Diploma Thesis in mathematics:

Comparison of Rosenbrock methods with modified Patankar schemes used in biogeochemical modelling

prepared by:	Bianca Schippmann
field of study:	Diplom Mathematics
institute:	of natural science
department:	of Mathematics
matriculation number:	3200641
first supervisor:	Prof. Dr. Hans Burchard
second supervisor:	Prof. Dr. Klaus Neymeyr
process time :	02.06.2008 - 02.12.2008

Für Oma Elli und Günter.
Ich wünschte, ihr könntet das noch erleben.

Danksagung

Ich danke Hans Burchard dafür, dass er mir die Möglichkeiten gab diese Arbeit zu schreiben, sowie für die gute Betreuung und die anregenden Diskussionen. Weiterhin danke ich ihm dafür, dass er meine Teilnahme an der BIOCAT Summer School in Kiel und auch an dem FORTRAN Programmierlehrgang in Stuttgart ermöglicht hat. Es waren tolle Erfahrungen, die ich nicht mehr missen möchte.

Ich danke außerdem auch Herrn Neymeyr für die hilfreichen Anmerkungen und Ratschläge.

Mein besonderer Dank gilt Inga Hense. Sie hat mich während der ganzen Zeit mit hilfreichen Korrekturen, Gesprächen und Diskussionen unterstützt. Inga stand mir immer, auch an ihren freien Tage, tatkräftig zur Seite und hat mir dadurch sehr bei der Erstellung dieser Arbeit geholfen.

Ich danke auch Elisabeth Fischer für Ihre Hilfe und Tipps bei all meinen LaTeX Problemen und auch allen anderen Doktoranden- und Diplomandenkollegen für die stetig gute Stimmung bei unseren gemeinsamen Pausen.

Weiterhin danke ich meinen Eltern dafür, dass sie mich in meiner ganzen Studienzeit und besonders in diesem letzten, von so schweren Verlusten gekennzeichneten Jahr, unterstützt haben wo sie nur konnten. Allein durch sie hatte ich die Kraft diese Arbeit zu schreiben. Ihr seid mein Fels in der Brandung!

Zum Schluß bedanke ich mich bei André Holtz für die unendliche Kraft und Motivation, die er mir gab, wann immer sie mich verlassen hatte, in dieser besonders für ihn so schweren Zeit.

Thesen

- Im Rahmen des *BMBF* "Verbundprojektes" *SOPRAN* (***S****urface* ***O****cean* ***Pr****ocesses in the* ***An****thropocene*), in das diese Arbeit eingegliedert ist, werden neben Feld-Studien in den Kap-Verden und der Ostsee auch biogeochemische Modelle (Anfangswertprobleme) benutzt, um biologische und chemische Schlüsselprozesse zu untersuchen. Diese Modelle müssen numerisch gelöst werden.

- Die numerischen Schemata müssen die Eigenschaften der Modelle - Positivität und Konservativität - berücksichtigen und widerspiegeln.

- Bisher nutzte man dafür unter anderem die expliziten Runge-Kutta und die quasiimpliziten modifizierten Patankar-Verfahren, die aber nur teilweise den gestellten Anforderungen genügen. Die Runge-Kutta-Methoden sind nicht positiv und eignen sich nicht zum Lösen von biogeochemischen Prozessen, die auf unterschiedlichen Zeitskalen ablaufen (steife Differentialgleichungen). Die quasi-impliziten Verfahren sind für einfache Probleme ungenau und grundsätzlich rechenaufwendig.

- Besonders für die Aufgabenstellung des *SOPRAN*, die in der Sektion *Physikalischen Ozeanographie und Messtechnik* des *Leibniz Instituts für Ostseeforschung Warnemünde (IOW)* bearbeitet wird, ist es wichtig genauere und schnelle Verfahren zu finden, die zudem auch noch steife Systeme lösen können. Gegenstand dieser Arbeit ist zu testen, ob die semi-impliziten Rosenbrock-Methoden eine geeignete Alternative darstellen.

- Die genannten Verfahren wurden auf drei unterschiedlich komplexe Modellprobleme angewendet und ihr Verhalten hinsichtlich Genauigkeit und Rechenaufwand miteinander verglichen.

- Die Rosenbrock-Methoden können alle Modellprobleme lösen und sind positiv, wenn die entsprechenden Parameter hinreichend klein gewählt werden.

- Zum Lösen von einfachen biogeochemischen Problemen eigenen sich die Runge-Kutta Verfahren auf Grund ihres geringeren Rechenaufwands besser, als die Rosenbrock-Methoden.

- Im Vergleich zu den Patankar-Verfahren stellen die Rosenbrock-Methoden eine gute Alternative zum Lösen biogeochemischer Probleme dar; besonders für solche, die auf unterschiedlichen Zeitskalen ablaufen.

- Um zu testen, wie gut sich die Rosenbrock-Methoden zum Lösen realistischer Ökosystemmodellprobleme eignen, die im Ozean auftreten, werden sie als nächstes in das *General Ocean Turbulence Modell (GOTM)* (www.gotm.net) eingebaut und ihr Verhalten untersucht.

Contents

List of Tables

List of Figures

1 Introduction

Marine biogeochemical modelling is a branch of earth system science, in which processes that govern the fluxes or cycling of energy or matter in the ocean are realised with computer models. Often, biogeochemical models are used as predictive instruments to simulate the behaviour of complex (three-dimensional and time-varying) systems and to compute scenarios of the system's behaviour under varying environmental conditions. Examples are studies addressing the changes in the discharge of river nutrients or the effect of climate change on the ocean carbon cycle.
Modelling related to climate change is conducted within a number of different national, European and international projects. One of them is the German *BMBF* "Verbundprojekt" *SOPRAN* (***S**urface* ***O**cean* ***Pr**ocesses in the* ***An**thropocene*), a part of the international *SOLAS* (***S**urface* ***O**cean* - ***L**ower* ***A**tmosphere* ***S**tudy*) programme. *SOPRAN* addresses the interactions between atmosphere, climate and ecosystems focusing on processes within and close to the surface ocean and their potential changes over the next years. Besides field studies at the Cape Verde site and the Baltic Sea, a number of biogeochemical models are developed and used to study biological and chemical key processes in the surface layer. This study has been carried out within the framework of *SOPRAN*.

In general biogeochemical models describe the growth and decay processes by mathematical systems, so-called ordinary differential equations. For most of them there are no analytical solutions and hence numerical integration schemes are applied to approximate the solution. Current schemes applied to biogeochemical models are for example the following numerical methods, which are implemented in the ***G**eneral* ***O**cean* ***T**urbulence* ***M**odel (GOTM)*, developed by *Burchard et al.* [1999], in order to simulate ecosystem processes: the Euler forward method, developed by Euler in 1768, the Runge Kutta schemes, first constructed by Runge 1895 and Heun 1900 and finally formulated by Kutta 1901, the Patankar methods of first and second order, developed by *Patankar* [1980] and *Burchard et al.* [2003], respectively, as well as the modified and extended modified Patankar schemes of first and second order, proposed by *Burchard et al.* [2003] and *Bruggeman et al.* [2007], respectively.
Due to their low computational effort the Runge Kutta methods are still used for modelling the so-called ***N**utrient* - ***P**hytoplankton* - ***Z**ooplankton* - ***D**etritus* - type models, see e.g. *Fasham et al.* [1990]. The NPZD-models describe biological processes on daily scales. Recent biogeochemical models consider also processes, which run on significantly shorter time scales, e.g. the iron speciation, which are more in detail explained by *Weber et al.* [2007]. In general, biogeochemical models consider processes running on different time scales. For solving these kind of models advanced numerical schemes (modified Patankar schemes) are applied.
From a numerical point of view, two characteristics are crucial for biogeochemical models.

First, the state variables represent non-negative quantities, such as concentrations of chemical compounds and second, they conserve mass as well as energy. Both features have to be respected by the numerical methods. The Euler forward as well as the Runge Kutta methods fulfil both characteristics, if the step size is chosen sufficiently small. Unfortunately, that leads to higher computational effort and hence they loose their advantage of computational efficiency. If larger time steps are used, they might compute negative concentrations also for positive initial values. The problem of non-negativity first was addressed by *Patankar* [1980] for numerical turbulence models. Based on the forward Euler method he proposed the *Patankar Euler method* - a numerical scheme of first order, which was extended to second order by *Burchard et al.* [2003]. Both schemes are not conservative, due to the fact that they have been developed for turbulence modelling, where conservativity is not essential. The further development of this so-called Patankar schemes resulted in the *modified Patankar schemes* of first and second order, proposed by *Burchard et al.* [2003]. These new schemes satisfy the underlying demands of positivity and conservativity, whereas the second order version is additionally suitable for solving stiff ordinary differential equations. Such equations are characterised e.g. by the involvement of significantly different time scales. However, the modified Patankar schemes have two important disadvantages: high computational effort, due to the necessity of solving a linear equation system on the one hand and on the other hand, they imply conservation on state variable level only. That means the conservativity does not hold in biochemical sense, where it refers to the conservation of atoms as well as of energy, as shown by *Bruggeman et al.* [2007]. The authors addressed this problem and developed the *extended modified Patankar schemes* of first and second order, inspired by the work of *Patankar* [1980] and *Burchard et al.* [2003]. Both methods satisfy the underlying main characteristics too, but also have two important disadvantages: first, they have high computational effort, due to solving a root problem, and second, they are not suitable for solving stiff ordinary differential equations.

