightarrow H_{\tilde{\rho},\alpha}, \hspace{1cm} 0 < \rho < \tilde{\rho} < 1,$$

and

$$H_{\rho,\alpha} \hookrightarrow H_{\rho,\beta}, \hspace{1cm} -1 < \alpha < \beta.$$  

As it turns out, the parameter $\rho$ is needed to guarantee existence, uniqueness and regularity of the solution, whereas the parameter $\alpha$ characterizes the “distance” to the geometric distribution in terms of $\rho$. In order to understand why not a single Hilbert space is enough to characterize the solvability of a CODE, just consider the operator

$$(A_1u)_s = -(s-1)u_s \hspace{1cm} (4.6)$$

which appears in the above Example 1. It is an easy task to verify that the linear operator $A_1$ is not bounded in $H_{\rho,0}$ for fixed parameter $\rho$, but Lipschitz continuous over the scale of spaces $H_{\rho,0}$ for varying $\rho$ in the sense that

$$\|A_1u\|_{\rho,0} \leq \frac{C}{\epsilon}\|u\|_{\rho-\epsilon,0}, \hspace{1cm} \epsilon > 0. \hspace{1cm} (4.7)$$

As a consequence, nonlinear operators will not be expected to be characterizable via a traditional Lipschitz condition, but rather by an extension of the above property (4.7). This motivates the following theorem due to [18, 19].
Theorem (Wulkow). Consider a sub-scale of $H_\rho$ spaces (with $\alpha$ omitted) for $\rho \in [\rho_0, 1)$, $0 < \rho_0 < 1$. Let $J = [0, T_f] \subset \mathbb{R}$ and assume:

- The operator
  \[ F : J \times H_\rho \longrightarrow H_{\bar{\rho}} \]
  is continuous for $\bar{\rho} > \rho$ and $F(t, u(0)) \in H_{\rho_0}$ on $J$.

- There exists a constant $M$ such that
  \[ \| F(t, u) - F(t, v) \|_{\bar{\rho}} \leq \frac{M}{(\bar{\rho} - \rho)\gamma} \| u - v \|_{\rho}, \quad 0 < \gamma \leq 1, \]
  for $t \in J$, $\bar{\rho} > \rho$ and $u, v \in H_\rho$.

Then for every $\rho \in (\rho_0, 1)$ the initial value problem

\[ u'(t) = F(t, u(t)), \quad u(0) = \varphi \in H_{\rho_0} \]

has a unique solution

\[ u : [0, \delta(\bar{\rho} - \rho)\gamma) \longrightarrow H_{\bar{\rho}}, \]

with $\delta = \min\{T_f, (Md_\gamma)^{-1}\}$. The constant $d_\gamma > 1$ can be computed in concrete cases, e.g.

\[ d_1 = e, \quad d_{1/2} = 2\sqrt{3}/3. \]

In order to overcome the second above mentioned difficulty, Wulkow adopted an adaptive Rothe method with a so-called multiplicative error correction, that Bornemann [4, 5] had designed for parabolic PDE’s. This method, which is theoretically based on analytic semigroup theory [14], also uses a scale of Hilbert spaces (appropriate for the parabolic case, of course). The name “Rothe method” stands for the fact that in this approach time is discretized first, which thus generates a stationary problem on a single Hilbert space. For example, time discretization of the linear CODE (3.1) by the implicit Euler scheme with time step $\tau$ leads to the countable system of algebraic equations

\[ (I - \tau A)\tilde{u}(t + \tau) = u(t). \quad (4.8) \]

Here $\tilde{u}(t + \tau)$ is understood to be the approximation of the solution $u(t + \tau)$. For the in general nonlinear CODE

\[ u'_s(t) = f_s(u_1(t), u_2(t), \ldots), \quad s = 1, 2, \ldots \]

the adaptive Rothe method may be based on the semi-implicit Euler discretization due to [11], which here reads

\[ (I - \tau A)\Delta u = f(u(t)) \]

\[ u(t + \tau) \approx u' = u + \Delta u \quad (4.10) \]
with \( u^I \) an approximation of \( u(t+\tau) \) and \( A \) an approximate Frechet derivative of \( f \) at some not further specified argument \( u \), which is computationally available. Formally, this approach involves solving an ODE in Hilbert space so that well-known devices like stepsize control can be exploited. Such control devices are based on higher order estimates of the error of the computed solution. In the present context, a reliable estimate of the approximation error is the second order correction \( \eta^I \) defined by

\[
(I - \tau A)\eta^I = -\frac{1}{2}\tau^2 Af(u(t)) .
\] (4.11)

Note that (4.9) implies

\[ u'' \approx A f. \]

Of course, this second order correction \( \eta^I \) can simultaneously be used to improve the approximation \( u^I \) to yield

\[ u(t + \tau) \approx u^2 = u^I + \eta^I. \] (4.12)

An estimate \( \bar{\tau} \) for a reasonable local step size will then be

\[
\bar{\tau} = \tau \sqrt{\frac{\text{eps}}{||\eta^I||}} .
\] (4.13)

wherein eps represents the user prescribed accuracy. At this point, the adaptation of the truncation index \( n \) enters the computational scheme, so that a time dependent truncation index \( n(t) \) can be realized without further complication. Crucial, however, for the actual implementation of this algorithmic approach is that the above corrections are not computed from differencing (which would involve possible cancellation of leading digits) but from division by large numbers, which is meant by the phrase “multiplicative error correction”. This feature is indeed realized in both (4.10) and (4.12).

The adaptive Rothe variant of the discrete Galerkin method in combination with the rather general weight function \( \Psi_{\rho,\alpha} \) has been implemented by Wulkow in the C++ program CODEX. This program has been applied successfully to an impressive number of macromolecular reaction systems in different fields of science and industrial engineering [16, 18, 19].
Bibliography


