Konrad-Zuse-Zentrum für Informationstechnik Berlin

Jens Lang   Artur Walter

A Finite Element Method Adaptive in Space and Time for Nonlinear Reaction–Diffusion–Systems

Preprint SC 92–5 (April 1992)

A Finite Element Method Adaptive in Space and Time for Nonlinear Reaction–Diffusion–Systems

JENS LANG   ARTUR WALTER

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

July 10, 1992

ABSTRACT

Large scale combustion simulations show the need for adaptive methods. First, to save computation time and mainly to resolve local and instationary phenomena. In contrast to the widespread method of lines, we look at the reaction–diffusion equations as an abstract Cauchy problem in an appropriate Hilbert space. This means, we first discretize in time, assuming the space problems solved up to a prescribed tolerance. So, we are able to control the space and time error separately in an adaptive approach. The time discretization is done by several adaptive Runge–Kutta methods whereas for the space discretization a finite element method is used. The different behaviour of the proposed approaches are demonstrated on many fundamental examples from ecology, flame propagation, electrodynamics and combustion theory.

Keywords: initial boundary value problem, Rothe–method, adaptive Runge–Kutta method, finite elements, mesh refinement.

AMS CLASSIFICATION: 65J15, 65M30, 65M50

Contents

Introduction

Diverse phenomena in many branches of physical chemistry, biology and electrodynamics are often described by reaction–diffusion processes. Examples include flame propagation in combustion, interactions of mobile populations in ecosystems, and confinement of a plasma column by radiation pressure. These problems are modeled by initial boundary value problems in which the governing parabolic partial differential equations are coupled through a highly nonlinear right hand side, often representing the reaction rate of the chemical components.

The numerical solution of nonlinear partial differential equations (PDEs) is a main subject of current mathematical research. With increasing complexity of the realistic described problems, there is an increasing need of reliable numerical tools to accurately and efficiently solve the phenomena. Adaptive techniques elaborated in recent years automatically change and evolve with the solution. Thus, they are often the only way for providing the computational codes necessary to solve some of these difficult problems.

In this paper we describe the implementation of the classical Rothe method, which is based on the discretization sequence first time then space. The approach used here was inspired by the results of Bornemann [6] for the scalar, linear and selfadjoint case with this method. He introduced a concept to control the error in the space and time discretization simultaneously, starting from a prescribed overall tolerance. Along this line, we reformulate the parabolic initial boundary value problem as an abstract Cauchy problem in an appropriate Hilbert space. Combining the theory of analytic semigroups with the technique of simplified Newton methods we obtain existence and uniqueness results for this Cauchy problem. The close connection of Newton's method in function space and stiff integrators for ordinary differential equations (ODEs) was introduced and first pointed out by Deuflhard [7]. Within this framework we are able to derive adaptive Runge-Kutta methods, which yield variable time steps and possibly variable orders controlled by the problem. The occurring elliptic subproblems are solved by a multilevel finite element method using linear elements in space. In consequence, a space mesh well suited to the problem under consideration is obtained in order to assure an accuracy required by the time discretization. The above mentioned facts enabled us to transmit the previous results to nonlinear systems of parabolic PDEs. The theoretical approach is independent of the space dimension. For solutions possessing sharp moving spatial transitions, like travelling wavefronts or emerging boundary and interior layers, an automatic adjustment of both the space and the time stepsize is likely to be more successful in efficiently resolving critical regions of high spatial and temporal activity. The programming effort of those adaptive methods cannot be overestimated and is a main aspect of their realization. The author's program used for the computations is based on the research code KASTI1 implemented by Bornemann [6] for scalar, linear equations. It was significantly enlarged to handle the nonlinear system case.

Most discretization methods of time-dependent problems for PDEs described in the literature are extensions of the well-known method of lines. First the space variables are discretized on an a priori selected, fixed space mesh, so as to convert the PDE into a usually stiff system of ODEs, which then can be solved very successfully by automatic stiff ODE solvers, however, ignoring the PDE context. An improvement is the moving finite element method developed by Miller and his co–workers [17], where a fixed number of nodes are still used while the nodal positions are computed together with the solution values at the nodes. Such methods require some form of tuning to safely govern the automatic choice of the changing space nodes. Furthermore, it is not clear how to extend the moving techniques from the 1D case to higher space dimensional problems.

Over the past several years the interest in regridding methods, nodes are both added and deleted after several time steps, has rapidly increased. The reader is especially referred to Bieterman and Babuska [4], Adjerid and Flaherty [1], Furzeland, Verwer and Zegeling [10]. In contrast to those, the algorithm proposed by us is equipped with an space–time error control and has almost none tuning parameters. This benefits especially in complicate problems, as is demonstrated in the numerous numerical examples presented.

The contents of the paper reads briefly as follows. In Section 1 we formulate the class of nonlinear problems under consideration. Section 2 deals with the analysis of the semilinear equation connected with the abstract Cauchy problem. Applying the theory of analytic semigroups we give an Newton-type uniqueness theorem. In Section 3 we derive the discretization, adaptive in time an space. Section 4 includes an extensive numerical testing on a set of six practical relevant nonlinear models to show the performance of the different methods used. Our conclusions and a summarizing discussion are contained in Section 5.

1 Problem Formulation

We aim to solve semilinear parabolic initial boundary value problems. In a bounded domain $\Omega \subset \mathbb{R}^n$ with boundary $\partial\Omega = \Gamma_D \bigcup \Gamma_C$, the problem formulation is

$$\frac{\partial u(t,x)}{\partial t} + A(x,\partial)u(t,x) = f(u,t),\ x \in \Omega,\ t \in (0,T]\ , \tag{1.1}$$

$$u(t,\cdot)|_{\Gamma_D} = \xi(t,\cdot),\ t \in (0,T]\ , \tag{1.2}$$

$$C(\cdot,\partial)u(t,\cdot)|_{\Gamma_C} = \zeta(t,\cdot),\ t \in (0,T]\ , \tag{1.3}$$

$$u(0,\cdot) = u_0\ . \tag{1.4}$$

Here, $A(x,\partial)$ denotes an elliptic differential operator of second order, and $C(x,\partial)$ the corresponding Cauchy boundary operator. We put

$$A = A(x,\partial) := -\nabla \cdot (a(x)\nabla) + b(x)\cdot\nabla + c(x),\ x \in \Omega\ ,$$

and

$$C = C(x,\partial) := -n(x)\cdot(a(x)\nabla) - \sigma(x),\ x \in \Gamma_C\ ,$$

where n is the outer unit normal on $\partial\Omega$. Assuming the boundary functions sufficiently smooth, we can, by a simple transformation, achieve that $\xi,\zeta \equiv 0$. Furthermore, with Γ_0 a closed subset of $\partial\Omega$, we introduce a Hilbert space of weak solutions

$$H_D^1(\Omega) = \{u \in H^1(\Omega)|\ u|_{\Gamma_D} = 0\}\ .$$

In the following, we consider the bilinear form $a(\cdot,\cdot)$ on $H_D^1(\Omega)\times H_D^1(\Omega)$:

$$a(u,v) = \int_\Omega (a\nabla u\cdot\nabla v + b\cdot\nabla uv + cuv)dx + \int_{\Gamma_C}\sigma uvds\ ,$$

for all $u,v \in H_D^1(\Omega)$. For further investigation the ellipticity and boundedness of the form $a(\cdot,\cdot)$ will be important. Therefore, we demand

$$|a(u,v)| \le c_1\|u\|_1\|v\|_1 \tag{1.5}$$

and

$$a(u,u) \ge c_2\|u\|_1^2 \tag{1.6}$$

for all $u,v \in H_D^1(\Omega)$, with $c_1,c_2 > 0$. The boundedness of $a(\cdot,\cdot)$ in (1.5) implies the existence of a bounded operator $A : D(A) \subset L^2(\Omega) \to L^2(\Omega)$ which obeys

$$a(u,v) = (Au,v),\ u \in D(A), v \in H_D^1(\Omega)\ . \tag{1.7}$$

The smooth boundary $\partial\Omega$ guarantees, that

$$D(A) = H^2(\Omega) \cap H_0^1(\Omega) \tag{1.8}$$

for the homogeneous Dirichlet boundary condition and

$$D(A) = H^2(\Omega) \cap \{Cu = 0 \text{ on } \partial\Omega\} \tag{1.9}$$

if $\Gamma_C = \partial\Omega$.

2 Analysis of the semilinear equation

The main aspect of the approach done here, is to reformulate problem (1.1) as the following semilinear abstract Cauchy problem in $L^2(\Omega)$

$$\begin{aligned} u_t + Au + Bu &= f(u,t) + Bu =: g(u,t) \\ u(0) &= u_0 \, . \end{aligned} \tag{2.1}$$

In our applications, e.g. Theorem 1 below, the operator $-B$ is an approximation of the Jacobian f_u of f, evaluated at $\bar{t}$ fixed,

$$-B \approx f_u(u(\bar{t}), \bar{t}) \, .$$

So, we assume B to be a linear bounded operator of the form $Bu = d(x)u$. This "formal linearization" extracts the linear part of f and allows a better inside in (1.1). This reformulation is widespread in the derivation of semilinear methods, e.g. [8], [11], [26]. For this, we first look at (1.1)–(1.4) and proceed by perturbation arguments to (2.1). We shall employ the standard technique using the theory of analytic semigroups, e. g. [20]. However, all estimates are to be done more finely. Doing this, we get more clearness about the appearing constants than usually.

Lemma 1. *The operator $-A$, given by $(Au, v) := a(u, v)$ is the infinitesimal generator of an analytic semigroup $T(t)$ of contractions satisfying*

$$\|T(t)\| \le e^{-c_2 t} \, . \tag{2.2}$$

Proof : The Lemma follows from Corollary 1.3.8 and Theorem 7.2.7 in [20, p.12]. The Rayleigh quotient of A gives

$$\operatorname{Re} \lambda_{\min} = \min_{u \in D(A)} \frac{a(u,u)}{\|u\|_0^2} \ge \min_{u \in D(A)} \frac{c_2 \|u\|_0^2}{\|u\|_0^2} = c_2 \, . \tag{2.3}$$

From this, (2.2) follows. ■

In the case $D(A)$ is (1.8) we can improve the estimate (2.3) with Poincare's inequality, which gives $\mathrm{Re}\,\lambda_{\min} \geq c_2(1 + 1/d^2)$. If $D(A)$ is (1.9) , the same argument, namely that $\|\nabla u\|$ is a norm on $D(A)$, gives an improvement, but with unknown constants. So, restriction to a somewhat smaller domain $\omega \subset \Omega$ can improve the coercivity of the operator. Advantage of this can be taken in the following considerations. A nice application are the observations done in the computations of the unsteady moving flame front in Section 4.5. It is defined on $\mathbb{R}$. The dependence of the schemes from the necessary restricted computational domain, as described above, was reported e.g. in [21].

In view of (2.1) and Lemma 1, we assume in addition to (1.5), (1.6)

$$(Bu, u) \geq K_0 \|u\|_0^2 \quad \forall u \in D(A). \tag{2.4}$$

For the operator $H := -(A + B)$ we obtain

$$a(u, u) + (Bu, u) \geq (c_2 + K_0)\|u\|_0^2 \,. \tag{2.5}$$

In consequence, we get

Corollary 1. *The operator $H := -(A + B)$ satisfying (2.5) with $K_0 > -c_2$ generates an analytic semigroup of contractions, i.e.*

$$\|S(t)\| \leq e^{-(K_0+c_2)t} \leq 1 \quad .$$

Remark : In the case $K_0 \leq -c_2$, a remedy is the classical Garding Transformation $v(t) := e^{-K_o t}u(t)$ applied to (2.1). This yields

$$v_t + (A + K_0)v = e^{-K_0 t} f(e^{K_0 t}v, t) \tag{2.6}$$

and the operator $-(A + K_0 I)$ generates a semigroup of contractions and

$$\|S(t)\| \leq e^{-c_2 t} \,. \tag{2.7}$$

Now, we give a Newton–type uniqueness theorem for equation (1.1). It was first introduced by Deuflhard [7] for ODE's. With the help of the semigroup theory, we can use his arguments in a similar way. The estimates of the exponential function in the ODE–case are replaced by the above derived estimates of the generators of a semigroup. The given exposition served to clarify the occuring constants. During the sketched proof of Theorem 1, equations of the form (2.1) occur and can be handled by (2.6) to reensure the bounds (1.5) and (1.6) for the perturbed operator. For clearness of presentation and without loss of generality, we restrict ourselves to the autonomous case $f(u(t),t) = f(u(t))$.

THEOREM 1. *(extension of Theorem 1, Deuflhard [7]) For (1.1) we assume that $f \in C^1(D(A))$ and $u_0 \in D(A)$. The function $f(u) - Au$ in (1.1) should be bounded*

$$\|f(u_0) - Au_0\| \leq L_0 \tag{2.8}$$

and it's Jacobian should be Lipschitz continuous

$$\|f_u(v) - f_u(w)\| \leq L_1\|v - w\| \quad \forall v, w \in L^2(\Omega) \ . \tag{2.9}$$

Then, existence and uniqueness of the solution of (1.1) is guaranteed in $[0, \tau]$ *for*

$$\tau \text{ unbounded, if } \mu\bar{\tau} \leq -1 \tag{2.10}$$

$$\tau \leq \bar{\tau}\psi(\mu\bar{\tau}) \ , \text{ if } \mu\bar{\tau} > -1 \ , \tag{2.11}$$

where $\bar{\tau} := (2L_0L_1)^{-\frac{1}{2}}$, $\mu := -(c_2 + K_0)$ *and*

$$\psi(s) := \begin{cases} \frac{\ln\,(1+s)}{s} & s \neq 0 \\ 1 & s = 0 \ . \end{cases} \tag{2.12}$$

Proof : The main point in the proof is to solve the variation of constants formula in semigroup context,[20, p.148] for (1.1)

$$F(u) = u(t) - T(t)u_0 - \int_0^t T(t-s)f(u(s))ds = 0 \tag{2.13}$$

by Newton's–Method

$$\begin{aligned} u^{i+1}(t) \ &+ \ \textstyle\int_0^t T(t-s)Bu^{i+1}(s)ds = T(t)u_0 + \\ &+ \ \textstyle\int_0^t T(t-s)\{f(u^i(s)) - Bu^i(s)\}ds \ , \end{aligned} \tag{2.14}$$

where $B := f_u(u^i)$. The rest of the proof is to guarantee that (2.14) is a contraction and can be copied from [7]. This will ensure the convergence of Newton's method to a unique solution of (2.13). ∎

Remark : A similar result can be obtained, starting with the Voltera–equation

$$F(u) = u(t) - u_0 - \int_0^t f(u(s))ds = 0$$

instead of (2.13), e.g. [20]. However, representation (2.13) fits better in our derivation of the numerical scheme.