While for biogeochemical modelling numerical methods, which are designed for solving stiff problems are seldom in use, in atmospheric chemistry such schemes are often applied. *Sandu et al.* [1997a] and *Sandu et al.* [1997b] searched for appropriate numerical schemes to solve chemical transport reaction equations, described by stiff ordinary differential equations. The authors benchmarked the Rosenbrock solvers - ROS3 and ROS4 - as the most accurate and most cost-effective numerical methods of all tested schemes for all tested models. The systems, describing these chemical transport reactions, are similar to those describing the processes in biogeochemistry and hence the Rosenbrock methods are also applicable to biogeochemical modelling.

The goal of my thesis is to compare currently used numerical schemes with the Rosenbrock methods and to investigate, whether the Rosenbrock solvers are suitable for application to biogeochemical models. Therefore, the two Rosenbrock solvers ROS3 and ROS4, the Euler forward, the Runge Kutta method of second and fourth order as well as the modified and extended modified Patankar schemes of second order are applied to three test cases and their performance in terms of accuracy and computational efficiency is compared.

The thesis is structured in the following way: In the second chapter the mathematical foundation and the most important mathematical definitions are given. The third chapter introduces the Rosenbrock methods. After a short introduction in biogeochemical modelling, the test cases are presented in chapter 4 and subsequently, the seven numerical methods used for comparison are specified in detail. In chapter 5 all tested results are compared with each other and the dependence on the two tolerance parameters of the Rosenbrock solvers is explained. The thesis ends with discussion and conclusions and with an outlook of the future work.

2 Mathematical foundations

To provide a consistent basis for the following study some essential information about *ordinary differential equations (ODEs)*, their solution and solution properties is given in this chapter as well as an overview about numerical methods for solving them.

Definition 2.0.1.
Let G be a subset of $\mathbb{R} \times \mathbb{R}^N$ and let $f\colon G \to \mathbb{R}^N$ be a continuous function such that $(t, c(t)) \mapsto f(t, c(t))$. Then

$$c'(t) = \frac{dc}{dt}(t) = f(t, c(t)) \tag{2.1}$$

is called a *system of N explicit ordinary differential equations of first order.*
It is called *implicit*, if the right hand side additionally depends on $c'(t)$ and furthermore the ODE (2.1) is called *autonomous*, if the right hand side of equation (2.1) does not explicitly depend on t, otherwise it is called *non-autonomous.*

Remark 2.0.1.
There is only a closed solution theory for (systems of) explicit ODEs.

The solution of equation (2.1) is a function ϕ: I $\mapsto$ $\mathbb{R}^N$ fulfilling the following characteristics:

1. $I \subset \mathbb{R}$ is an interval and G contains the graph of ϕ, i.e.

$$\Gamma_\phi := \{(t, c(t)) \in I \times \mathbb{R} : c(t) = \phi(t)\} \subset G \tag{2.2}$$

2.

$$\phi'(t) = \frac{d\phi}{dt}(t) = f(t, \phi(t)) \quad \forall t \in I. \tag{2.3}$$

In general the solution set of an ODE is not uniquely constrained by the equation but needs further initial or boundary values. The latter are not further discussed here for the present study.

Definition 2.0.2.
Let G be a subset of $I \subset \mathbb{R} \times \mathbb{R}^N$ and $f\colon G \to \mathbb{R}^N$ a continuous function such that
$(t, c(t)) \mapsto f(t, c(t))$. An *initial value problem* (also called *Cauchy Problem*) is given in the following form:

$$\begin{cases} \frac{dc}{dt}(t) & = f(t, c(t)), \quad t \in \ I \\ c(t_0) & = c_0, \quad t_0 \in I \text{ fixed.} \end{cases} \tag{2.4}$$

In practise initial values mostly are measured and not given, which leads to inexact initial values and hence to not accurate solutions. Thus the notation of *stability* is important because it determines to what extent the solution is sensitive towards a perturbation of the initial values. An assumption for stability is the *Lipschitz condition* of the right hand side of equation (2.1), which is defined in the following way:

Definition 2.0.3.
Let G be a subset of $\mathbb{R} \times \mathbb{R}^N$ and let $f\colon G \to \mathbb{R}^N$ be a function. While f fulfils a *Lipschitz condition* in G, in case all $(t, c(t)), (t, \widetilde{c}(t)) \in G$ apply

$$\|f(t, c(t)) - f(t, \widetilde{c}(t))\| \leq L \cdot \|c(t) - \widetilde{c}(t)\| \tag{2.5}$$

where $L \in \mathbb{R}_{\geq 0}$ is the *Lipschitz constant* and $\|\cdot\|$ is an arbitrary vector norm.

Theorem 2.0.1.
Let $c, y : [t_0, t_e] \to \mathbb{R}^N$ be the solutions of the following two initial value problems

$$\begin{cases} \frac{dc}{dt}(t) & = f(t, c(t)) \\ c(t_0) & = c_0 \end{cases} \qquad \begin{cases} \frac{dy}{dt}(t) & = f(t, y(t)) \\ y(t_0) & = y_0, \end{cases} \tag{2.6}$$

and $L \in \mathbb{R}_{\geq 0}$. If f applies the Lipschitz condition for all $t \in [t_0, t_e]$ and all $c(t), y(t) \in G$, then:

$$\|c(t) - y(t)\| \leq e^{L \cdot (t - t_0)} \|c_0 - y_0\| \tag{2.7}$$

follows. That means, the solutions depend Lipschitz continuously on the initial values.

More details and the proof of this theorem can be found e.g. in *Heuser* [1989].

2.1 Existence and uniqueness of solutions

In general, the solution of the initial value problem (2.4) is not unique, as can be seen in the next example.

Example 2.1.1.
The initial value problem

$$\begin{cases} \frac{dc}{dt}(t) & = c^{2/3}(t) \\ c(t_o) & = 0 \end{cases} \tag{2.8}$$

has two solutions

$$\begin{aligned} c_1(t) &= \frac{1}{27}(t - t_0)^3 \quad \text{and} \\ c_2(t) &= 0 \end{aligned} \tag{2.9}$$

and thus cannot uniquely be solved.

Addressing the initial value problem the question raises, whether there is a local solution of equation (2.1) by given initial values. Before giving sufficient criteria, which respond to that, the notation of the *local* Lipschitz condition is formulated.

Definition 2.1.1.
Again let G be a subset of $\mathbb{R} \times \mathbb{R}^N$ and let $f\colon G \rightarrow \mathbb{R}^N$ be a function. f applies a *local Lipschitz condition*, if every point $(t, c(t)) \in G$ has a neighbourhood U, such that f fulfils a Lipschitz condition in $G \cap U$ with a U depending constant $L \in \mathbb{R}_{>0}$, i.e.

$$\|f(t, c(t)) - f(t, \widetilde{c}(t))\| \leq L \cdot \|c(t) - \widetilde{c}(t)\| \tag{2.10}$$

$\forall (t, c(t)), (t, \widetilde{c}(t)) \in\ U.$

Theorem 2.1.1. *Theorem of uniqueness*
Let G be a subset of $\mathbb{R} \times \mathbb{R}^N$ and let $f\colon G \rightarrow \mathbb{R}^N$ be a continuous function, fulfilling the local Lipschitz condition. Furthermore let $\phi, \psi\colon \mathrm{I} \rightarrow \mathbb{R}^N$ be two solutions of the ODE (2.1). In case that

$$\phi(t_0) = \psi(t_0) \quad \text{for one } t_0 \in I \tag{2.11}$$

then

$$\phi(t) = \psi(t) \quad \text{for all } t \in I \tag{2.12}$$

becomes essential.

Theorem 2.1.2. *Theorem of existence from Picard and Lindelöf*
There is an $\epsilon > 0$ and a solution

$$\phi : [t_0 - \epsilon, t_0 + \epsilon] \rightarrow \mathbb{R}^N \tag{2.13}$$

of the ODE (2.1) with the initial value

$$\phi(t_0) = g \quad \forall (t_0, g) \in G, \tag{2.14}$$

if $G \subset \mathbb{R} \times \mathbb{R}^N$ is an open set and $f\colon G \rightarrow \mathbb{R}^N$ is a continuous function, fulfilling the local Lipschitz condition.

Detailed proofs of both theorems can be found in *Forster* [1999].

Remark 2.1.1.
ϕ is uniquely determined by the initial value $\phi(t_0) = g$ and theorem (2.1.1).

2.2 Numerical methods

Unfortunately, not all initial value problems have analytical solutions and hence it is necessary to use numerical methods in order to solve them.

Remark 2.2.1.
In the following, the variable t denotes the time.

Definition 2.2.1.
A numerical method is a procedure that computes an approximated solution $c^n(t_n)$ of the analytical solution $c(t)$ on a discrete grid I_h of the interval I for every $t \in I$ and every $t_n \in I_h$.

Remark 2.2.2.
For all observed numerical schemes and test cases, which will be introduced in chapter 4, the following notations are taken:

1. c^n is the abbreviation of $c^n(t_n)$
2. c^{n+1} is the abbreviation of $c^{n+1}(t_{n+1})$
3. c_i^n is the i-th component of the approximation c^n, $i = 1, \dots, N$
4. c_i^{n+1} is the i-th component of the approximation c^{n+1}, $i = 1, \dots, N$
5. $c^{(k)}$ is an intermediate step, $k = 1, 2, \dots$, at the old time step t_n
6. $c_i^{(k)}$ is the i-th component of the intermediate step, $i = 1, \dots, N$
7. $h = t_{n+1} - t_n$ is the step size, $h \in \mathbb{R}_{\geq 0}$

Generally, in mathematics two kinds of numerical methods are distinguished, *explicit* and *implicit* ones. These are defined as follows:

Definition 2.2.2.
A numerical method is called *explicit*, if the approximated solution c^{n+1} at the new time step t_{n+1} is given as function of the approximated solutions
$c^n, c^{n-1}, \dots, c^0$ at the old time steps $t_n, t_{n-1,}, \dots, t_0$. Otherwise it is called *implicit*, i.e. c^{n+1} is given as the solution of a (non-) linear equation system.

2.2.1 One step methods

In order to solve the initial value problem (2.4) two kinds of methods are used: *one step methods (OSMs)* and multi step methods. In the following the focus is on the OSMs. In contrast to the multi step methods, the approximated solution c^{n+1} of an OSM does only depend on c^n, and not on the previous solution-approximations $c^{n-1}(t), c^{n-2}(t), \dots, c^0(t)$. From definition 2.2.2 the following definition of an explicit OSM can be derived:

Definition 2.2.3.
The *explicit OSM* is of the form:

$$c^{n+1} = c^n + h_n \cdot \varphi(t_n, c^n, h_n) \tag{2.15}$$

where $\varphi : [t_0, t_e] \times \mathbb{R}^N \times \mathbb{R}_+ \to \mathbb{R}$ denotes the *increment function*, determining the OSM. Such methods give approximations c^n of the exact solution $c(t_n)$ at each grid point in $[t_0, t_n]$.

In general, error estimations play an important role because as mentioned above, numerical schemes give only approximations of the solution and hence discretisation errors evolve in each step. The following definitions for OSMs are made, to quantify the occurring errors:

Definition 2.2.4.
An OSM has the *order* $p \geq 1$ *of convergence*, if the global discretisation error ε, also called global method error

$$\varepsilon = \max_{l=0,\ldots,n} \|c^l(t) - c(t_l)\|_2, \tag{2.16}$$

fulfils the following inequality:

$$\varepsilon \leq C\cdot h_{max}^p, \tag{2.17}$$

where

$$h_{max} = \max_{l=0,\ldots,n} \{t_{l+1} - t_l\}, \tag{2.18}$$

$C \in \mathbb{R}_{\geq 0}$ is a grid independent constant and $\|\cdot\|_2$ denotes the Euclidean norm, which is defined as follows

$$\|\underline{c}\|_2 := \sqrt{\sum_{i=1}^{N} c_i^2}, \tag{2.19}$$

for each vector $\underline{c} \in \mathbb{R}^N$.

Definition 2.2.5.
The *truncation error* of a OSM is defined as

$$\tau(t,h) := \frac{c(t+h) - c(t)}{h} - \varphi(t, c(t), h). \tag{2.20}$$

The OSM with increment function φ is called *consistent*, if

$$\lim_{h\to 0} \tau(t,h) = 0 \quad \forall\, t \in I \tag{2.21}$$

and furthermore the method has *order* $p \geq 1$ *of consistency*, if

$$\tau(t,h) = \mathcal{O}(h^p), \quad h \to 0 \tag{2.22}$$

where $\mathcal{O}(\cdot\,)$ is the *Landau symbol*, see *Strehmel and Weiner* [1995].

Euler forward and Runge Kutta method

Two well known examples of OSMs are the explicit *Euler method (EM)* and the *Runge Kutta method* (RKM).
The EM, developed by Euler in 1768, is the oldest and easiest numerical method for solving initial value problems. It locally replaces the unknown solution by the well known tangent - the right hand side of the equation.
The increment function of the EM is given by:

$$\varphi(t, c(t), h) = f(t, c(t)). \tag{2.23}$$

Generally, the evaluation of the order of consistency is derived by comparing the Taylor series of the solution with the Taylor series of the increment function. For the EM that means:

$$\begin{aligned} c(t+h) &= c(t) + h\cdot \frac{dc}{dt}(t) + \mathcal{O}(h^2) \\ &= c(t) + h\cdot f(t, c(t)) + \mathcal{O}(h^2) \end{aligned} \tag{2.24}$$

Rearranging equation (2.24) to the following

$$\mathcal{O}(h) = \frac{c(t+h) - c(t)}{h} - f(t, c(t)) = \frac{c(t+h) - c(t)}{h} - \varphi(t, c(t), h) \tag{2.25}$$

the order of consistency equals 1.
The Runge Kutta methods have been constructed by Runge (1895) and Heun (1900) and finally formulated by Kutta (1901) as in the following definition.

Definition 2.2.6.
An *s-stage Runge Kutta method* has the following increment function:

$$\varphi(t, c(t), h) = \sum_{i=1}^{s} b_i\cdot k_i \tag{2.26}$$

where

$$\begin{aligned} k_1 &= f(t, c(t)) \\ k_2 &= f(t + d_2\cdot h, c(t) + h\cdot a_{21}\cdot k_1) \\ k_3 &= f(t + d_3\cdot h, c(t) + h\cdot (a_{31}\cdot k_1 + a_{32}\cdot k_2) \\ &\vdots \\ k_s &= f(t + d_s\cdot h, c(t) + h\cdot \sum_{i=1}^{s-1} a_{si}\cdot k_i) \end{aligned} \tag{2.27}$$

and d_i, a_{ij} and b_i are scalar variables, generally tabulated in so-called *Butcher Tableaus*, see *Hairer et al.* [1991] and table 2.2.1.

$$
\begin{array}{c|ccccc}
0 & & & & & \\
d_2 & a_{21} & & & & \\
d_3 & a_{31} & a_{32} & & & \\
\vdots & & & & & \\
d_s & a_{s1} & a_{s2} & \dots & a_{s,s-1} & \\
\hline
 & b_1 & b_2 & \dots & b_{s-1} & b_s
\end{array}
$$

Table 2.1: Butcher tableau of an s-stage RKM.

Remark 2.2.3.

1. The Euler forward method is a 1-stage RKM with $b_1 = 1$.
2. The *RKM of 2nd order (RK2)* is also called *Heun method* and has the increment function
$$\varphi(t, c(t), h) = \frac{1}{2}k_1 + \frac{1}{2}k_2 \tag{2.28}$$
where
$$\begin{aligned} k_1 &= f(t, c(t)) \\ k_2 &= f(t + h, c(t) + h \cdot k_1). \end{aligned} \tag{2.29}$$
3. The *classical RKM (RK4)* is a four-stage OSM. It has the increment function
$$\varphi(t, c(t), h) = \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \tag{2.30}$$
where
$$\begin{aligned} k_1 &= f(t, c(t)) \\ k_2 &= f(t + \frac{1}{2} \cdot h, c(t) + h \cdot \frac{1}{2} \cdot k_1) \\ k_3 &= f(t + \frac{1}{2} \cdot h, c(t) + h \cdot \frac{1}{2} \cdot k_2) \\ k_4 &= f(t + h, c(t) + h \cdot k_3) \end{aligned} \tag{2.31}$$
and the order of consistency equals 4, as well as the number of steps, see *Hairer et al.* [1991].

2.3 Stiffness and stability

In general, ODEs are distinguished in *stiff* and *non-stiff*. Unfortunately, there is no comprehensive and mathematical definition of stiffness so far, however a good argument

for calling an ODE stiff is the better performance of implicit numerical methods in contrast to explicit ones, as mentioned in the historically first opinion of Curtiss and Hirschfelder 1952.
A well known example for that is given by the comparison of the performance of the explicit and implicit EM applied to the simple stiff ODE

$$\frac{dc}{dt}(t) = \lambda \cdot c(t) \quad \lambda < 0, \tag{2.32}$$

see Figure 2.1 and 2.2. The analytical solution of equation (2.32)

$$c(t) = e^{\lambda t} \cdot c_0 \tag{2.33}$$

is also depicted in this Figure. The order of stiffness of equation (2.32) strongly depends on the coefficient λ.

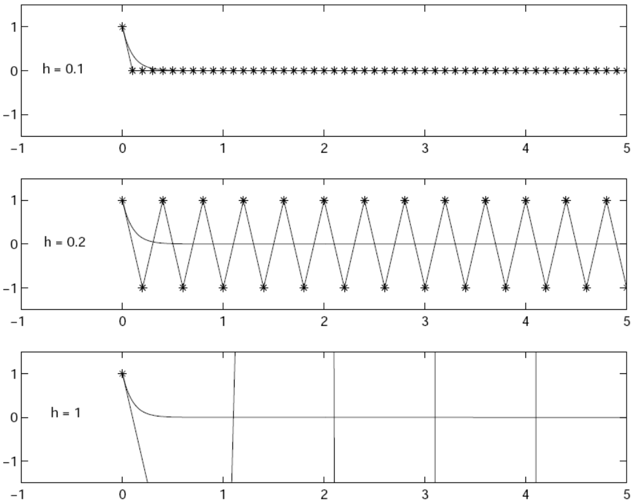


Figure 2.1: explicit EM applied to equation (2.32) with $\lambda = -10$ and step size $h = 0.1$ in the upper plot, $h = 0.2$ in the middle plot and $h = 1$ in the lower plot. The x-axis shows the dimensionless time and the y-axis the co-domain. The solid line shows the analytical solution and the star line gives the approximated solution, cp. *Simeon*.