The above existence results are obtained by using Newton's method for the formula (2.13). If (1.1) represents a system of equations, a great simplification in numerical calculations can be achieved, using an approximation B of the Jacobian of $-f(u)$ occuring in the Newton method (2.14). The influence of this approximation on the existence interval (2.10), (2.11) of (1.1) is shown in the following Corollary.

COROLLARY 2. *(extension of Theorem 2 of Deuflhard [7]) With all assumptions of Theorem 1 and in addition for the perturbation of the Jacobian*

$$\|B + f_u(u_0)\| \leq \delta_0 \text{ and } (Bu, u) \geq K_0 \|u\|_0^2 \tag{2.15}$$

with $K_0 > -c_2$,
fulfilled, the bounds (2.10), (2.11) remain valid with $\bar{\tau}$ *replaced by*

$$\hat{\tau} := \frac{\bar{\tau}}{1 + \delta_0 \bar{\tau}} .$$

Proof : Using standard semigroup techniques, the same arguments as in [7] can be used to proof the Corollary. ■

3 Derivation of the Discretizations

3.1 Time discretization in Hilbert space

For further investigation, it is convenient to use again the abstract Cauchy problem in $L^2(\Omega)$ (2.1)

$$u_t + Au + Bu = f(u,t) + Bu =: g(u,t) \tag{3.1}$$

$$u(0) = u_0 \tag{3.2}$$

By perturbation theory, the operator $H := -(A + B)$ generates an analytic semigroup $S(tH) := \exp(tH)$. In contrast to the preceding chapters we use the operator H explicitly as an argument of the semigroup. For the derivation of the schemes, it's not necessary for $S(tH)$ to be contractive.

The formal solution of (3.1) is given by

$$u(\tau) = S(\tau H)u_0 + \int_0^\tau S((\tau - s)H)g(u(s), s)ds,\ \tau \in (0, T]\ . \tag{3.3}$$

As done in the existence theorem above, we start from (3.3) to construct a numerical scheme for (3.1). For this, we will consider adaptive Runge–Kutta methods, which are included in the class of semi–implicit methods. They have been extensively studied by Strehmel and Weiner [26], see also references therein.

First, let $R_0^{(i)}(z)$ be a rational approximation of sufficiently high order r_i to $\exp(z)$, this means

$$S(z) = R_0^{(i)}(z) + O(z^{r_i+1}) \text{ for } z \to 0,\ z \in \mathbb{C}\ . \tag{3.4}$$

The approximation $R_0^{(i)}$ is said to be

- strongly A–stable if
$$|R_0^{(i)}(z)| < 1 \text{ for } Re(z) < 0, \quad \text{and } |R_0^{(i)}(-\infty)| < 1;$$
- L–stable if it is strongly A–stable with
$$R_0^{(i)}(-\infty) = 0\ .$$

Furthermore, we define

$$\begin{aligned} R_1^{(i)}(z) &:= (R_0^{(i)}(z) - I)\frac{1}{z}\ , \\ R_{l+1}^{(i)}(z) &:= (lR_l^{(i)}(z) - I)\frac{1}{z},\ l = 1, 2, \ldots \end{aligned} \tag{3.5}$$

We note that the functions $R_l^{(i)}$ possess the same denominator as $R_0^{(i)}$, such that the factor z^{-1} cancels out. So, no problem occurs, if in the following H becomes singular. The numerical analogue of the abstract solution formula (3.3) via the adaptive s–stage Runge–Kutta method is given by

$$\begin{aligned} u^{(1)} &:= u_0 \\ u^{(i)} &:= R_0^{(i)}(c_i\tau H)u_0 + \tau \textstyle\sum_{j=1}^{i-1} A_{ij}g_j,\ i = 2, 3, \ldots, s+1 \\ u_1 &:= u^{(s+1)} \end{aligned} \tag{3.6}$$

with

$$A_{ij} = \sum_{l=0}^{p_i} R_{l+1}^{(i)}(c_i \tau H) \lambda_{lj}^{(i)} c_i^{l+1} \; .$$

Here, $g_j := g(u^{(j)}, c_j\tau)$ where $u^{(j)}$ describes an approximation of the solution u at $t = c_j\tau$. The vector $c := (0, c_2, \ldots, c_s, 1)$ satisfies the condition $0 < c_i \leq 1, \; i = 2, \ldots, s$. The constants $\lambda_{lj}^{(i)}$ and p_i determine the method.

Remark : During the construction of such methods the first step is to approximate the function $g(u(t), t)$ from (3.3) in each stage $i = 2, 3, \ldots, s+1$ by a polynomial in t. By partial integration and approximation of the exponential function afterwards, we arrive at (3.6).

In our context, we must solve an elliptic problem for each stage . Therefore, we restrict ourselves to 1– and 2–stage methods and set $c_2 = c_3 = 1$.

1. In the case $s = 1$ we get with $\lambda_{01}^{(2)} = 1$ the so–called adaptive Euler method

   (3.7) $$u^{(2)} = R_0^{(2)}(\tau H) u_0 + \tau R_1^{(2)}(\tau H) g_1 \; .$$

2. For $s = 2$ we get the following one–parameter scheme:

   (3.8) $$\begin{aligned} u^{(1)} &= u_0 \\ u^{(2)} &= R_0^{(2)} u_0 + \tau R_1^{(2)} g_1 \\ u^{(3)} &= R_0^{(3)} u_0 + \tau \{ (R_1^{(3)} \kappa + R_2^{(3)} (1 - 2\kappa)) g_1 + \\ &\quad + (R_1^{(3)} (1 - \kappa) + R_2^{(3)} (2\kappa - 1)) g_2 \} \; , \end{aligned}$$

   where $\kappa \in R$ and for simplicity $R_l^{(i)} := R_l^{(i)}(\tau H)$.

Our aim is to derive a method which minimizes the demand of storage and the effort of work. To this end, we look for an embedded formula

(3.9) $$u^{(3)} = u^{(2)} + \eta_2 \; ,$$

where η_2 can be directly computed. In order to achieve such a structure, first we write the rational approximation $R_0^{(3)}(z)$ to e^z as

(3.10) $$R_0^{(3)}(z) = R_0^{(2)}(z) + Q_0^{(2)}(z) \; .$$

We want to compute the correction term η_2 in the additive representation (3.9). We start with the semi–implicit Euler method first introduced by

Deuflhard [8],

$$
\begin{aligned}
u^{(1)} &= u_0 \\
u^{(2)} &= R_0^{(2)}(z)u_0 + \tau R_1^{(2)}(z)g_1
\end{aligned}
\tag{3.11}
$$

where $R_0^{(2)}(z) = \frac{1}{1-z}$. This method has the consistency order $p = 1$. Once chosen the denominator to be a part of $(1-z)$, we can obtain only two approximations by (3.10), namely

$$
\begin{aligned}
Q_0^{(2)}(z) &:= Q_A^{(2)}(z) = -\frac{1}{2}\frac{z^2}{(1-z)^2} \quad , \\
Q_0^{(2)}(z) &:= Q_L^{(2)}(z) = -\frac{1}{2}\frac{z^2}{(1-z)^3} \quad ,
\end{aligned}
\tag{3.12}
$$

to get a strongly A–stable or L–stable method with minimal approximation order $r_3 \geq 2$, [25]. Applying the adaptive Euler method (3.7) in the third stage we find

$$
\begin{aligned}
u^{(3)} &= R_0^{(3)}(z)u_0 + \tau R_1^{(3)}(z)g_1 \\
&= \left(R_0^{(2)}(z) + Q_0^{(2)}(z)\right) u_0 + \tau\left(R_1^{(2)}(z) + Q_0^{(2)}(z)\cdot \tfrac{1}{z}\right) g_1 \ .
\end{aligned}
\tag{3.13}
$$

Therefore

$$
\eta_2^{(1)} := Q_0^{(2)}(z)\left(u_0 + \frac{\tau}{z}g_1\right) \ .
\tag{3.14}
$$

We remark, that this method can be described as a special 2–stage Rosenbrock method (ROW–method) of consistency order $p = 2$ when the exact Jacobian $B := -f_u(u_0)$ is used, [13].
Further, for the method (3.8) we get

$$
\begin{aligned}
u^{(3)} &= R_0^{(3)}(z)u_0 + \tau\left\{(R_2^{(3)}(z)(1-2\kappa))g_1 + \right. \\
&\quad \left. +(R_1^{(3)}(z)(1-\kappa) + R_2^{(3)}(z)(2\kappa-1)g_2\right\} \\
&= R_0^{(3)}(z)u_0 + \tau\left(R_1^{(3)}(z)(\kappa-1) + R_2^{(3)}(z)(1-2\kappa)\right)(g_1-g_2) + \\
&\quad +\tau R_1^{(3)}(z)g_1 \\
&= u^{(2)} + Q_0^{(2)}(z)\left(u_0 + \frac{\tau}{z}g_1\right) + \\
&\quad +\tau\left(R_1^{(3)}(z)(\kappa-1) + R_2^{(3)}(z)(1-2\kappa)\right)(g_1-g_2) \ .
\end{aligned}
\tag{3.15}
$$

We are interested in two special cases : $\kappa = 0$ leads to a method of order 2 with

$$
\eta_2^{(2)} := \eta_2^{(1)} + \tau(R_2^{(3)}(z) - R_1^{(3)}(z))(g_1 - g_2)
\tag{3.16}
$$

and $\kappa = 1$ proposed by [26] yields

$$\eta_2^{(3)} := \eta_2^{(1)} - \tau R_2^{(3)}(z)(g_1 - g_2) \; , \tag{3.17}$$

where again order 2 is achieved. Both methods only use an approximation of the Jacobian, i.e. $B \approx -f_u(u_0)$, and with this, they are a W–method.

Remark : In the context of PDEs, in each time step, we have to solve a large linear system. By backward error analysis, the inevitable errors can be seen as a perturbation of the exact Jacobian $f_u(u_0)$. At first sight, classical Rosenbrock methods seem therefor unappropriate. Nevertheless, we have tested the adaptive Euler method in the numerical applications with reasonable success.

3.2 Derivation of the Schemes from Newton's Method

Throughout this chapter, we again restrict ourselves to the autonomous case. We start with the abstract solution formula (3.3) of problem formulation (2.1).

$$u(t) = S(t)u_0 + \int_0^t S(t-s)g(u(s))ds \tag{3.18}$$

Again, $S(t)$ is the semigroup generated by $-(A+B)$, where $-B$ is some approximation of the Jacobian.

The scheme is derived by a polynomial approximation in s of $g(u(s))$, a rational approximation of the semigroup $S(t)$ and finally, one step of a simplified Newton method. During our approach to the 1– and 2–stage formulas, which are used in the calculations, we try to approximate $g(u(s))$ by a constant and linear polynomial P in s respectively, depending on u_0 and $u(\tau)$. A first simple approximation in $[0,\tau]$ reads

$$P_0(s) = g(u(\tau)) \quad . \tag{3.19}$$

In consequence, we get in (3.18)

$$\begin{aligned}
&S(\tau)u_0 &+& \int_0^\tau S((\tau-s)g(u(s))ds && (3.20)\\
&\approx S(\tau)u_0 &+& \int_0^\tau S((\tau-s)dsg(u(\tau)) && (3.21)\\
&= S(\tau)u_0 &+& (S(\tau)-I)H^{-1}g(u(\tau)) && (3.22)\\
&\approx R_0^{(j)}u_0 &+& \tau R_1^{(j)}g(u(\tau)) \quad . && (3.23)
\end{aligned}$$

The resulting implicit equation

$$F_1(u) := u - R_0^{(j)} u_0 - \tau R_1^{(j)} g(u) = 0 \tag{3.24}$$

is solved by one step of a simplified Newton method. It is started with the only information available at this moment $u^0 = u_0$. Using the identity as approximate Jacobian of (3.24) we end up with

$$u^1(\tau) = R_0^{(j)} u_0 + \tau \, R_1^{(j)} g(u_0) \, . \tag{3.25}$$

The derived algorithm is called adaptive Euler scheme and is identical to our first method (3.7). Let us now approximate $g(u(s))$ by a linear polynomial P in s

$$P(s) = -\frac{s}{\tau}(g(u_0) - g(u(\tau))) + g(u_0) \quad . \tag{3.26}$$

We look at P in the simplest case $S(t) \approx I$ in (3.18). Of course, for this P the trapezoidal rule is exact

$$\int_0^\tau P(s)ds = \frac{\tau}{2}(g(u_0) + g(u(\tau))) \, . \tag{3.27}$$

We introduce for the end points the parameter κ, such that

$$\begin{aligned} P(0) &= \kappa g(u_0) + (1-\kappa) g(u(\tau)) \\ P(\tau) &= \kappa g(u(\tau)) + (1-\kappa) g(u_0) \, . \end{aligned} \tag{3.28}$$

In consequence, we get instead of (3.26) the new linear polynomial

$$P_1(s) = \frac{s}{\tau}(1-2\kappa)\,(g(u_0) - g(u(\tau))) + \kappa g(u_0) + (1-\kappa) g(u(\tau)) \, , \tag{3.29}$$

which also satisfies (3.27). Replacing $g(u(s))$ by $P(s)$ in (3.18), we get analogously to (3.24) the implicit equation

$$\begin{aligned} F_2(u) \;:=\; & u - R_0^{(j)} u_0 - \tau\{(\kappa R_1^{(j)} + (1-2\kappa) R_2^{(j)}) g(u_0) + \\ & ((1-\kappa) R_1^{(j)} + (2\kappa - 1) R_2^{(j)}) g(u)\} \, . \end{aligned}$$

The approximations $R_1^{(j)}$ and $R_2^{(j)}$ are obtained by partial integration and the recurrence relation (3.5). Afterwards, a step of a simplified Newton method is done starting with u^1 and the identity as Jacobian. Thus, we get the second stage of our adaptive Runge–Kutta scheme (3.8)

$$\begin{aligned} u^2(t) = \; & R_0^{(j)} u_0 + t\{(\kappa R_1^{(j)} + (1-2\kappa) R_2^{(j)}) g(u_0) + \\ & ((1-\kappa) R_1^{(j)} + (2\kappa - 1) R_2^{(j)}) g(u^1)\} \, . \end{aligned}$$

3.3 Stepsize Bounds for the Discretizations

The results of the previous sections are now used to derive stepsize estimates for the proposed schemes. The idea, introduced by Deuflhard in [7], is to interpret the derived algorithms as one step of a simplified Newton method. The analysis for the semi–implicit Euler scheme can be carried over to our adaptive Runge-Kutta schemes. For this, we will apply only linear analysis for the restriction of the feasible time step τ, which becomes more clear during the proof of the next theorem.
To characterize our several methods we define the following special stability functions:

$$R_{0,L}^{(2)}(z) := \frac{1}{1-z} \tag{3.30}$$

$$R_{0,A}^{(3)}(z) := R_{0,L}^{(2)}(z) + Q_A^{(2)}(z) = \frac{1 - z - \frac{1}{2}z^2}{(1-z)^2} \tag{3.31}$$

$$R_{0,L}^{(3)}(z) := R_{0,L}^{(2)}(z) + Q_L^{(2)}(z) = \frac{1 - 2z + \frac{1}{2}z^2}{(1-z)^3} \tag{3.32}$$

Theorem 2. *We assume that $f \in C^1(D(A))$ and $u_0 \in D(A)$. The function $f(u) - Au$ should be bounded*

$$\|f(u^{(j)}) - Au^{(j)}\| \le L_0 \quad , \; j = 1,2, \tag{3.33}$$

and it's Jacobian should be Lipschitz continuous

$$\|f_u(v) - f_u(w)\| \le L_1 \|v - w\| \quad \forall v, w \in L_2(\Omega) \, . \tag{3.34}$$

For the perturbation of the Jacobian we assume

$$\|B + f_u(u^{(j)})\| < \delta_0 \quad , \; j = 1,2. \tag{3.35}$$

*Then, existence and uniqueness of the solutions of the above studied 1– and 2–stage adaptive Runge–Kutta–methods is guaranteed as follows:
The methods (3.7), (3.8) converge to unique solutions $u^{(2)}$ and $u^{(3)}$, if*

$$\tau \text{ unbounded, for } \mu\hat{\tau} \le \alpha(R_0^{(2)}, R_0^{(3)}, \kappa), \tag{3.36}$$

$$\tau \le \frac{\hat{\tau}}{\mu\hat{\tau} - \alpha} \text{ , for } \mu\hat{\tau} > \alpha(R_0^{(2)}, R_0^{(3)}, \kappa) \, , \tag{3.37}$$

with $\hat{\tau} := \bar{\tau}/(1+\beta(R_0^{(2)}, R_0^{(3)}, \kappa)\delta_0\bar{\tau})$, *where*

$$\begin{aligned} \alpha(R_{0,A}^{(3)}) &= \alpha(R_{0,L}^{(3)}) = -2 \quad , \\ \beta(R_{0,A}^{(3)}) &= \beta(R_{0,L}^{(3)}) = 1 \end{aligned} \tag{3.38}$$

$$\begin{aligned} \alpha(R_{0,L}^{(2)}, R_{0,A}^{(3)}, 0) &= \alpha(R_{0,L}^{(2)}, R_{0,L}^{(3)}, 0) = -\frac{1}{2}\sqrt{6} \quad , \\ \beta(R_{0,A}^{(3)}) &= \beta(R_{0,L}^{(3)}) = \frac{1}{6}\sqrt{6} \end{aligned} \tag{3.39}$$

$$\begin{aligned} \alpha(R_{0,L}^{(2)}, R_{0,A}^{(3)}, 1) &= -\frac{1}{2}\sqrt{42} \quad , \\ \beta(R_{0,L}^{(2)}, R_{0,A}^{(3)}, 1) &= \frac{1}{14}\sqrt{42} \end{aligned} \tag{3.40}$$

$$\begin{aligned} \alpha(R_{0,L}^{(2)}, R_{0,L}^{(3)}, 1) &= -\frac{3}{2}\sqrt{10} \quad , \\ \beta(R_{0,L}^{(2)}, R_{0,L}^{(3)}, 1) &= \frac{1}{10}\sqrt{10}. \end{aligned} \tag{3.41}$$

Proof : The general approach proving the Theorem is demonstrated by the example of the adaptive Euler scheme. According to (3.24) we make the identification

$$F(u) := u - R_0^{(3)}u_0 - \tau R_1^{(3)}g(u) \quad , \tag{3.42}$$

$$C := I \approx F_u(u^{(1)}) = I - \tau R_1^{(3)}(f_u(u^{(1)}) + B) \quad . \tag{3.43}$$

As in [7] we have to check

$$\|u^{(3)} - u^{(1)}\| \le \frac{2}{1-\tau\mu}\tau L_0 =: \gamma \quad , \tag{3.44}$$

$$\|C^{-1}(F_u(x) - F_u(y))\| \le \tau L_1 \|R_1^{(3)}\| \le \frac{2}{1-\tau\mu}\tau L_1 =: \omega \quad , \tag{3.45}$$

and

$$\|C^{-1}(C - F_u(u^{(1)}))\| \le \tau\delta_0\|R_1^{(3)}\| \le \frac{2\tau\delta_0}{1-\mu\tau} := \bar{\delta}_0 < 1 \quad . \tag{3.46}$$

It is easy to see, that the inequalities remain unchanged for $R_0^{(3)} = R_{0,A}^{(3)}$ or $R_0^{(3)} = R_{0,L}^{(3)}$. The modified Kantorovitch condition

$$\frac{\gamma \cdot \omega}{(1-\bar{\delta}_0)^2} < \frac{1}{2}$$

yields

$$\frac{2\tau}{1-\mu\tau} \leq \bar{\tau}(1 - \frac{2\tau\delta_0}{1-\mu\tau}) \ ,$$

which is equivalent to

$$\tau < \frac{\hat{\tau}}{2+\mu\hat{\tau}}$$

with $\hat{\tau} := \bar{\tau}/(1+\delta_0\bar{\tau})$. Thus, (3.38) holds true.
For the 2–stage schemes, two independent simplified Newton methods must be studied. Selecting the stronger restriction of τ, exactly the same arguments as in the proof for the adaptive Euler scheme show, that (3.39)–(3.41) is also true. ∎

3.4 Stepsize Selection

Following our derivation, we get for the above described methods the simple error estimates

$$\varepsilon_2^{(i)} := \|u^{(3)} - u^{(2)}\|_0 = \|\eta_2^{(i)}\|_0, \ i = 1,2,3 \ . \tag{3.47}$$

The semi–implicit Euler method has the order 1, i.e.

$$\|u(\tau) - u^{(2)}\|_0 \leq c\tau^2 \ . \tag{3.48}$$

Hence, the corresponding new time steps are

$$\tau_2^{(i)} = \sqrt{\frac{\mathrm{TOL}}{\varepsilon_2^{(i)}}}\tau, \ i = 1,2,3 \ . \tag{3.49}$$

3.5 Computation of the Stability Bounds

In Theorem 2 we obtained a stepsize bound due to the stability criterion (3.36), (3.37). These bounds depend on the quantities μ and $\hat{\tau}$. The calculation of $\hat{\tau}$ is already too difficult, even in the case of ODE's, and seems too costly compared with the information gained. So, we restrict ourselves to an estimation of μ to check the overall assumption $\mu\tau > -1$.

In all proposed schemes, we have to solve the linear system

$$(I - \tau H)y = \tau(f(u^{(1)}) - Au^{(1)}) \tag{3.50}$$

with $y := u^{(2)} - u^{(1)}$.

Equation (3.50) is looked at as the first step of an inverse Power Method [12], with shift $\frac{1}{\tau}$:

$$(\frac{1}{\tau}I - H)(\tau y) = \tau(f(u^{(1)}) - Au^{(1)}) .$$

Since the occurrence of a negative eigenvalue μ_1 causes a step size reduction, we can assume, that after the shift, μ_1 is the eigenvalue with smallest absolute value. For the calculation of

$$\frac{1}{\lambda} = \frac{y^T(I - \tau H)^{-1}y}{y^T y}$$

an additional solution of a linear system (3.50) is needed. It can be shown by the Bauer–Fike Theory for special cases of H and numerical experiments have confirmed, that 2–4 SSOR steps are enough for the estimation of

$$\mu_1 = \frac{1}{\lambda\tau} - \frac{1}{\tau} .$$

As proposed in [12], we take the first two iterates of the inverse power method to get it's deviation

$$d = y^{(2)} - cy^{(1)}$$

with

$$c := \frac{\text{sign}(y^{(2)})_i}{\text{sign}(y^{(1)})_i}, \quad |(y^{(2)})_i| = \|y^{(2)}\|_\infty.$$

In the code, proposed stepsizes τ are rejected as soon as

$$\mu_1\tau > c \cdot 1$$

with an appropriate chosen safety factor c, usually between 0.5 and 0.9.

3.6 Space Discretization

During the realization of one time step of the adaptive Runge–Kutta method (3.6) we have to approximate the arising elliptic problems. They appear in two forms

$$u = (I - \tau H)^{-1}w, \ w \in L^2(\Omega)$$

and

$$u = (I - \tau H)^{-1}\tau Hw, \ w \in L^2(\Omega) .$$

Analogous to (1.7) $-H$ is the weak representation of the elliptic operator $-H(x,\partial) = A(x,\partial) + B(x)$. Therefore, the above problems are equivalent to the variational problems

$$(3.51) \qquad \begin{aligned} &(u,v) + \tau h(u,v) = (w,v), \quad \text{for all } v \in H^1_D(\Omega) \\ \text{and} \quad & \\ &(u,v) + \tau h(u,v) = -\tau h(w,v), \quad \text{for all } v \in H^1_D(\Omega)\ , \end{aligned}$$

where $h(\cdot,\cdot)$ denotes the bilinear form associated with $-H$.

For our purpose, the elliptic solver has to solve (3.51) within a given accuracy and return a suitable error estimation. From this viewpoint, it is reasonable to use an adaptive FEM–method consisting of three modules: linear solver, error estimator and refinement strategy.

For the solution of (3.51) we applied the finite element method with linear basis functions. Using localization principles, we solve on each finite element the same elliptic problem with imposing the current FEM–approximation as Dirichlet boundary condition, [2]. These problems are not solved exactly, but applying quadratic finite elements, improving the solution local inside of each element. In consequence, we get a reasonable τ–independent local error estimator as shown by [5].

Finally we get our computable local error estimator

$$(3.52) \qquad [\delta_j] \quad , \ j = 1, \ldots, \mathrm{n} := \text{number of elements}$$

and the corresponding global one

$$(3.53) \qquad [\delta] := \left(\sum_{j=1}^{n} [\delta_j]^2 \right)^{\frac{1}{2}} \ .$$

Since we are equipped with local error indicators $[\delta_j]$, the obvious idea is to improve the finite element solution until a given tolerance eps in an adaptive process through equidistribution of all element errors. We search for a refinement strategy

$$\text{Refine } I_j \text{ if } [\delta_j] > \ \text{cut} \ \ .$$

In order to achieve this, we determine "cut" by

$$\text{cut} \ := \frac{c}{n} \sum_{j=1}^{n} [\delta_j]$$

with some constant c which guarantees a desirable rate of refinement. In practical computations 30% – 50% are usually refined.

Some remarks about the linear solver remain. Clearly, in the scalar, 1–D case the stiffness–matrix is only tridiagonal and the linear equation can be numerically stable solved by direct Gauss–elimination without pivoting in $\mathcal{O}(n)$ operations, [14], and only the local stiffness–matrices are assembled.

In the system case and higher space dimensions, the number of unknowns quickly become very large such that solving in direct manner isn't possible. Hence, an iterative solver must be applied. We have used a block version of the SSOR–algorithm, [12]. To this end, numbering of the unknowns in nodal sequence to concentrate more information about the mutual influence of the system components in the direct nearness of the matrix diagonal is essential. In each refinement level, we solve for all ingredients of the algorithm, this means for $u^{(2)}$, η_2. According to the multilevel structure we get in a natural way good starting values for each elliptic subproblem by interpolation from coarse to fine grid. It benefits to couple the prescribed tolerance with a prescribed error in the linear solver. In the numerical examples, we used an infinity norm of the corrections of 10^{-4} as a stopping criteria. This corresponds to ≈ 10 iteration steps of the linear solver. One should be aware, that the linear solver is the most time–consuming part of the method and needs special attention in higher dimensions.

3.7 Matching of Spatial Errors

Starting from a perturbation concept introduced by Bornemann [6] for scalar, linear selfadjoint problems, we derive rules to handle the spatial errors. We want to extend this concept to the nonlinear case. More precisely: We have to determine a relation between the accuracy *eps* of the elliptic solver and the parabolic accuracy TOL.

Computation of each $u^{(j)}$ in (3.6) requires the solution of several elliptic problems considered in the last chapter. In general, we get perturbed solutions

$$\hat{u}^{(j)} := u^{(j)} + \delta^{(j)},\ j = 2, 3, \ldots, s+1\ , \tag{3.54}$$

with perturbation $\delta^{(j)} \in L^2(\Omega)$. As shown above, we are able to compute estimates $[\delta^{(j)}]$ of $\|\delta^{(j)}\|_0$. In consequence, we cannot get exact time error estimates, but

$$\hat{\varepsilon}_j := \varepsilon_j + \Theta_j,\ j = 2, 3, \ldots, s+1\ . \tag{3.55}$$

In analogy, we denote by $[\Theta_j]$ estimates of $|\Theta_j|$.

We accept the approximation $\hat{u}^{(s+1)}$ if the inequalities

$$\hat{\varepsilon}_s + [\delta^{(s+1)}] < \mathrm{TOL}\ , \tag{3.56}$$

and

$$[\Theta_j] < \frac{1}{4}\hat{\varepsilon}_j,\ j = 2, 3, \ldots, s \tag{3.57}$$

are fulfilled. From inequality (3.56) follows, that we have to compute time steps with respect to $\rho \cdot \mathrm{TOL}$ instead of TOL, where $0 < \rho < 1$.

Now we want to make a pass through the criterion (3.56) possible, imposing accuracies

$$\mathsf{eps}_j = \kappa(j, s)(1-\rho)\,\mathrm{TOL} \tag{3.58}$$

to the elliptic solver, computing $u^{(j)}$.