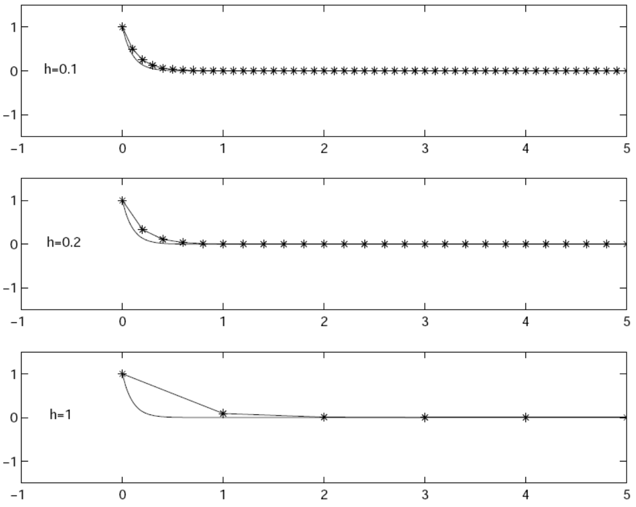


Figure 2.2: implicit EM applied to equation (2.32) with $\lambda = -10$ and step size $h = 0.1$ in the upper plot, $h = 0.2$ in the middle plot and $h = 1$ in the lower plot. The x-axis shows the dimensionless time and the y-axis the co-domain. The solid line shows the analytical solution and the star line gives the approximated solution, cp. *Simeon*.

Figure 2.1 shows the results of the explicit EM with $\lambda = -10$ and the effect of too large step sizes. The plots make clear that this method only gives accurate results, if the step size h is chosen small enough ($h = 0.1$). For $h = 0.2$ the approximated solution strongly oscillates and for $h = 1$ it even blows up. In contrast to that the implicit EM gives accurate results for all time steps, which can be seen in the three plots of Figure 2.2.
An analyse of the discretisation of equation (2.32) confirms the step size restriction for the explicit in contrast to the implicit EM.

- First, the attention is on the explicit EM. Applying the discretisation formula to equation (2.32) results in:

$$\begin{aligned} c^{n+1} &= c^n + h\cdot \lambda\cdot c^n \\ &= (1 + h\cdot \lambda)\cdot c^n. \end{aligned} \tag{2.34}$$

  In order to get the decreasing approximation of the analytical solution in case 1, the following inequality has to be fulfilled:

$$|1 + h\cdot \lambda| \leq 1. \tag{2.35}$$

  That means, if the step size does not fulfil this restriction for any given λ, the approximation will increase and that implies divergence.

- Second, the implicit EM is applied to equation (2.32) leading to:

$$\begin{aligned} c^{n+1} &= c^n + h\cdot \lambda\cdot c^{n+1} \\ &= \frac{1}{1 - h\cdot \lambda}\cdot c^n. \end{aligned} \tag{2.36}$$

The consequential inequality

$$\left|\frac{1}{1 - h \cdot \lambda}\right| \leq 1 \tag{2.37}$$

is fulfilled for any $h > 0$, because of the condition $\lambda < 0$. That means the approximation always decreases.

The advantage of implicit versus explicit methods is not the only property of stiff ODEs. Important factors are also the Jacobian matrix, such as the dimension of the system or the integration interval. More details to this topic can be found in *Hairer and Wanner* [1991].
In order to benchmark and analyse numerical methods for solving stiff problems they are applied to the *Dahlquist test equation*

$$\frac{dc}{dt}(t) = \lambda \cdot c(t) \quad \lambda \in \mathbb{C},\ \Re(\lambda) \leq 0, \tag{2.38}$$

where $\Re(\lambda)$denotes the real part of λ.
Solving equation (2.38) for c^{n+1} with the initial value $c_0 = 1$ results in

$$c^{n+1} = \mathcal{R}(z)c^n(t) \tag{2.39}$$

for any OSM, where $\mathcal{R}(z)$ denotes the *stability function* with $z = h\lambda$.
The set

$$S = \{z \in \mathbb{C} : |\mathcal{R}(z)| \leq 1\} \tag{2.40}$$

is called the *stability domain* of the method and with that notation a new definition of stability has been suggested, see e.g. *Hairer and Wanner* [1991] or *Strehmel and Weiner* [1995].

Definition 2.3.1.
A numerical method with step size $h > 0$ is called

1. *A-stable*, if there are no restrictions for Dahlquist's model problem, i.e. the application to

$$\frac{dc}{dt}(t) = \lambda c(t), \quad \Re(\lambda) \leq 0 \tag{2.41}$$

results in a sequence of approximated solutions that is restricted by $\|c_0\|$.

2. *L-stable*, if it is *A-stable* and if in addition the following equation is valid

$$\lim_{z\to\infty} R(z) = 0. \tag{2.42}$$

Remark 2.3.1.

1. If the left half-plane is part of stability domain S, than the numerical method is *A-stable.*

2. All explicit Runge Kutta methods are not A-stable, because their stability domains are only subsets of the left half-plane, as can be seen in Figure 2.3.

3. All explicit methods are not appropriated for solving stiff problems, cp. *Strehmel and Weiner* [1995].

Examples of the stability functions and domains are presented in Table 2.2 and plotted in Figure 2.3. The Table shows the stability functions of the EM, the RK2 and RK4, as well as their stability domains. In Figure 2.3 the stability domains of the explicit RKMs of 1st to 4th order are depicted.

method	stability function $\mathcal{R}(z)$	stability domain S
EM	$1+z$	$\{z \in \mathbb{C} : \lvert 1+z \rvert \leq 1\}$
RK2	$1+z+\frac{z^2}{2}$	$\left\{z \in \mathbb{C} : \lvert 1+z+\frac{z^2}{2} \rvert \leq 1\right\}$
RK4	$1+z+\frac{z^2}{2}+\frac{z^3}{6}+\frac{z^4}{24}$	$\left\{z \in \mathbb{C} : \lvert 1+z+\frac{z^2}{2}+\frac{z^3}{6}+\frac{z^4}{24} \rvert \leq 1\right\}$

Table 2.2: Stability functions and domains of the EM, RK2 and RK4.

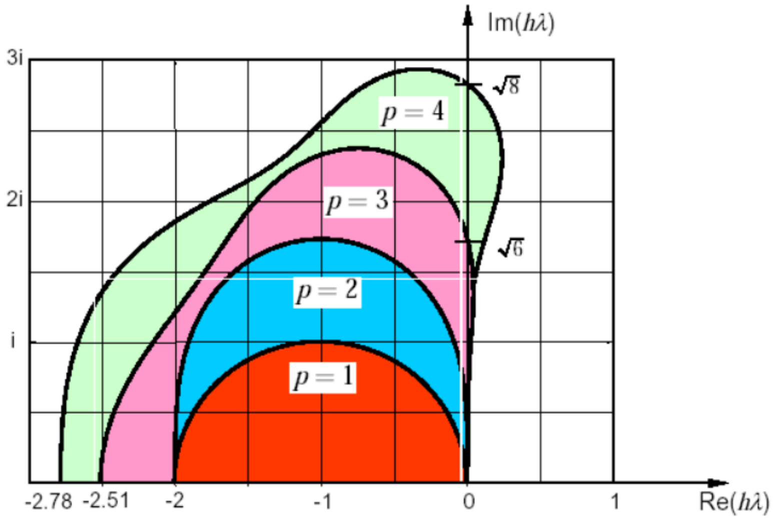


Figure 2.3: Stability domain of the explicit RK methods of order p: p=1 EM, p=2 RK2, p=3 3-stage RKM and p=4 RK4 . The coordinate system shows the complex plane where the x-axis denotes the real part of z and the y-axis the imaginary part of z. All stability domains are subsets of the left half-plane, cp. *Seiler* [2006].

3 Rosenbrock methods

In order to determine the approximated solution of stiff problems numerical schemes generally use some implicit discretisation formula for reason of numerical stability, cp. *Janelli and Fazio* [2006]. Consequently, a system of non-linear equations has to be solved and the most reliable approach for that is to apply *Newton's method*

$$t_{n+1} = t_n - \frac{f(t_n)}{f'(t_n)}, \tag{3.1}$$

which unfortunately demands an evaluation of the user specified Jacobian matrix at each iteration, see *Janelli and Fazio* [2006]. In order to get around this time consuming procedure *Rosenbrock* [1963] implemented the Jacobian matrix directly into the numerical integration formula. This idea results in a generally accepted integration formula - the so-called *Rosenbrock method (RBM)*, see *Hairer and Wanner* [1991]. The RBMs are also called *linear implicit or semi-implicit Runge Kutta methods*, due to the fact that they are derived from the fully implicit RKMs.

3.1 Derivation

An s-stage diagonal implicit Runge Kutta scheme is given by

$$\begin{aligned} k_l &= h \cdot f\left(c^n + \sum_{j=1}^{l-1} a_{lj}k_j + a_{ll}k_l\right) \qquad l = 1,..,s \\ c^{n+1} &= c^n + \sum_{l=1}^{s} b_l k_l, \end{aligned} \tag{3.2}$$

with the coefficients a_{lj}, b_l identical to the explicit scheme.
If equation (3.2) is regarded as a root problem of a function $g : \mathbb{R}^N \to \mathbb{R}^N$ defined as

$$g(k_l) = k_l - h \cdot f\left(c^n + \sum_{j=1}^{l-1} a_{lj}k_j + a_{ll}k_l\right) \tag{3.3}$$

the root can be computed with the iteration formula of Newton's method:

$$k_l^{(n+1)} = k_l^{(n)} - g'\left(k_l^{(n)}\right)^{-1} \cdot g\left(k_l^{(n)}\right) \qquad n = 0, 1, \ldots \quad . \tag{3.4}$$

The evaluation of $g'\left(k_l^{(n)}\right)^{-1}$ gives:

$$\left(\mathbf{I} - h \cdot a_{ll} \cdot f'\left(c^n + \sum_{j=1}^{l-1} a_{lj}k_j + a_{ll}k_l^{(n)}\right)\right)^{-1} \tag{3.5}$$

where $\mathbf{I}$ is the n dimensional identity matrix. Rearranging equation (3.4), where

$$\mathbf{J} := f'(c^n) \tag{3.6}$$

is the abbreviation of the approximation of the Jacobian matrix

$$f'\left(c^n + \sum_{j=1}^{l-1} a_{lj}k_j + a_{ll}k_l^{(n)}\right), \tag{3.7}$$

ends in a linear equation system of the form

$$(\mathbf{I} - h\cdot a_{ll}\cdot \mathbf{J})\cdot k_l^{(n+1)} = h\cdot f\left(c^n + \sum_{j=1}^{l-1} a_{lj}k_j + a_{ll}k_l^{(n)}\right) - h\cdot a_{ll}\cdot \mathbf{J}\cdot k_l^{(n)} \tag{3.8}$$

that can be uniquely resolved for small values of h. This linearisation presents the main idea of the RBMs.

It is not necessary to make more than one step of Newton's iteration to obtain good accuracy and hence together with the initial value

$$k_l^{(0)} = -\frac{1}{a_{ll}}\cdot \sum_{j=1}^{l-1} \gamma_{lj}k_j, \tag{3.9}$$

which is chosen as linear combination of the known $k-$values, and the following notations

$$k_l := k_l^{(1)}, \quad \alpha_{lj} := a_{lj} - \gamma_{lj}, \quad \gamma_{ll} := a_{ll}, \tag{3.10}$$

equation (3.2) can be converted into

$$\begin{aligned} (\mathbf{I} - h\cdot \gamma_{ll}\cdot \mathbf{J})\cdot k_l &= h\cdot f\left(c^n + \sum_{j=1}^{l-1} \alpha_{lj}k_j\right) - h\cdot \mathbf{J}\cdot \sum_{j=1}^{l-1} \gamma_{lj}k_j \\ c^{n+1} &= c^n + \sum_{l=1}^{s} b_l k_l. \end{aligned} \tag{3.11}$$

System 3.11 consists of a sequence of s linear equations, which have to be solved to compute the k_l. Thus, the formula of the RBMs is derived and the following definition is valid:

Definition 3.1.1.

An s-stage Rosenbrock method is given as

$$\begin{aligned} k_l &= h\left(f\left(c^n + \sum_{j=1}^{l-1} \alpha_{lj}k_j\right) + \mathbf{J}\sum_{j=1}^{l} \gamma_{lj}k_j\right), \quad l = 1,..,s \\ c^{n+1} &= c^n + \sum_{l=1}^{s} b_l k_l \end{aligned} \tag{3.12}$$

where α_{lj}, γ_{lj}, b_l are the determining coefficients.

Remark 3.1.1.
RBMs are applicable to non-autonomous ODEs, because they can be put in autonomous form by augmenting

$$\frac{dt}{dt} = 1, \tag{3.13}$$

i.e. treating t as a dependent variable.

3.2 Reducing computational effort

The most expensive procedures of the RBM are on the one hand the LU decomposition of $(\mathbf{I} - h\gamma\mathbf{J})$, though it possibly is sparsely populated. However, assuming $\gamma_{ll} = \gamma$ for all l, only one LU-factorisation per step is needed, because the same matrix $(\mathbf{I} - h\gamma\mathbf{J})$ is used to evaluate all k_l. On the other hand the matrix-vector multiplication takes a lot of the computing time. In order to avoid it, the following notation is introduced

$$u_l = \sum_{j=1}^{l} \gamma_{lj} k_j, \tag{3.14}$$

which leads to a new formula for computing k_l:

$$k_l = \frac{1}{\gamma} \cdot u_l - \sum_{j=1}^{l-1} g_{lj} u_j, \quad l = 1, \dots, s. \tag{3.15}$$

The substitution of (3.15) into (3.14) yields

$$\begin{aligned} (\mathbf{I} - h\gamma\mathbf{J}) u_l &= h \left(f \left(c^n + \sum_{j=1}^{l-1} a_{lj} u_j \right) + \mathbf{J} \sum_{j=1}^{l-1} g_{lj} u_j \right), \quad l = 1, \dots, s \\ c^{n+1} &= c^n + \sum_{j=1}^{s} m_j u_j, \end{aligned} \tag{3.16}$$

with

$$\begin{aligned} G &= \operatorname{diag}\left\{\gamma^{-1}, \dots, \gamma^{-1}\right\} - \Gamma^{-1} \\ \Gamma &= (\gamma_{lj}) \\ a_{lj} &= (\alpha_{lj})\Gamma^{-1} \quad \text{and} \\ (m_1, \dots, m_s) &= (b_1, \dots, b_s)\Gamma^{-1}. \end{aligned} \tag{3.17}$$

Formula (3.16) also avoids n^2 multiplications for $h\gamma\mathbf{J}$, cp. *Hairer and Wanner* [1991].

3.3 Consistency and stability

The order of consistency as well as the stability properties are the main points characterising the performance of an integration scheme and hence they are defined here for RBMs. The RBMs are OSMs and thus the error definitions 2.2.4 and 2.2.5 from page 9 are still valid. Further statements about the consistency behaviour of the RBMs can be made:

p	number	order conditions
1	1	$\sum_l b_l = 1$
2	2	$\sum_k \beta_{jk} = \frac{1}{2} - \gamma$
3	3	$\sum_{k,l} \alpha_{jk}\alpha_{jl} = \frac{1}{3}$
	4	$\sum_{k,l} \beta_{jk}\beta_{kl} = \frac{1}{6} - \gamma + \gamma^2$
4	5	$\sum_{klm} \alpha_{jk}\alpha_{jl}\alpha_{jm} = \frac{1}{4}$
	6	$\sum_{klm} \alpha_{jk}\beta_{kl}\alpha_{jm} = \frac{1}{8} - \frac{\gamma}{3}$
	7	$\sum_{klm} \beta_{jk}\alpha_{kl}\alpha_{km} = \frac{1}{12} - \frac{\gamma}{3}$
	8	$\sum_{klm} \beta_{jk}\beta_{jk}\beta_{lm} = \frac{1}{24} - \frac{\gamma}{2} + \frac{3\gamma^2}{2} - \gamma^3$

Table 3.1: Order conditions for RBMs up to order 4.

Remark 3.3.1.

- The truncation error of the RBM is only of size

$$\mathcal{O}\left(\frac{h^2}{z}\right), \tag{3.18}$$

  if the coefficients of a RBM satisfy

$$\begin{aligned} \alpha_{si} + \gamma_{is} &= b_i \quad \text{and} \\ \alpha_s &= 1, \end{aligned} \tag{3.19}$$

  for $i = 1, \ldots, s$. This implies that the RBMs asymptotically reach the exact solution for $z \to \infty$. For more details see *Hairer and Wanner* [1991].

- For obtaining the order p of consistency, the coefficients of a RBM have to fulfil special conditions up to the desired order, see table 3.1 and compare e.g. *Hairer and Wanner* [1991], with the following abbreviations:

$$\begin{aligned} \alpha_l &= \sum_{j=1}^{l-1} \alpha_{lj} \\ \beta_{lj} &= \sum_{j=1}^{l-1} \alpha_{lj} + \gamma_{lj}. \end{aligned} \tag{3.20}$$

The RBMs are derived from implicit RKMs, as shown above, and hence they are also suitable for solving stiff problems. The stability function can be obtained by applying the

RBMs to *Dahlquist's model problem* (see page 14), which yields a rational function of the form

$$\mathcal{R}(z) = \frac{P(z)}{(1-\gamma z)^{s'}}, \tag{3.21}$$

where $P(z)$ is a polynomial of degree s', $s' \leq s$, cp. *Sandu et al.* [1997b]. In case of *L-stability* that is focused here, the degree is less than or equal to $s' - 1$.