For simplicity, let us again consider the 2–stage methods of Chapter 3.1. Given an accuracy eps_2 we first get an approximation $\hat{u}^{(2)}$ of the adaptive Euler scheme

$$\hat{u}^{(2)} = u^{(2)} + \delta^{(2)} \tag{3.59}$$

together with an estimate $[\delta^{(2)}] \le \mathsf{eps}_2$ of $\|\delta^{(2)}\|_0$. Afterwards, we compute the correction $\eta_2^{(1)}$ of the method (3.13). In order to avoid additional perturbations, we use the same grid chosen by the elliptic solver to compute $\hat{u}^2$. As mentioned above, two choices of $Q_0^{(2)}$ are possible. According to

$$Q_L^{(2)}(z) = \frac{1}{1-z} Q_A^{(2)}(z) \tag{3.60}$$

it is useful to derive first a computable expression for

$$\eta_2^{(1)} = Q_A^{(2)}(z)(u_0 + \frac{\tau}{z} g_1)\ . \tag{3.61}$$

Using the definition $z = \tau(A+B) = -\tau H$ and (2.1) simple conversions lead to

$$\begin{aligned} \eta_2^{(1)} &= -\frac{1}{2(1-z)}\left(\frac{z^2}{1-z}u_0 + \frac{z}{1-z}\tau g_1\right) \\ &= -\frac{\tau}{2}(I-\tau H)^{-1}\left(\frac{u^{(2)}-u_0}{\tau} + Au_0 - f(u_0, 0)\right)\ . \end{aligned} \tag{3.62}$$

The corresponding approximation

$$\hat{\eta}_2^{(1)} = -\frac{\tau}{2}(I-\tau H)^{-1}\left(\frac{u^{(2)}+\delta^{(2)}-u_0}{\tau} + Au_0 - f(u_0,0)\right) + \hat{\omega}_0 \tag{3.63}$$

has the representation

$$\hat{\eta}_2^{(1)} = \eta_2^{(1)} + \hat{\omega}_1$$

where

$$\hat{\omega}_1 = -\tfrac{1}{2}(I-\tau H)^{-1}\delta^{(2)} + \hat{\omega}_0\ .$$

In general, $\delta^{(2)}$ will dominate the perturbation $\hat{\omega}_0$, and we get the inequality

$$\|\hat{\omega}_1\|_0 \leq \tfrac{1}{2}\|(I-\tau H)^{-1}\|\|\delta^{(2)}\|_0 \leq \frac{1}{2(1-\mu\tau)}\|\delta^{(2)}\|_0 \ . \tag{3.64}$$

This is in contrast to the linear theory in [6], where the operator norm $(I-\tau H)^{-1}$ is estimated by 1, which leads to much simpler estimates. Hence, we can derive the estimator

$$[\Theta_2] = \frac{1}{2(1-\mu\tau)}[\delta^{(2)}] \ . \tag{3.65}$$

The spatial perturbations of the correction $\eta_2^{(1)}$ yield perturbations of the approximation $u^{(2)}$, namely

$$\hat{u}^{(3)} = \hat{u}^{(2)} + \hat{\eta}_2^{(1)} = u^{(3)} + \delta^{(3)} \ . \tag{3.66}$$

We end up with the estimate

$$[\delta^{(3)}] = [\delta^{(2)}] + [\Theta_2] \ . \tag{3.67}$$

Now, we are able to determine eps_2 by the following guideline:
Set $[\delta^{(2)}] = \mathsf{eps}_2$ in such a way that

$$[\delta^{(3)}] = (1-\rho)\,\mathrm{TOL} \tag{3.68}$$

and

$$\hat{\varepsilon}_2 + [\delta^{(3)}] < \mathrm{TOL} \ . \tag{3.69}$$

By using (3.65) and (3.67) we get

$$[\delta^{(3)}] = \left(1 + \frac{1}{2(1-\mu\tau)}\right)[\delta^{(2)}] = \frac{3-2\mu\tau}{2-2\mu\tau}\,\mathsf{eps}_2 \ . \tag{3.70}$$

This means we have to impose the elliptic accuracy

$$\mathsf{eps}_2 = (1-\rho)\frac{2-2\mu\tau}{3-2\mu\tau}\cdot \mathrm{TOL} \ . \tag{3.71}$$

In order to give a practical meaning to this equation, we estimate the unknown constant μ by its value computed in the previous time step, see Chapter 3.5. For the choice of ρ we look for a value which makes the requirements (3.56) and (3.57) effective for a large class of realistic situations. We try to balance the effects of each dimension, so in the 1–D case

$$\rho = \tfrac{1}{2} \ . \tag{3.72}$$

By analogous arguments we get for $Q_L^{(2)}(z)$ in (3.61) the elliptic accuracy

$$\mathsf{eps}_2 = (1-\rho)\frac{2(1-\mu\tau)^2+1}{2(1-\mu\tau)^2}\,\mathrm{TOL}\ . \tag{3.73}$$

To make implementation easier, we transform the corrections of the second and third method, (3.16) and (3.17) in computable form :

$$\begin{aligned}
\eta_2^{(2)} &= \eta_2^{(1)} + \tau\left(R_2^{(3)}(z) - R_1^{(3)}(z)\right)(g_1-g_2)\\
&= -\frac{\tau}{2(1-z)}\left(\frac{z}{\tau}(u^{(2)}-u_0)+g_1-g_2\right)\ ,\\
\eta_3^{(2)} &= \eta_2^{(1)} - \tau R_2^{(3)}(z)\,(g_1-g_2)\\
&= -\frac{\tau}{2(1-z)}\left(\frac{z}{\tau}(u^{(2)}-u_0\right)+2\,(g_1-g_2)+\\
&\quad +\frac{1}{1-z}\,(g_2-g_1)\ .
\end{aligned} \tag{3.74}$$

Both corrections are special cases of (3.15) with

$$\begin{aligned}
\eta_2 &= \eta_2^{(1)} + \tau(R_1^{(3)}(z)(\kappa-1)+R_2^{(3)}(z))(1-2\kappa)\,(g_1-g_2)\\
&:= \eta_2^{(1)} + \tau G(z;\kappa)\,(g_1-g_2)\ .
\end{aligned} \tag{3.75}$$

For the corresponding approximation of the linear error, an analysis by Taylor expansion leads to the estimate

$$\|\hat\eta_2\|_0 \le \|\eta_2\|_0 + (\tfrac{1}{2}\|(I-\tau H)^{-1}\| + \tau\|G(z;\kappa)\|\|\frac{\partial}{\partial u}g_2\|)\|\delta^{(2)}\|_0 \tag{3.76}$$

where again $Q_A^{(2)}(z)$ is used. Clearly, the evaluation of the second term becomes more complicate. Since in practical computation a good approximation of the Jacobian is necessary such that the update $\eta_2^{(1)}$ realizes a nearly second–order improvement of the approximation, the remainder term in (3.75) yields only a change in the constant of the third order term in Taylor series. In this case, the error of η_2 will be dominated by the error of $\eta_2^{(1)}$. Hence we can again use the elliptic accuracy (3.71) or (3.73).

4 Application of the Schemes

All the schemes developed in the previous sections were applied to a variety of systems of nonlinear parabolic equations. They differ mainly in the nonlinearity, i. e. the right hand side and the boundary conditions. The equations selected here have almost established a standard test set, because they are well known for their difficulties and therefor treated so many times in papers on stiff integrators, moving finite elements and finite differences, [22], [1], [21], [9]. An other aspect is, that for the given examples so many references obtained with different methods are in the literature and we can demonstrate the remarkable behaviour of our schemes for quite different applications.

For clearness of presentation the following abbreviations for the schemes under consideration are used. The adaptive Euler scheme described in (3.11), (3.13) with

$$Q_0^{(2)}(z) = -\frac{1}{2}\frac{z^2}{(1-z)^2}$$

is denoted by m1A (first method, A–stable) and

$$Q_0^{(2)}(z) = -\frac{1}{2}\frac{z^2}{(1-z)^3}$$

is denoted by m1L (first method, L–stable). Analogously we indicate the method (3.8) , compare also (3.11), (3.15) for $\kappa = 0$ with m2A, m2L and for $\kappa = 1$ with m3A, m3L. In Table I we give the amount of numerical work for all methods used. They all have consistency order 2. For problems

	kind of method	# elliptic problems	# g–evaluations
m1A	ROW	2	1
m2A	W	2	2
m3A	W	3	2
m1L	ROW	3	1
m2L	W	3	2
m3L	W	4	2

Table I: Features of the schemes

described below, the Jacobian is given in analytic form and we use numerical approximations of the exact Jacobian in the bilinear forms on each time level. Furthermore, in the case of a system of partial differential equations only one mesh is used for all solution components. In all applications, the initial mesh consisted of 5 equidistributed points.

In view of applications in more space dimensions, we want to get within technical tolerances. That is the reason why we restrict ourselves to second order schemes, because in such cases the higher order codes do not pay off, e.g. [19]. We don't compare CPU–time between our and other algorithms published. In our opinion the run time depends much on the implementation of every component and one has to distinguish between very elaborated codes as LIMEX,[8] and a research code used here. For example, in our code the stiffness matrix is assembled and makes the program remarkably slower than a clever sparse solver. On the other hand, one has to pay for the security of an adaptive algorithm. This means, e.g. during the formation of the grid costly decisions are made and so one has a larger overhead. With respect to this, we think a comparison of CPU–time is not so instructive than the possibility to use our method in more space dimension. In a forthcoming paper we will do this and will give questions of implementation more attention.

4.1 Population Ecology Model

This system of nonlinear elliptic equations was proposed by [18] to model certain planktonit predator–prey situations in which crowding is a factor. With the settings

$v(x,t)$	:=	number of predators, i.e. zooplankton, and
$u(x,t)$	:=	essentially static number of prey, i.e. phytoplankton

the system reads in the domain $\Omega = (0, 2.5)$ for $t > 0$:

$$\begin{aligned} u_t - 0.0125\, u_{xx} &= \left(\frac{35 + 16u - u^2}{9} - v\right) \cdot u \\ v_t - v_{xx} &= \left(u - \frac{5 + 2v}{5}\right) \cdot v \end{aligned} \tag{4.1}$$

with the boundary conditions

$$u_x = v_x = 0 \quad \text{for} \quad t > 0 \ , \ x = 0 \text{ and } x = 2.5$$

and the initial condition

$$u_0 = \begin{cases} 5 & \text{for} \quad 0 \leq x < 1.0 \\ 4x+1 & \text{for} \quad 1.0 \leq x < 1.25 \\ -4x+11 & \text{for} \quad 1.25 \leq x < 1.5 \\ 5 & \text{for} \quad 1.5 \leq x < 2.5 \end{cases}$$

$$v_0 = \begin{cases} 10 & \text{for} \quad 0 \leq x < 1.0 \\ 4x+6 & \text{for} \quad 1.0 \leq x < 1.25 \\ -4x+16 & \text{for} \quad 1.25 \leq x < 1.5 \\ 10 & \text{for} \quad 1.5 \leq x < 2.5 \end{cases}$$

The initial population (u_0, v_0) together with the steady state to which the solution evolves calculated by method m1L is shown in Figure 1 and 2. The tolerance was set to $TOL = 10^{-2}$ and the initial time step was $\tau_0 = 10^{-4}$. The "steady state" was reached after 10 time units. There are different steady state solutions and to which of them the initial one (u_0, v_0) evolves depends on which eigenfunction of the linearized coupled elliptic operator the perturbation of the initial data from $(5, 10)$ most resembles. So, we have to prescribe that the initial condition is represented exactly on the initial grid. This is not a severe restriction here. The problem is relatively easy with respect to the nonlinearity and can be solved with a properly selected number of points on a uniformly spaced grid. We choose this system because the shape of the solution becomes complicate during time. At the steady state, one needs 400 adapted grid points to keep the error below 10^{-2} while 5 points are enough to resolve the initial condition exactly. For the same tolerance, one needs 512 uniform elements to represent the steady state solution. This proofs especially the superiority of our approach to methods with a fixed number of nodes. In Figure 3 the evolution of the number of nodes is depicted, and the efficiency of the algorithms is demonstrated in this case. The adaptive grid is constructed due to the error estimator described in Section 3.4. In Figure 4, we see the length of the time step during the integration of system (4.1). One immediately recognizes, that in the transient phase all algorithms are comparable while in the steady state only the simple adaptive Euler scheme uses greater time steps, the others remaining restricted. Table II contains the statistics of the problem.

4.2 Troesch's Problem

The problem arises in the investigation of the confinement of a plasma column by radiation pressure [27]. It is an inherently unstable two point boundary