3.4 Step size strategy

The achievement of numerical methods to solve stiff problems depends on the use of adaptive step size mechanisms controlling the truncation error. For the RBMs the following strategy is applied, cp. *Hairer and Wanner* [1991]:
Let $\tilde{c}^{n+1}$ be the solution of the embedded Rosenbrock formula that is given by

$$\begin{aligned} k_l &= h\left(f\left(c^n + \sum_{j=1}^{l-1} \alpha_{lj} k_j\right) + \mathbf{J}\sum_{j=1}^{l} \gamma_{lj} k_j\right), \quad l = 1, .., s \\ \tilde{c}^{n+1} &= c^n + \sum_{l=1}^{\tilde{s}} \tilde{b}_l k_l \quad \tilde{s} \leq s. \end{aligned} \tag{3.22}$$

Note, the only difference from formula (3.12) is the choice of the weights $\tilde{b}_i$. These are chosen to achieve $\tilde{p} = p - 1$ as order of consistency, where p is the order of c^{n+1}, i.e.:

$$\begin{aligned} c^{n+1} &= c^n(t_n + h) + \mathcal{O}(h^{p+1}) \\ \tilde{c}^{n+1} &= \tilde{c}^n(t_n + h) + \mathcal{O}(h^{\tilde{p}+1}). \end{aligned} \tag{3.23}$$

Taking the difference of c^{n+1} and $\tilde{c}^{n+1}$, the *local error estimator*

$$Est := \tilde{c}^{n+1} - c^{n+1} \tag{3.24}$$

is defined. This value is an estimation of the main part of the local discretisation error of the method with order q= min $(p, \tilde{p})$. The order of a pair of equations for c^{n+1} and $\tilde{c}^{n+1}$, described in the formulas (3.12) and (3.22), is denoted by $p(\tilde{p})$, as done in *Sandu et al.* [1997b].
Additionally, let n be the dimension of the ODE system, see in chapter 2, and *atol* and *rtol* the user-specified *absolute* and *relative error* tolerances. The tolerances occurring in each step are denoted by

$$Tol_i = atol + rtol \cdot |c_i^{n+1}|, \quad i = 1, \ldots, N. \tag{3.25}$$

Taking

$$err = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{Est}{Tol_i}\right)^2} \tag{3.26}$$

as a measure we find an optimal step size h_{opt} by comparing err to 1 and using the relations $err \approx Ch^{q+1}$ and $1 \approx Ch_{opt}^{q+1}$. Thus we obtain the optimal step size as

$$h_{opt} = h \cdot \left(\frac{1}{err} \right)^{\frac{1}{q+1}} . \tag{3.27}$$

The new step size proposal

$$h_{new} = h \cdot \min \left\{ facmax, \max \left\{ facmin, fac \cdot \left(\frac{1}{err} \right)^{\frac{1}{q+1}} \right\} \right\} \tag{3.28}$$

is obtained by using err with q as order of consistency instead of p. The integration of the growth factors *facmax* and *facmin* to equation (3.27) prevents for too large step increase and contribute to the safety of the code. Additionally, using the safety factor *fac* makes sure that err will be accepted in the next step with high probability. The step is accepted, in case that $err \leq 1$ otherwise it is rejected and then the procedure is redone. In both cases the new solution is computed with h_{new} as step size, decreased by a factor of ten, if there are two consecutive rejection steps. Generally, the new step size is constrained by a user-specific maximum h_{max}. According to *Hairer et al.* [1991] and the references therein, the maximal growth factor *facmax* should be set to 1 right after a rejection step.

4 Application of numerical methods in biogeochemical models

Marine biogeochemical modelling is a section of earth system science in which processes that govern the fluxes or cycling of energy or matter in the ocean are simulated with computer models. In contrast to physical oceanography there are no fundamental equations and thus, depending on the system under study, biogeochemical models differ significantly with respect to the type and number of state variables, processes and functions. A widely used model is the so called *NPZD*-type model, see e.g. *Evans and Parslow* [1985] or *Fasham et al.* [1990], where N denotes the pool of nutrients, P phytoplankton, Z zooplankton and D detritus. The fluxes between these state variables are biogeochemical processes including nutrient uptake by phytoplankton, grazing of herbivorous zooplankton and mortality, excretion of zooplankton and remineralisation of dead organic matter into nutrients.

In order to describe the processes occurring in biogeochemical models *partial differential equations (PDEs)* are used. By applying the *operational split methods* the PDEs can simply be subdivided into integrable pieces, which are successively solved, see *Hairer et al.* [2006]. Furthermore, this method has good accuracy properties by numerically approximating the solution.

Example 4.0.1.
Let c be the vector of concentrations, t the time and x the location. The linear advection equation with constant velocity u is given by

$$\frac{\partial c}{\partial t}(t) + u\frac{\partial c}{\partial x}(t) = -ac(t). \tag{4.1}$$

Applying the split method to equation (4.1) results in

$$\begin{aligned} \frac{c_i^{n+\frac{1}{2}} - c_i^n}{\Delta t} + u\frac{c_i^n + c_{i-1}^n}{\Delta x} &= 0 && (4.2)\\ \frac{c_i^{n+1} - c_i^{n+\frac{1}{2}}}{\Delta t} \qquad\qquad &= -ac_i^{n+1}, \quad i = 1, \dots, N && (4.3) \end{aligned}$$

where the first equation denotes the so-called *advection step* and the second equation denotes the *process step.* Taken the sum of both leads to the discretisation of the whole advection equation.

Remark 4.0.1. In the following the index i ranges from 1 to N.

4.1 Production-destruction equation systems

The ODE calculated in an operational split method for the complete biogeochemical model is of high numerical relevance. It remains

$$\frac{dc_i(t)}{dt} = P_i(\underline{c}(t)) - D_i(\underline{c}(t)), \tag{4.4}$$

where $\underline{c}(t) = (c_1(t), \dots, c_N(t))^T$ denotes the vector of concentrations. The right hand side describes the fluxes, where $P_i(\underline{c}(t))$ and $D_i(\underline{c}(t))$ represent the production (source) and destruction (sink) rates of the i-th constituent. Both may depend either linearly or non-linearly on $\underline{c}(t)$ and can be rewritten as

$$\begin{aligned} P_i(\underline{c}(t)) &= \sum_{j=1}^{N} p_{ij}(\underline{c}(t)) \\ D_i(\underline{c}(t)) &= \sum_{j=1}^{N} d_{ij}(\underline{c}(t)), \end{aligned} \tag{4.5}$$

with $p_{ij}(\underline{c}(t)) \geq 0$ representing the rate at which the j-th constituent transforms into the i-th, while $d_{ij}(\underline{c}(t)) \geq 0$ denotes the rate at which the i-th constituent transforms into the j-th, cp. *Burchard et al.* [2003].

In simple *NPZD*-type models all state variables are based on the same measurable unit, e.g. carbon, and the reactive terms do only exchange mass between state variables fulfilling

$$\begin{aligned} &p_{ij}(\underline{c}(t)) = d_{ji}(\underline{c}(t)), \quad \text{for } i \neq j \quad \text{and} \\ &p_{ii}(\underline{c}(t)) = d_{ii}(\underline{c}(t)) = 0. \end{aligned} \tag{4.6}$$

Theorem 4.1.1.
Equation (4.4) guarantees the conservation of mass.

Proof.

$$\begin{aligned} \frac{d}{dt}\left(\sum_{i=1}^{N} c_i(t)\right) &= \sum_{i=1}^{N} \left(P_i(\underline{c}(t)) - D_i(\underline{c}(t)\right) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij}(\underline{c}(t)) - d_{ij}(\underline{c}(t)) \\ &= \sum_{i=1}^{N} p_{ii}(\underline{c}(t)) - d_{ii}(\underline{c}(t)) \\ &= 0. \end{aligned} \tag{4.7}$$

□

The model is positive, if the following condition holds for non-negative initial values, i.e. the solution $c_i(t)$ is greater than zero for all times t and $i = 1, \ldots, N$, see *Burchard et al.* [2003]:

$$d_{j,i}(\underline{c}(t)) \to 0 \text{ for } c_i(t) \to 0. \tag{4.8}$$

From a numerical point of view

1. positivity and
2. conservativity

are the two main characteristics of the model system. These have to be respected by the numerical schemes.

4.2 Test cases

The comparison of the numerical schemes applied to marine biogeochemical models is conducted on the basis of three test cases. These are now presented.

1 - A simple linear model

This test case, taken from *Burchard et al.* [2003] describes the mass exchange between two constituents and is given by:

$$\begin{aligned} \frac{dc_1(t)}{dt} &= c_2(t) - ac_1(t) \\ \frac{dc_2(t)}{dt} &= ac_1(t) - c_2(t) \end{aligned} \tag{4.9}$$

All components of the vector of initial values $\underline{c}(0)$ are positive as well as the dimensionless constant a. Writing system (4.9) in production-destruction notation, the terms are given as:

$$P = \begin{pmatrix} 0 & c_2(t) \\ ac_1(t) & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & ac_1(t) \\ c_2(t) & 0 \end{pmatrix},$$

and the Jacobian matrix of the system has the following form

$$\mathbf{J} = \begin{pmatrix} -a & 1 \\ a & -1 \end{pmatrix}.$$

The analytical solution of system (4.9) is

$$c(t) = \frac{1}{6} \cdot \begin{pmatrix} 1 \\ 5 \end{pmatrix} + \frac{11}{15} \cdot e^{-6 \cdot t} \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \tag{4.10}$$

where $a = 5$ and the vector of initial values is chosen as $c(0) = (0.9, 0.1)$. The graphical presentation of the analytical solution can be seen in Figure 4.1.