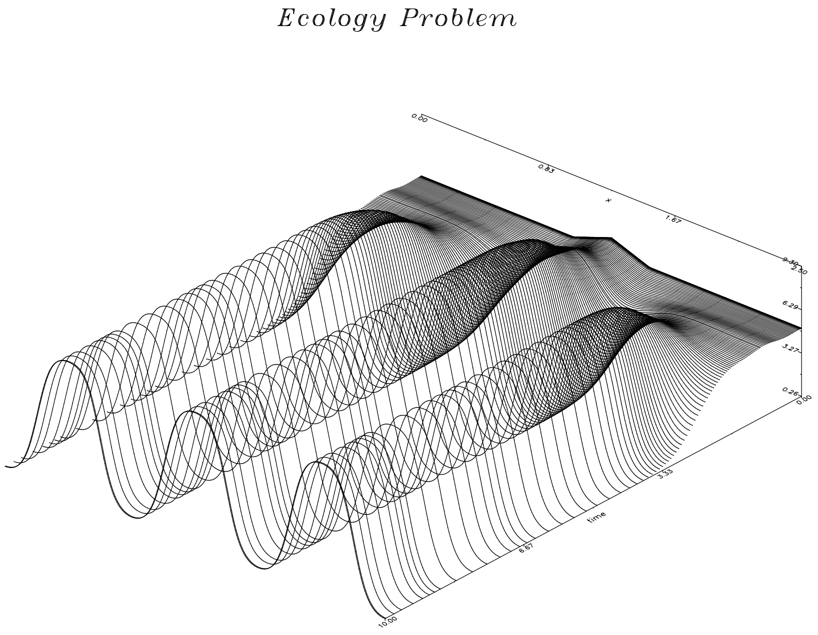

Figure 1: Development of u – component

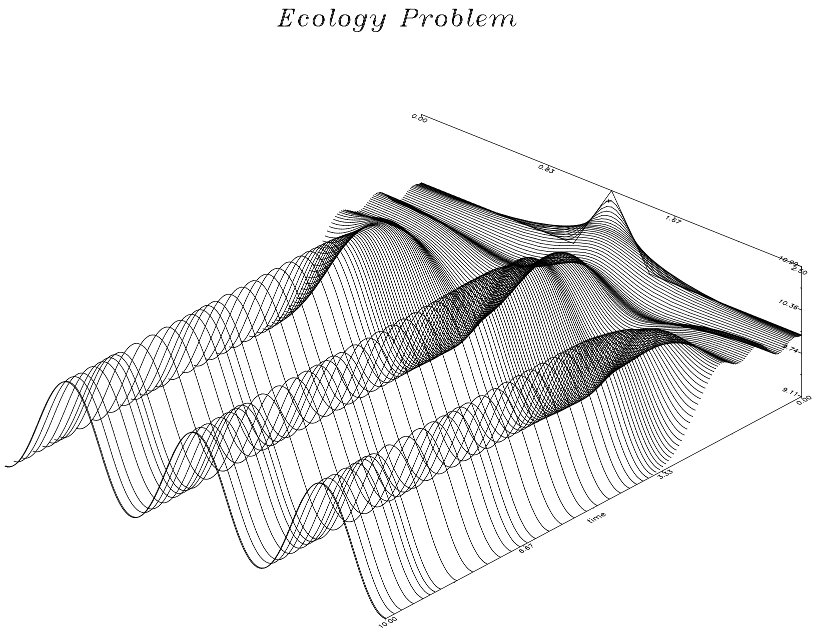

Figure 2: Development of v – component

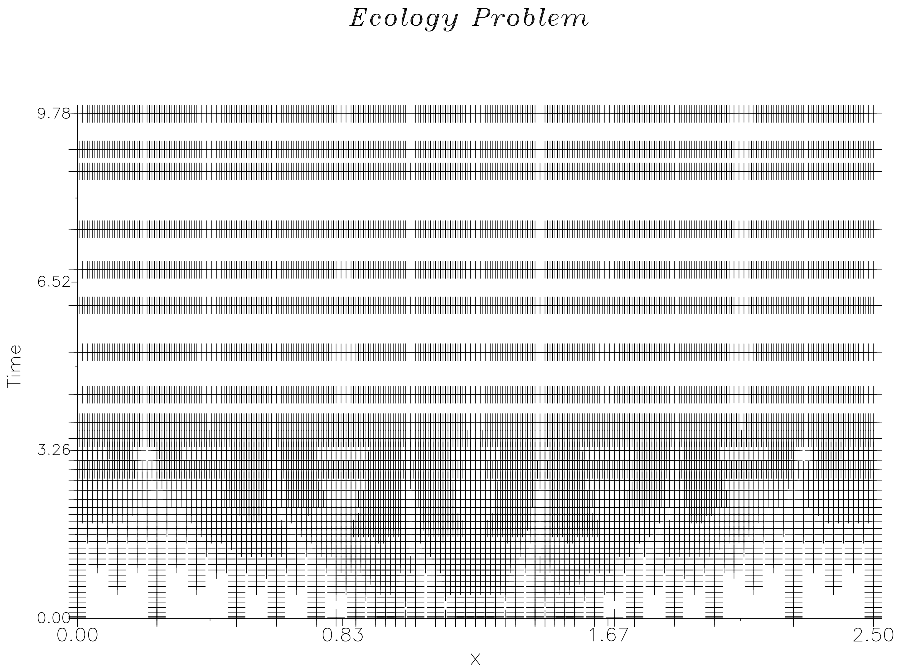

Figure 3: Evolution of grid points

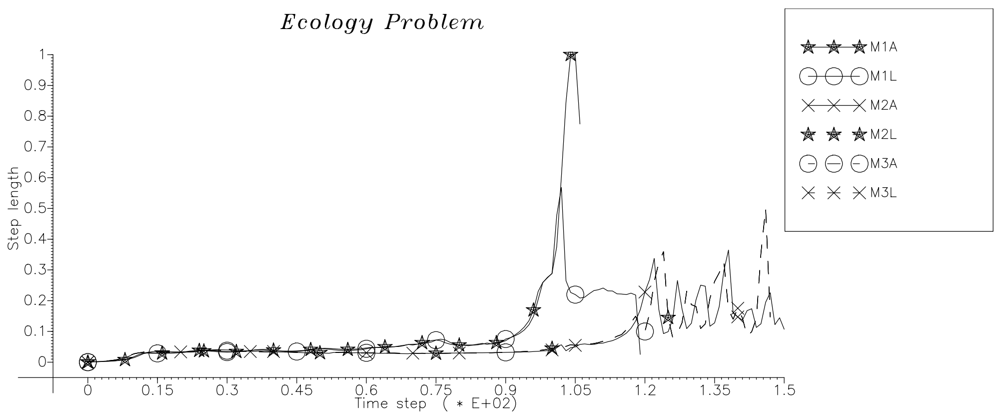

Figure 4: Steplength during integration

	# steps	maximum # nodes	ø # nodes
m1A	107	307	146
m2A	151	363	187
m3A	148	395	181
m1L	120	311	161
m2L	148	395	181
m3L	148	395	187

Table II: Statistics of ecology problem

value problem. For this, it was attacked by many different techniques and serves as a severe test case for any shooting method, [23]. The equation is in $\Omega = (0,1)$ given by

$$u_{xx} - \ 10\sinh(10u) = 0$$

together with it's boundary conditions

$$u(t,0) = 0.0 \quad , u(t,1) = 1.0 \quad .$$

To bring the equation in our context, we define a related parabolic problem

$$u_t - u_{xx} = - \ 10\sinh(10u) \tag{4.2}$$

with the artificial initial condition

$$u(x,0) = 0.0 \quad .$$

A steady state solution is obtained for $t > 0.1$. In Figure 5 the initial condition and the evolution to the steady state are pictured for method m3L. To resolve the inconsistent initial condition, an initial time step $\tau_0 = 10^{-8}$ and a prescribed tolerance of 10^{-3} is needed. After the formation of the boundary layer, the time step can be enlarged significantly. The grid constructed to keep the error below the given tolerance is shown in Figure 6. Nearly all the points are concentrated on the right boundary to model the boundary layer. This strengthens the advantage of a good a priori control of the space discretization. All the schemes are compared in Table III. We mention that the simplest methods m1A, m1L were not able to achieve the prescribed tolerance, necessary to handle the inconsistency and failed. It also demonstrates the improvement through L–stability in this case.

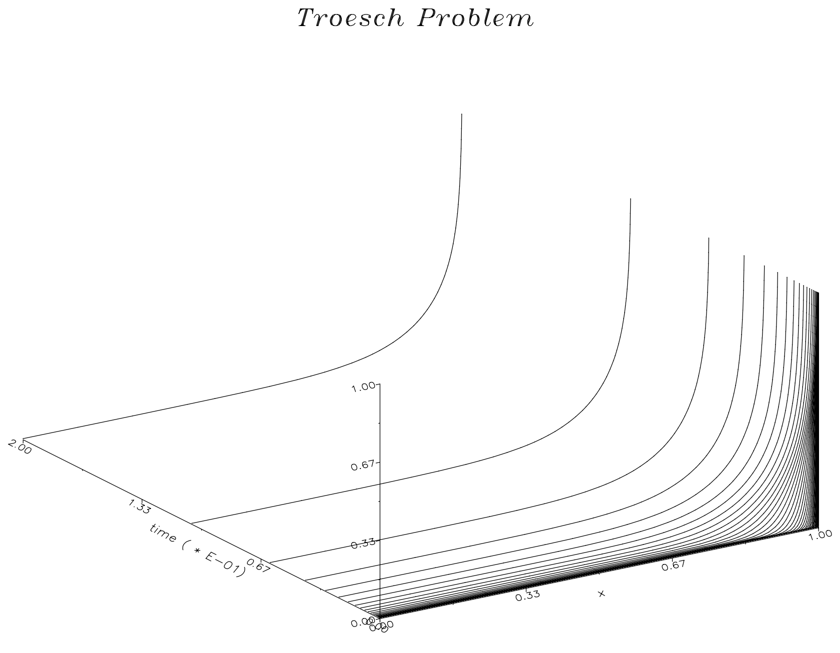

Figure 5: Solution of Troesch's Problem

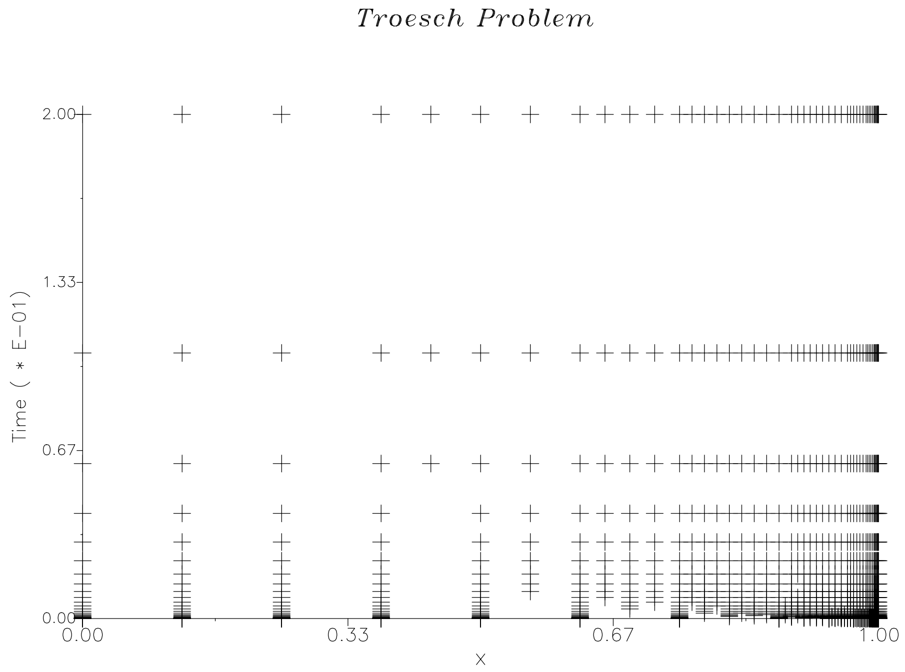

Figure 6: Grids for Troesch's Problem

	# steps	maximum # nodes	ø # nodes
m1a	–	–	–
m2a	69	49	32
m3a	85	49	35
m1l	–	–	–
m2l	55	49	33
m3l	62	49	34

Table III: Statistics of Troesch's problem

4.3 A Problem from Electrodynamics

Both, the problem formulated below and it's stationary version have been treated extensively in [3]. The nonlinear elliptic system defined in $\Omega = (0,1)$ is :

$$\begin{aligned} u_t \; - 0.024 \;\; u_{xx} &= -g(u-v) \\ v_t \; - 0.17 \;\; v_{xx} &= g(u-v) \qquad (4.3) \\ g(\alpha) &:= \exp(5.73\alpha) - \exp(-11.46\alpha) \end{aligned}$$

with the boundary conditions

$$\begin{array}{lclclcl} u_x(0,t) & = & 0.0 & \text{and} & u(1,t) & = & 1.0 \\ v(0,t) & = & 0.0 & \text{and} & v_x(1,t) & = & 0.0 \end{array}$$

and the initial condition

$$u(x,0) = 1.0 \quad \text{and} \quad v(x,0) = 0.0 \ .$$

This nonlinear system is a representative of a singular perturbation problem. The coefficients of u_{xx} and v_{xx} are small and the right hand side $g(\alpha)$ changes rapidly with a small change of α. Consequently, boundary layers are to be expected at $x = 0$ and $x = 1$. From the very beginning, the boundary layer structure of the problem is visible, especially at $x = 1.0$ where u is very steep compared with the rest of the interval. One also sees in Figure 7 and 8 obtained by method m1L with tolerance $TOL = 5 \cdot 10^{-3}$ and the initial step $\tau_0 = 10^{-4}$, that the steady state curves of u and v intermingle between 0.1 and 0.9 and separate at the boundary. They coincide within the straight line $y = 0.8x$ except at the boundary layers. The steplengths during the integration over the time interval $[0.0, 4.5]$ are given in Figure 9. This figure

Electro Problem

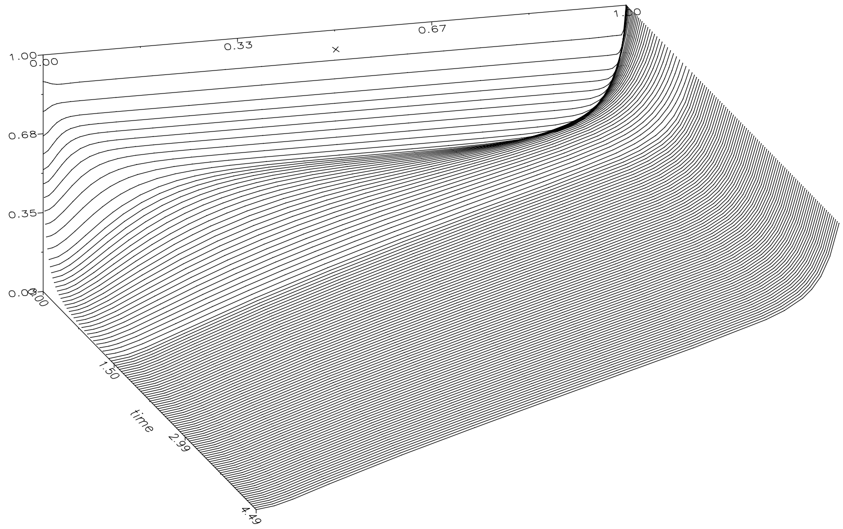

Figure 7: Solution of Electrodynamic Problem, u – component

Electro Problem

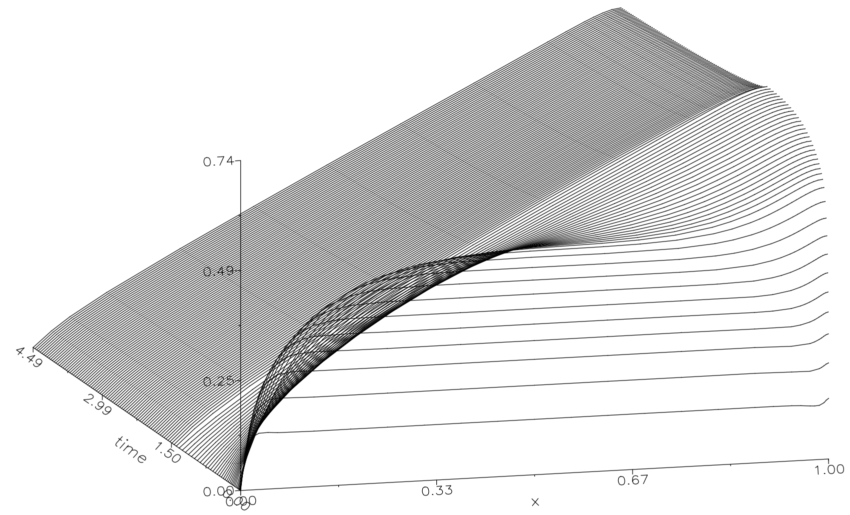

Figure 8: Solution of Electrodynamic Problem, v – component

nicely reflects the hard to resolve initial state, the transient state with rapidly increasing time steps and the dynamic steady state. This can also be seen

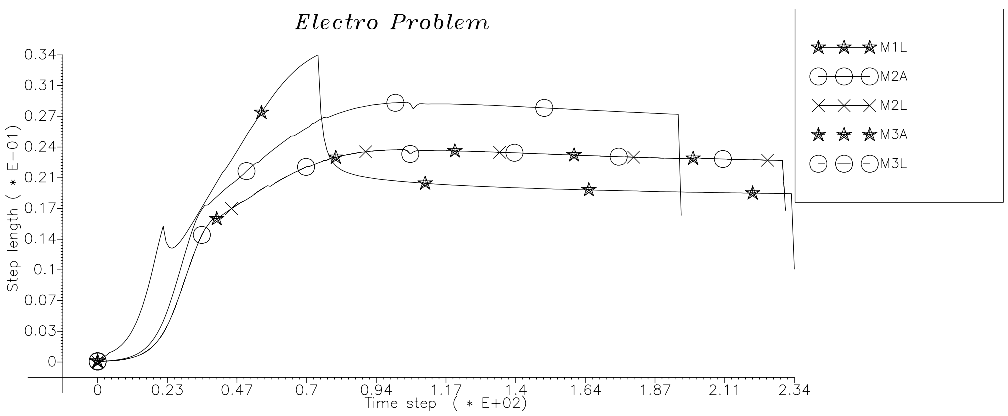

Figure 9: Steplength during integration

in Figure 10, showing the grid at each time level. The main difficulty in this problem is to form the line $y = 0.8x$ from the initial data. The representation of this simple solution shape can be done with much fewer nodes necessary to resolve the initial boundary layers. This again demonstrates the efficiency of our approach. It's remarkable that in this problem method m1A was not able to solve it within the demanded tolerance. The demanded space tolerance leads for all methods to an average node number of 25, shown in Table IV.