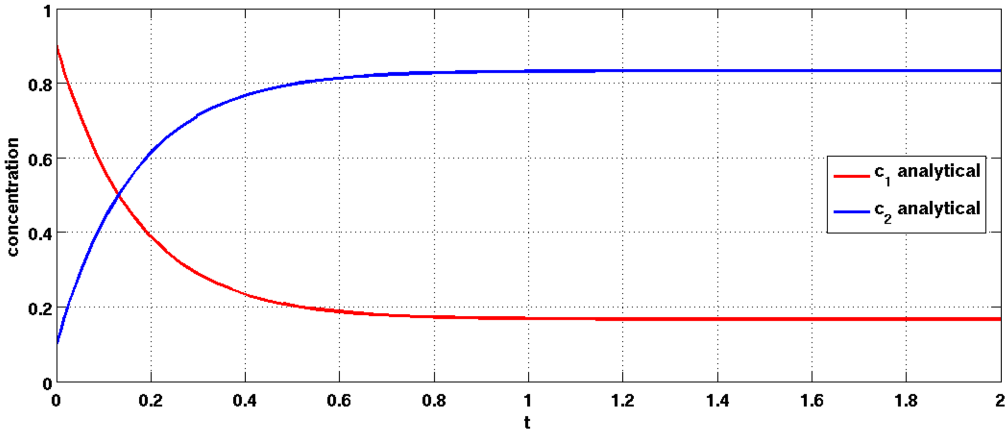


Figure 4.1: The analytical solution of the linear test case versus non-dimensional time. The vector of initial values is chosen as $\underline{c}(0) = (0.9, 0.1)$. The red line shows $c_1(t)$ and the blue line shows $c_2(t)$.

2 - A non-linear model

This test case, also taken from *Burchard et al.* [2003], describes mass exchange between three constituents and is given in the following way:

$$\begin{aligned} \frac{dc_1(t)}{dt} &= -\frac{c_1(t)}{c_1(t)+1}c_2(t) \\ \frac{dc_2(t)}{dt} &= \frac{c_1(t)}{c_1(t)+1}c_2(t) - ec_2(t) \\ \frac{dc_3(t)}{dt} &= ec_2(t). \end{aligned} \tag{4.11}$$

The production and destruction terms are:

$$P = \begin{pmatrix} 0 & 0 & 0 \\ \frac{c_1(t)}{c_1(t)+1}c_2(t) & 0 & 0 \\ 0 & ec_2(t) & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & \frac{c_1(t)}{c_1(t)+1}c_2(t) & 0 \\ 0 & 0 & ec_2(t) \\ 0 & 0 & 0 \end{pmatrix}.$$

and the Jacobian matrix of system (4.11) is given as follows:

$$\mathbf{J} = \begin{pmatrix} \frac{-c_2(t)}{(c_1(t)+1)^2} & \frac{-c_1(t)}{c_1(t)+1} & 0 \\ \frac{c_2(t)}{(c_1(t)+1)^2} & \frac{c_1(t)}{c_1(t)+1} - e & 0 \\ 0 & e & 0 \end{pmatrix}.$$

The constituents $c_1(t), c_2(t), c_3(t)$ may be interpreted as nutrient, phytoplankton and detritus and the system as a biogeochemical model for the upper oceanic layer in spring,

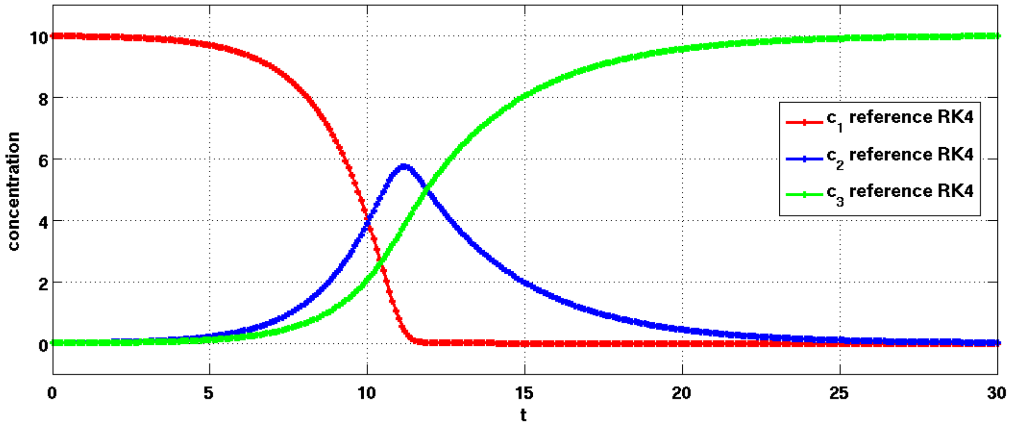


Figure 4.2: High order reference solution (RK4) of the non-linear test case, versus non-dimensional time, with step size $h = 0.5$. The red line shows c_1^n (nutrients), the blue line c_2^n (phytoplankton) and the green line c_3^n (detritus).

when nutrient rich surface water is captured in the euphotic zone, where the mineralisation of detritus is not included, cp. *Burchard et al.* [2003].
For this test case no analytical solution can be obtained and hence the RK4, see 4.15 on page 30, is used as a reference solution with $\underline{c}(0) = (9.98, 0.01, 0.01)$ as the vector of initial values. The parameter e has been chosen as 0.3 and the dimensionless step size has been set to $h = 0.1$, see Figure 4.2.

3 - The Robertson test problem

The stiff Robertson test case for chemical reactions

$$\begin{aligned}\frac{dc_1}{dt}(t) &= Ac_2(t)c_3(t) - Bc_1(t)\\ \frac{dc_2}{dt}(t) &= Bc_1(t) - Ac_2(t)c_3(t) - Cc_2(t)^2\\ \frac{dc_3}{dt}(t) &= Cc_2(t)^2\end{aligned} \tag{4.12}$$

describes the kinetics of an auto-catalytic reaction given by *Robertson* [1966]. The production and destruction terms are:

$$P = \begin{pmatrix} 0 & Ac_1(t)c_2(t) & 0 \\ Bc_1(t) & 0 & 0 \\ 0 & Cc_2(t)^2 & 0 \end{pmatrix} \quad D = \begin{pmatrix} 0 & Bc_1(t) & 0 \\ Ac_1(t)c_2(t) & 0 & Cc_2(t)^2 \\ 0 & 0 & 0 \end{pmatrix}$$

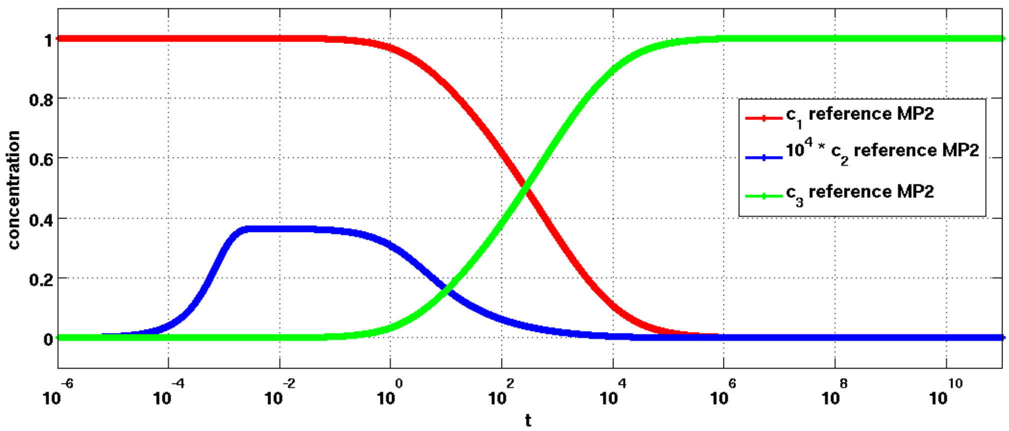


Figure 4.3: High order reference solution of the Robertson test case computed with the MP2 at a number of 241795 exponentially increasing time steps $h^n = 2 \cdot 10^{-14} \cdot 1.0002^n$ versus a non-dimensional time. For clarity c_2^n (blue line) is multiplied with a factor of 10^4. The red line shows c_1^n and the green line c_3^n.

and the Jacobian matrix of the Robertson test problem is given by

$$\mathbf{J} = \begin{pmatrix} -B & Ac_3(t) & Ac_2(t) \\ B & -Ac_3(t)2Cc_2(t) & -Ac_2(t) \\ 0 & 2Cc_2(t) & 0 \end{pmatrix}.$$

The reference solution for this test case, see Figure 4.3, is computed with the MP2 method at very short time steps. The concentrations of the chemical constituents are taken as:

$$A = 10^4 s^{-1}, \quad B = 0.04 s^{-1}, \text{ and } C = 3 \cdot 10^7 s^{-1},$$

and the vector $\underline{c}(0)$ of initial values is chosen as $(1, 0, 0)$, as done by *Burchard et al.* [2003].

4.3 Application of numerical methods

As mentioned above, mostly the ODEs that occur in biogeochemical models cannot be solved analytically, but the solutions are numerically approximated. In order to obtain satisfying results for the whole biogeochemical problem it is important and necessary to achieve accurate approximated solutions for the ODE part.

Taking into account the two characteristics mentioned on page 25, three criteria for comparing numerical schemes in biogeochemical models can be taken. These refer to the ability of the schemes:

1. to be *unconditionally positive*

   Definition 4.3.1. An integration scheme Φ is called *unconditionally positive* if $c^{n+1} > 0$ for any arbitrary time step $h > 0$ and $c^n > 0$.