4.4 The Dwyer–Sanders Flame Propagation Model

The following equations were first proposed by [9] to simulate several basic features of flame propagation. The system governs the density u of a single

Electro Problem

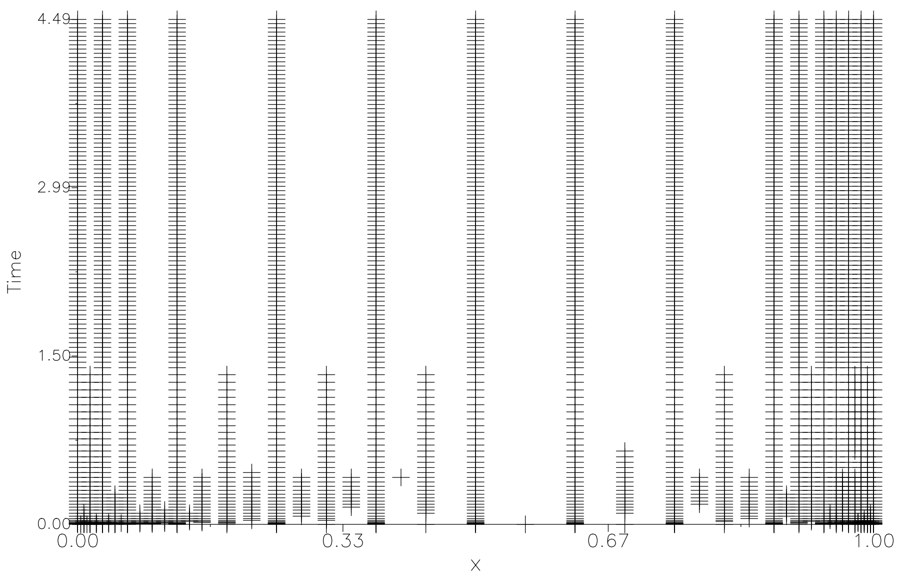

Figure 10: Evolution of the grid

	# steps	maximum # nodes	ø # nodes
m1a	–	–	–
m2a	197	39	23
m3a	232	39	23
m1l	235	39	21
m2l	232	39	23
m3l	232	39	23

Table IV: Statistics of electrodynamic problem species and the temperature v in $\Omega = (0,1)$:

$$
\begin{aligned}
u_t - u_{xx} &= -uf(v) \\
v_t - v_{xx} &= uf(v) \qquad (4.4)\\
f(v) &:= 3.52 \cdot 10^6 \exp(-\frac{4}{v})
\end{aligned}
$$

The boundary and and initial conditions are

$$
\begin{aligned}
u_x(0,t) &= 0.0 \quad , \quad v_x(0,t) = 0.0 \\
u(1,t) &= 0.0
\end{aligned}
$$

$$
v(1,t) = \begin{cases} 0.2 + \dfrac{t}{0.0002} & \text{for} \quad 0 < t \leq 0.0002 \\ 1.2 & \text{for} \quad 0.0002 \leq t \leq 0.006 \end{cases}
$$

$$
u(x,0) = 1.0 \quad , v(x,0) = 0.2 \ .
$$

The time–dependent forcing function for the temperature at the right boundary models a heat source which generates a steep flame front. When v reaches it's maximum, this front starts to propagate from right to left at a relatively high, almost constant velocity ≈ 150. For $t = 0.006$ the front has nearly reached the left boundary. Figures 11 and 12 show the density and temperature computed by m1L from 0.0 to the final time 0.006 with tolerance $TOL = 10^{-2}$ and initial time step $\tau_0 = 10^{-8}$. The small lump for early times is genuine and resolved without any overshoots. This relatively small initial step was necessary to resolve the influence of the heat source. The developing in time of the initial 5–point grid is pictured in Figure 13. A uniform start grid provides a difficult test of the error estimator in space and time, since the method must rapidly refine near $x = 1$ in order to simulate the fast generation of the steep front accurately and to restrict the proposed time step severely. One can see, that the maximum point density exactly follows the flame front, ensuring good resolution and high economy. The statistics of all methods is given in Table V and Figure 14. It's worth mentioning that the simples schemes m1A and m1L have the largest time steps. In contrast they need more nodes to suppress spatial oscillations near the steep flame front. In Figure 15 the quality of the time step control is demonstrated. In Theorem 2 the maximal value of τ was given, still ensuring contractivity of the corresponding Newton method. The lower bound was $\mu * \tau \approx -1$. The figure

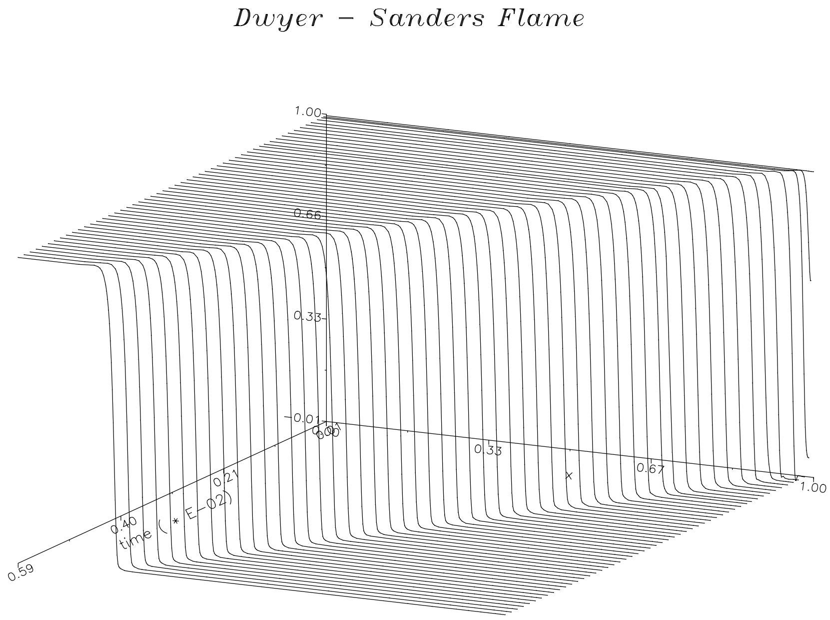

Figure 11: Development of the density

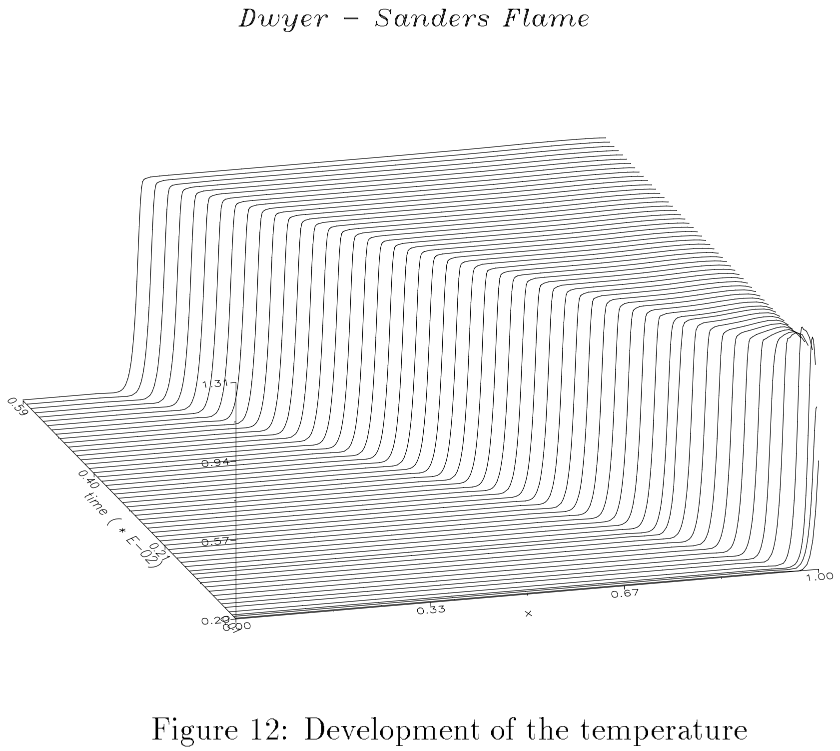

Figure 12: Development of the temperature

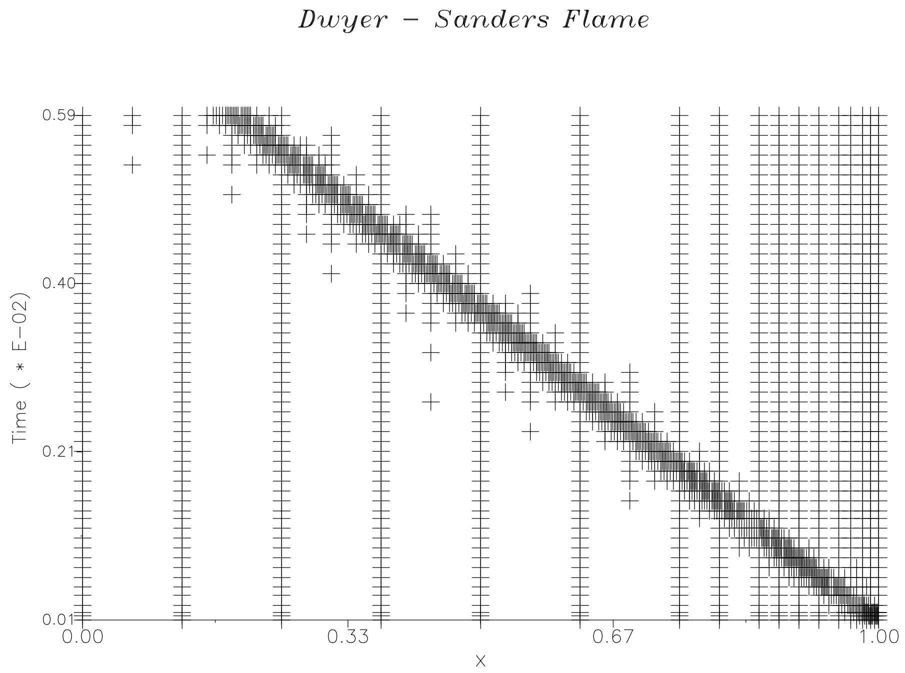

Figure 13: Grid development

	# steps	maximum # nodes	ø # nodes
m1A	156	56	40
m2A	197	33	27
m3A	233	33	27
m1L	164	50	40
m2L	233	33	27
m3L	233	33	27

Table V: Statistics of the Dwyer – Sanders problem

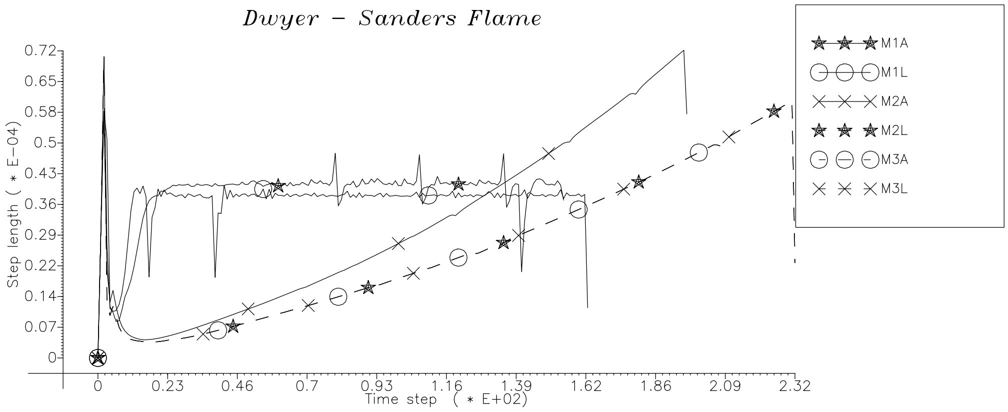

Figure 14: Steplength during integration

nicely reflects the fact, that all algorithms try to use the maximal time step possible. Inspection of the figures justifies the conclusion that our algorithm are able to resolve this difficult problem very satisfactory over the entire space–time domain.

4.5 A Moving Unsteady Flame Front

The equations are a selected case from the bench mark test of numerical methods in flame propagation proposed by [21]. It was the hardest case, for which many participants failed. The flame seems to establish a steep front propagating unchanged in time with a relatively constant speed. However, the moving flame changes it's shape and velocity in the course of integration. Only high skilled algorithms were able to resolve the periodic behaviour of the temperature and the propagation speed of the wave front. With the settings

$u(x,t)$:= temperature,
$v(x,t)$:= chemical species

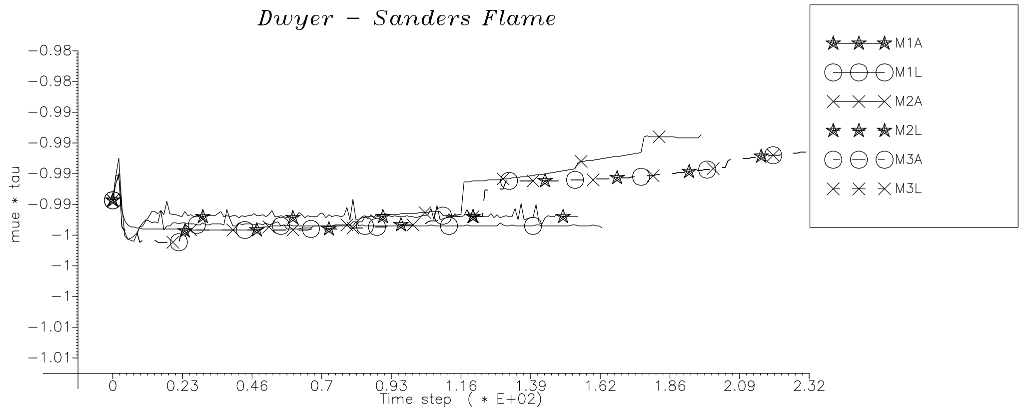

Figure 15: Stepsize control during integration

the reaction–diffusion system is in $\Omega = \mathbb{R}^1$:

$$
\begin{aligned}
u_t \quad - \quad & u_{xx} && = \quad f(u,v) \\
v_t \quad -\frac{1}{Le} \quad & v_{xx} && = \quad -f(u,v) \\
& f(u,v) && := \quad \frac{\beta^2}{2Le} v \exp\left(\frac{-\beta(1-u)}{1-\alpha(1-u)}\right) \quad .
\end{aligned}
\tag{4.5}
$$

The Lewis number is set to $Le = 2$, the nondimensional activation energy is $\beta = 20$ and for the nondimensional heat release we choose $\alpha = 0.8$. The boundary and and the initial conditions are

$$
\begin{aligned}
u(-\infty,t) &= 0.0 \quad , \quad u_x(\infty,t) = 0.0 \\
v(-\infty,t) &= 1.0 \quad , \quad v_x(\infty,t) = 0.0
\end{aligned}
$$

$$u(x,0) = \begin{cases} \exp(x) & \text{for} \quad x \leq 0 \\ 1.0 & \text{for} \quad x > 0 \end{cases}$$

$$v(x,0) = \begin{cases} 1.0 - \exp(Le \cdot x) & \text{for} \quad x \leq 0 \\ 0.0 & \text{for} \quad x > 0 \end{cases} .$$

We solve the system from $t = 0$ to $t = 15.0$ in the restricted domain $\Omega = (-40.0, 20.0)$ with a accuracy of $2 \cdot 10^{-3}$ and an initial time step of 10^{-3}. The solution components computed with m1A are shown in Figure 16 and 17. The participants of the benchmark [21] reported a strong influence of the measure of the restricted domain Ω on their methods. This is a direct consequence of Poincare's Inequality used in Section 2. The peak in the

Moving Flame

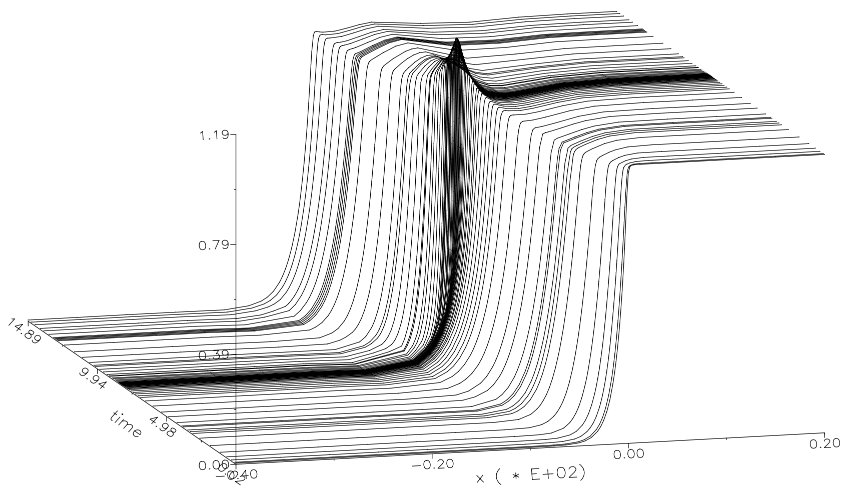

Figure 16: Evolution of the temperature

temperature was only resolved by m1A and m1L. The other schemes, proving to be reliable in the above examples, seem to have some "overstability". This fact allows greater time steps and damps the genuine oscillations in the solution. The effect can also be observed when the tolerance is decreased. The time steps of the schemes are given in Figure 18. In Figure 19 the differences in time step control of the different methods are shown. The lower

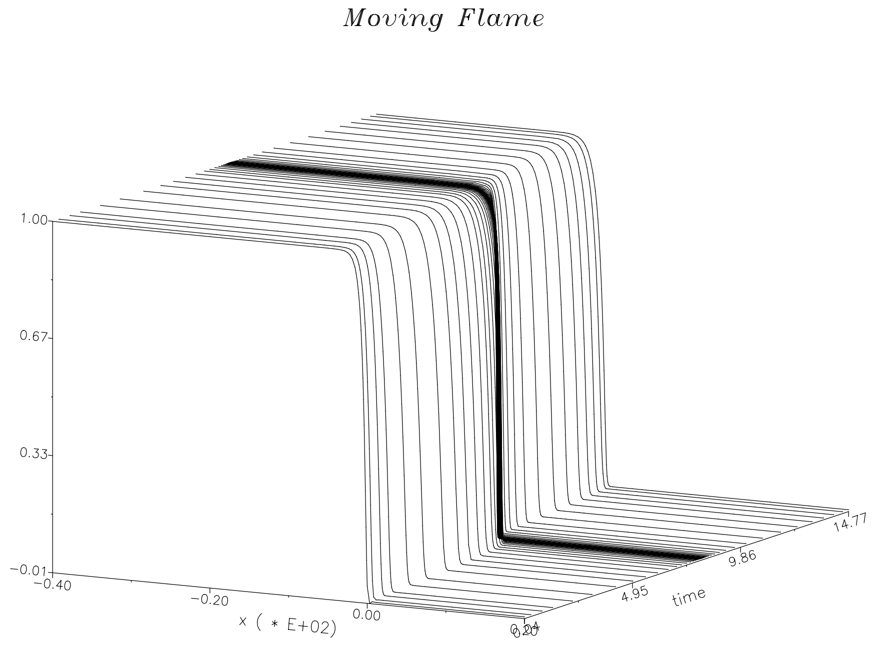

Figure 17: Evolution of the density

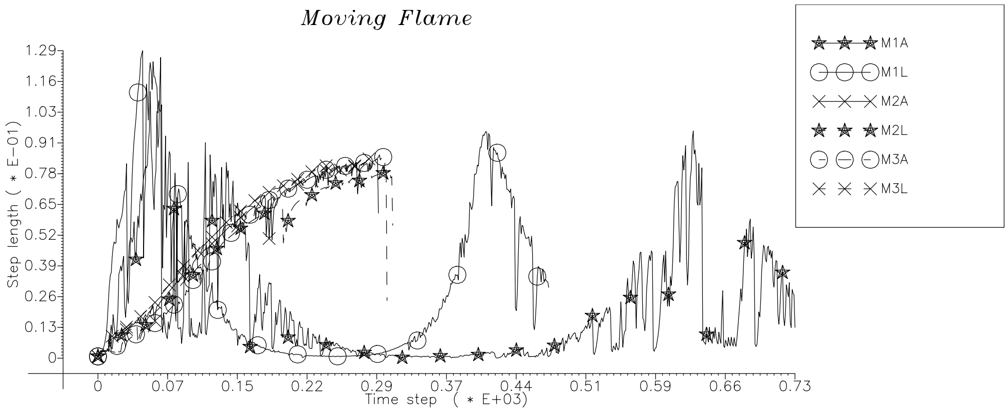

Figure 18: Steplength during integration

bound for the steering quantity was given in Theorem 2 as $\mu * \tau \approx -1$. All methods try to use the largest time step possible, still ensuring contractivity of the underlying Newton method. It can be seen, that some information contained in the smallest eigenvalue μ of the Jacobian is lost for the schemes m2A, m2L, m3A and m3L. These methods "see" a relatively constant μ and therefor tend towards the stability bound -1. One can see from the

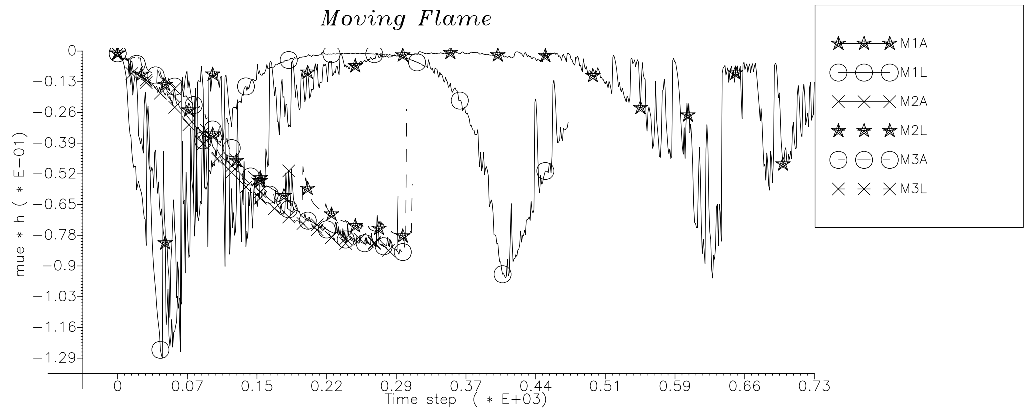

Figure 19: Stepsize control during integration

statistics in Table VI that m1A and m1L need about 25% more nodes for the same tolerance than the other methods. This stems from their better resolution in time. The computation times given, demonstrate the curiosity that the additional effort of the L–stable methods doesn't benefit here. It's interesting, that the formation of the peak is also reflected in the evolution of the grids given in Figure 20. The most remarkable aspect of this example is, that our simplest schemes, due to their step size restriction, are able to resolve the micro–structure of this problem better than more stable methods.

4.6 A Problem from Combustion Theory

The system below was introduced by Kapila [15] to describe a single one step reaction $A \rightarrow B$ of a reacting mixture in a region $0 < x < 1$. The solution component u is the mass fraction of the reacting reactant and v

	# steps	maximum # nodes	ø # nodes	time/sec (SUN–SPARC 1)
m1A	733	47	39	954
m2A	295	34	32	915
m3A	304	34	31	1045
m1L	474	45	38	1040
m2L	310	33	30	951
m3L	286	33	31	969

Table VI: Statistics of the unsteady flame front problem

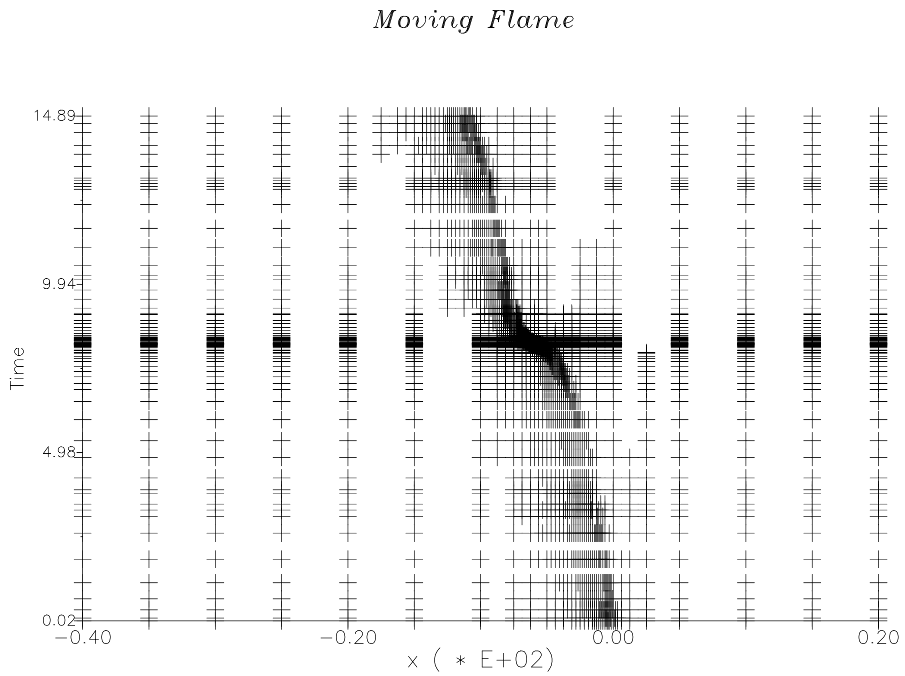

Figure 20: Evolution of the grid

is the reactant temperature. The meaning of the constants are : Le is the Lewis number, α is the heat release, δ is the activation energy, D is the Damköhler number and $R > 0.88$ is the reaction rate. The model is given by the following reaction–diffusion system

$$
\begin{aligned}
u_t \quad &- \quad u_{xx} &= \quad & -Du\exp(-\frac{\delta}{v}) \\
Le \cdot v_t \quad &- \quad v_{xx} &= \quad & \alpha Du\exp(-\frac{\delta}{v}) \\
& \quad D &:= \quad & \frac{R}{\alpha\delta}\exp(\delta)
\end{aligned}
\tag{4.6}
$$

For the different parameters, we choose

$$\alpha = 1.0 \quad , \quad R = 5.0 \quad , \quad \delta = 20.0 \quad \text{and } Le = 0.9 \enspace .$$

The boundary and the initial conditions are

$$
\begin{aligned}
u_x(0,t) &= v_x(0,t) &= 0.0 \\
u(1,t) &= v_x(1,t) &= 1.0 \\
u(x,0) &= v(x,0) &= 1.0
\end{aligned}
$$

At the very beginning, the temperature increases from unity with a "hot spot" at $x = 0$. With the time proceeding, ignition occurs and the temperature at $x = 0$ raises rapidly from about one to near $1 + \alpha$. The formation of a steep flame front begins traveling toward x with speed approximately $\dfrac{e^{\alpha\delta}}{2(1+\alpha)} \approx 10^9$. Thus, the flame front moves very fast after ignition. The problem exhibits a steady state solution once the flame propagates to $x = 1$. The solution in the time interval $0 \leq t \leq 0.25$ obtained from m2L with $TOL = 10^{-4}$ and an initial time step of $\tau_0 = 10^{-8}$ is pictured in Figure 21 and 22. The crucial point here is to get the correct speed of the flame front. It depends on the spatial resolution to simulate the ignition process to a certain degree of accurateness. The coarser the grid the faster is a relatively too flat flame front and many different flame speeds are published in the literature. A very low tolerance of 10^{-4} for the discretization is therefor necessary. In Figure 23 we see that the grid is able to follow the dynamics of the problem, and the constant speed of the front is shown. A comparison of the flame speed with other, very accurate computations, c.f. [22], gives no significant difference. For this examples, the methods m1A and m1L were not able to resolve the starting process and returned a much too slow and too

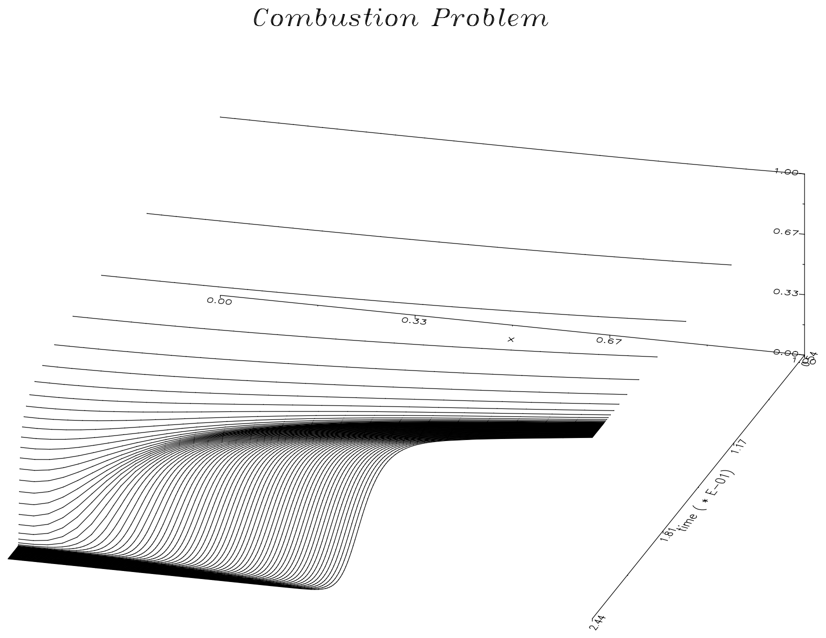

Figure 21: Solution of the combustion problem, u–component

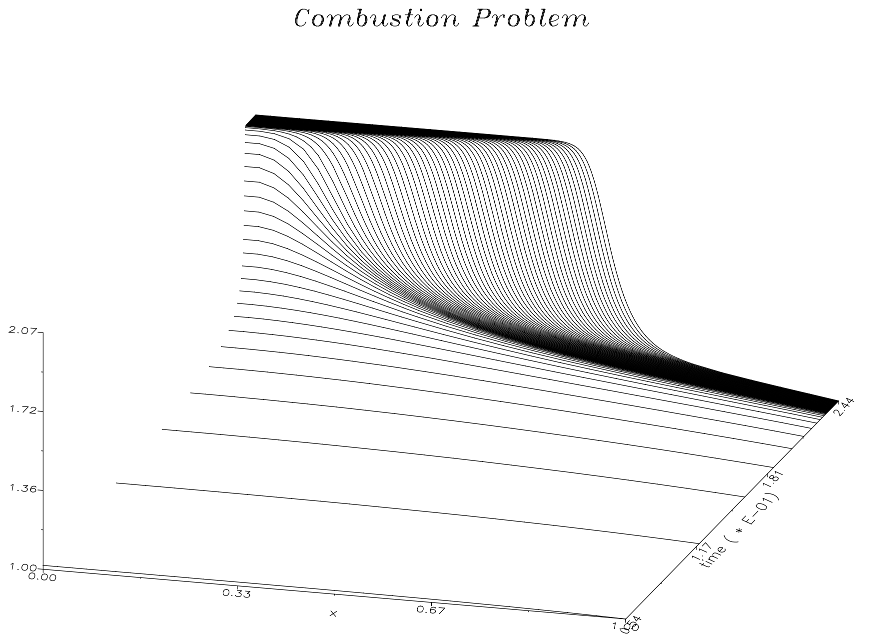

Figure 22: Solution of the combustion problem, v–component

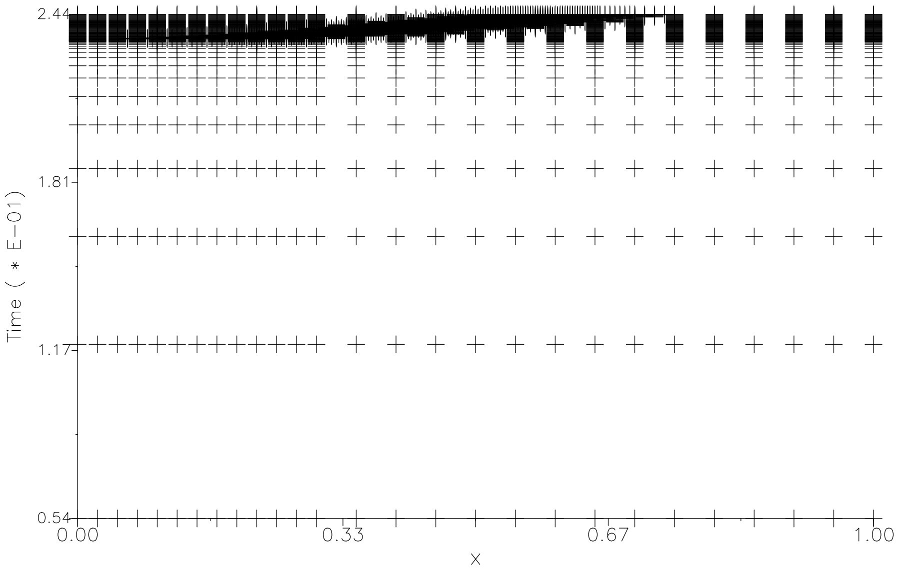

Figure 23: Development of the grid

	# steps	maximum # nodes	ø # nodes
m1A	–	–	–
m2A	776	91	56
m3A	762	90	56
m1L	–	–	–
m2L	762	90	56
m3L	762	90	56

Table VII: Statistics of the combustion problem

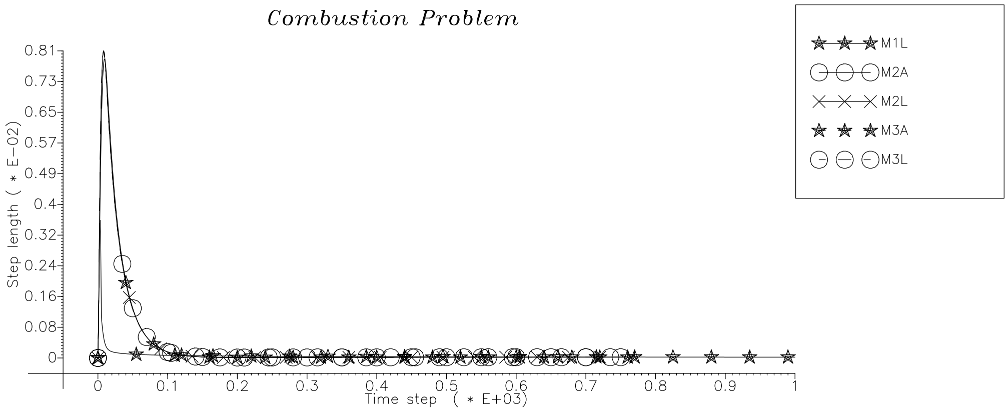

Figure 24: Steplength during integration

flat solution curve. The steplength of the other methods are given in Figure 24 and the statistics is contained in Table VII. This is a very difficult problem, but our methods are capable of finding a solution with relative ease. In this problem the benefits of L-stability and more elaborated schemes are evident.

5 Conclusion

The aim of our project was to develop a reliable and efficient method, which can be used to resolve e.g. unsteady flame propagations in a full adaptive way. Furthermore, the user should be relieved from fine tuning of several parameters, steering the adaptation process, as far as possible. The techniques elaborated in our paper go towards fulfilling this requirements.
We have applied an adaptive finite element method to a class of nonlinear initial boundary value problems of the reaction–diffusion type with possible rapid spatial and temporal transitions. The spatial adaptation is connected with an equidistribution of the meaningful measured spatial error necessary to follow steep travelling fronts and to resolve boundary and interior layers. The second adaptation is the use of variable time stepsize in the numerical

integration. The steplength is determined in an efficient way during the solution process.
The numerical results obtained by the ROW–schemes m1A and m1L (adaptive Euler) show an unbalanced behaviour: They work either very well or fail. However, L–stability of the method m1L leads to an efficient solution of all discussed equations excepting Troesch's problem, characterized by an inconsistency of initial data. The W–methods m2A, m3A and m2L, m3L do not differ considerably. Here, the addition of L–stability yields also an improvement of the efficiency due to larger time steps. The unsteady flame problem indicates, that their good linear stability properties are not the full truth, as is shown in the resolution of the peak in the temperature.
We consider the presented nonlinear test problems to be an interesting set for the design of adaptiv methods since they contain all typical difficulties of reaction–diffusion type systems. Since our approach is able to handle systems in two space dimensions also, this will be investigated in a next paper.

Acknowledgements : The authors wish to acknowledge P. Deuflhard for inspiring discussions about his work on stiff ODE integrators. He and F. A. Bornemann did also a first critical reading of our manuscript and gave many constructive remarks. In addition, we wish to thank F. A. Bornemann for placing his program KASTIO1 at our disposal. We also thank U. Nowak for supplying us with reference solutions obtained by his moving finite difference method on some numerical examples presented here.

References

[1] S. Adjerid, J. E. Flaherty : *A Moving Finite Element Method with Error Estimation and Refinement for One–Dimensional Time Dependent Partial Differential Equations* SIAM J. Numer. Anal. **23**, pp. 778–796 (1986)

[2] I. Babuska, W. C. Rheinboldt : *Error Estimates for Adaptive Finite Element Computations* SIAM J. Num. Anal. **15**, pp. 736–754 (1978)

[3] M. Bakker : *Software for Semi–Discretization of Time–Dependent Partial Differential Equations in One Space Variable* Preprint NW 52/77, Mathematisch Centrum Amsterdam (1977)

[4] M. Bieterman, I. Babuska : *An Adaptive Method of Lines with Error Control for Parabolic Equations of the Reaction-Diffusion Type* J. Comp. Phys. **63**, pp. 33–66 (1986)

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

[6] 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**, pp. 93-122 (1991)

[7] P. Deuflhard : *Uniqueness Theorems for Stiff ODE Initial Value Problems* D. F. Griffiths, G. A. Watson (eds.) : Numerical Analysis 1989, Proceedings of the 13th Dundee Conference, June 1989 John Wiley & Sons, New York (1989)

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

[9] H. A. Dwyer, R. J. Kee, B. R. Sanders : *Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer* AIAA J. **10**, pp. 1205–1212 (1980)

[10] R. M. Furzeland, J. G. Verwer, P. A. Zegeling : *A Numerical Study of Three Moving Grid Methods for One-Dimensional Partial Differential Equations which are based on the Method of Lines* Report NM–R8806, Centre for Mathematics and Computer Science (CWI), Amsterdam (1988)

[11] A. Friedli : *Verallgemeinerte Runge–Kutta Verfahren zur Lösung steifer Differentialgleichungssysteme* Lecture Notes in Mathematics, Vol. 631, pp. 35–50, Berlin–Heidelberg–New York (1970)

[12] G. H. Golub, C. F. Van Loan : *Matrix Computations* The John Hopkins University Press, Baltimore and London (1989)

[13] E. Hairer, G. Wanner : *Solving Ordinary Differential Equations II, Stiff and Differential–Algebraic Problems* Springer Series in Computational Mathematics 14. Springer, Berlin–Heidelberg–New York (1991)

[14] N. J. Higham : *Bounding the error in Gaussian elimination for tridiagonal systems* Technical Report 88–953, Department of Computer Science, Cornell University (1988)

[15] A. K. Kapila : *Asymptotic Treatment of Chemically Reacting Systems* Pitman Applicable Mathematics Series, Pitman, New York (1983)

[16] R. Dautray, J. Lions : *Mathematical Analysis and Numerical Methods for Science and Technology, Volume 2* Springer, Berlin–Heidelberg–New York (1988)

[17] K. Miller, R. N. Miller : *Moving finite elements I.* SIAM J. Numer. Anal. **18**, pp. 1019–1032 (1981)

[18] J. D. Murray : *Lecture on Nonlinear–Differential–Equation Models in Biology* Oxford University Press, Oxford (1977)

[19] A. Ostermann, P. Kaps, T. D. Bui : *The Solution of a Combustion Problem with Rosenbrock Methods* ACM Trans. Math. Software **12**, pp. 354–361 (1986)

[20] A. Pazy : *Semigroups of Linear Operators and Applications to Partial Differential Equations.* Springer, Berlin–Heidelberg–New York (1983)

[21] N. Peters, J. Warnatz (eds.) : *Numerical Methods in Laminar Flame Propagation.* Notes in Numerical Fluid Dynamics 6, pp. 232–260, Vieweg (1982)

[22] L. R. Petzold : *Observations on an Adaptive Moving Grid Method for One–Dimensional Systems of Partial Differential Equations* Applied Numerical Mathematics **3**, pp. 347–360, North–Holland (1987)

[23] S. M. Roberts, J. S. Shipman : *Solution of Troesch's Two–Point Boundary Value Problem by a Combination of Techniques* Journal of Computational Physics **10**, pp. 232–241, (1972)

[24] E. Rothe : *Zweidimensionale parabolische Randwertaufgaben als Grenzfall eindimensionaler Randwertaufgaben* Math. Ann. **102**, 650–670 (1930)

[25] J. L. Siemieniuch : *Properties of certain rational approximations to e^{-t}* BIT **16**, 172–191 (1976)

[26] K. Strehmel, R. Weiner : *Linear–implizite Runge–Kutta–Methoden und ihre Anwendungen (German)* B. G. Teubner Verlagsgesellschaft Stuttgart–Leipzig (1992)

[27] B. A. Troesch : *Intrinsic Difficulties in the Numerical Solution of a Boundary Value Problem* Internal Report NN–142, TRW Inc. Redondo Beach Cal. (1960)