2. to be *conservative*

   Definition 4.3.2. An integration scheme Φ is called *conservative* if

   $$\sum_{i=1}^{n} \left(c_i^{n+1} - c_i^n\right) = 0,$$

   for all fully conservative ODEs in form of equation (4.4), and $p_{ii}(c(t)) = d_{ii}(c(t))$.

3. to have a *high order of accuracy for low computational effort*

The former two properties are considered when comparing numerical schemes. In this study however, the main focus is on the accuracy and the computing time.

4.3.1 Explicit schemes with fixed time steps

The well known OSMs -
the EM

$$c_i^{n+1} = c_i^n + h \cdot \left(P_i(c^n) - D_i(c^n)\right) \tag{4.13}$$

and RK2

$$\begin{aligned} c_i^{(1)} &= c_i^n + h \cdot \left(P_i(c^n) - D_i(c^n)\right) \\ c_i^{n+1} &= c_i^n + \frac{h}{2} \cdot \left(P_i(c^n) + P_i(c^{(1)}) - D_i(c^n) - D_i(c^{(1)})\right) \end{aligned} \tag{4.14}$$

are often used for biogeochemical modelling, because they are conservative and have low computational effort. However, they may compute negative values for sufficiently large time steps. In order to avoid that, the use of smaller time steps is necessary, but the smaller the time step the higher the computational effort and hence the costs increase significantly.

Applying the EM to test case 1 and 2, see Figure 4.4 and 4.7, the effect of non-positivity for too long time steps can be seen. For test case 1 the scheme strongly oscillates, see Figure 4.4, and in test case 2 negative nutrient concentrations occur, which lead to mass exchange from phytoplankton to nutrient and give a artificial increase of nutrient ($t = 13$), see Figure 4.7. For test case 3 the simulation aborts after a few seconds, because negative concentrations and subsequent instabilities occur, cp. *Burchard et al.* [2003].

The RK2 uses the EM as predictor step and the disadvantages of non-positivity and non-stability can be seen in Figure 4.5 and 4.8. The numerical solution of test case 1 has low accuracy though the scheme is of order 2. The approximated solution of the non-linear model is little accurate in the region, where the predictor step is negative, that is the moment of nutrient depletion ($t \approx 11$). However, in contrast to the EM the RKM has high accuracy in reproducing the initial phase of nutrient uptake.

For the stiff Robertson test problem the simulation also aborts, as well as it happens for the EM, due to the fact that negative concentrations and instabilities occur.
The RK4:

$$\begin{aligned}
c_i^{(1)} &= c_i^n + \frac{h}{2}\left(P_i(c^n) - D_i(c^n)\right)\\
c_i^{(2)} &= c_i^n + \frac{h}{2}\left(P_i(c^{(1)}) - D_i(c^{(1)})\right)\\
c_i^{(3)} &= c_i^n + h\left(P_i(c^{(2)}(t)) - D_i(c^{(2)})\right)\\
c_i^{n+1} &= c_i^n + \frac{h}{6}\left(P_i(c^n) - D_i(c^n) + 2c^{(1)} + 2c^{(2)} + c^{(3)}\right)
\end{aligned} \tag{4.15}$$

gives more accurate results for test case 1 and 2 than the RK2, and of course it is conservative. However, applied to test case 1, see Figure 4.6 the initial phase ($t < 0.6$) has low accuracy though the scheme is of order four. The approximated solution of test case 2, see Figure 4.9, shows high accuracy properties, notwithstanding the little variations at the top of the phytoplankton bloom on the one hand and at the moment of nutrient depletion ($t \approx 11$) on the other hand. These also occur, due to the non-positivity of the schemes. Similar to the RK2, the RK4 is not suitable for solving stiff problems, because negative concentrations and instabilities occur, too.

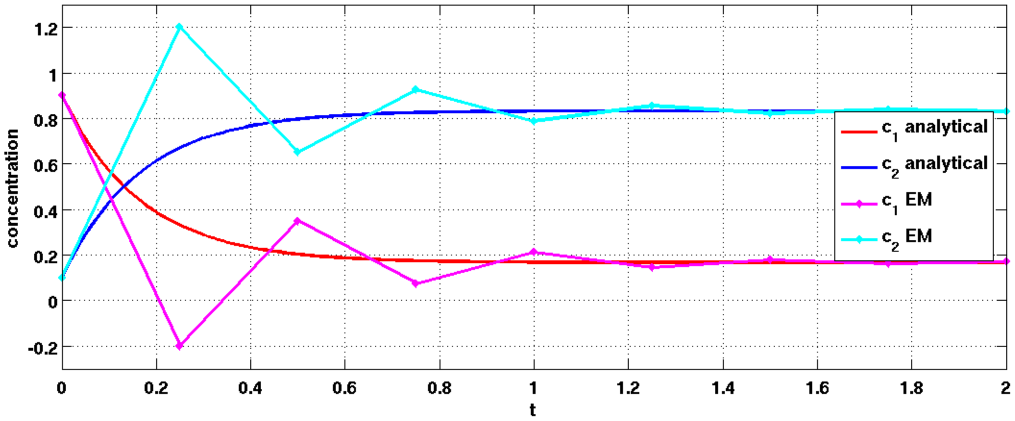


Figure 4.4: The EM with step size $h = 0.25$ is applied to test case 1 and gives the two angular lines (pink for c_1^n and cyan for c_2^n). The analytical solution is also plotted, in red ($c_1(t)$) and blue ($c_2(t)$) lines. Please note that negative values appear for the approximated solution at $t = 0.25$.

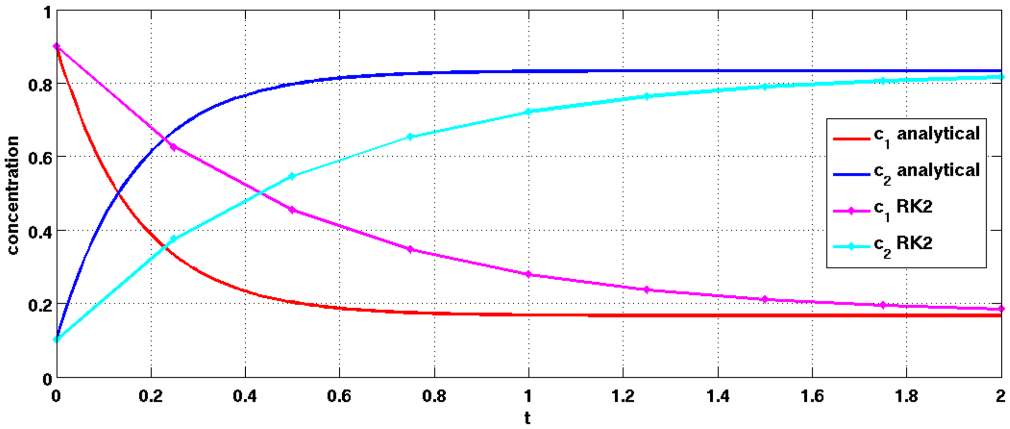


Figure 4.5: The RK2 applied to test case 1 with step size $h = 0.25$ can be seen as the two angular lines, the pink one shows c_1^n and the green one shows c_2^n, where no negative values appear. The analytical solution is simulated in red ($c_1(t)$) and blue ($c_2(t)$) lines.

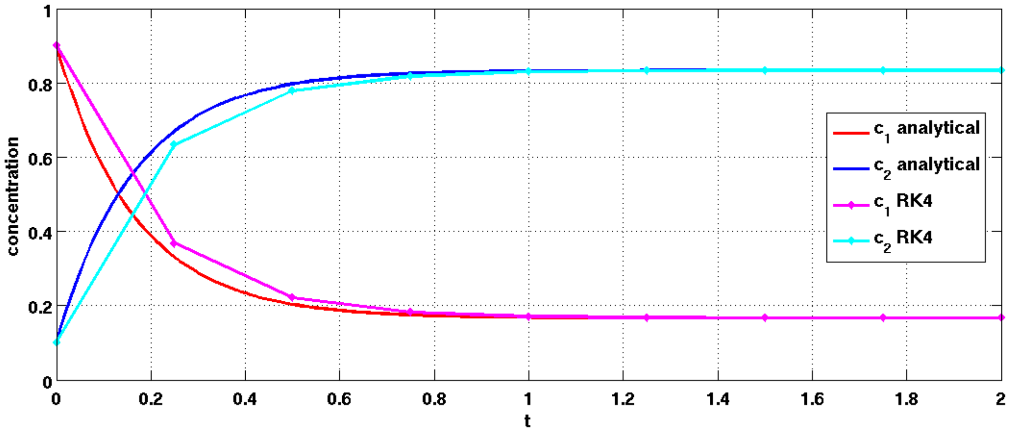


Figure 4.6: The RK4 applied to test case 1 with step size $h = 0.25$ can be seen as the two angular lines in pink (c_1^n) and green (c_2^n). The analytical solution is simulated in red ($c_1(t)$) and blue ($c_2(t)$) lines.

4.3.2 Quasi-implicit schemes with fixed time steps

As mentioned above it is necessary that numerical schemes retain the non-negativity of a model problem. This was first addressed by *Patankar* [1980] for numerical turbulence models.

Example 4.3.1. A typical model problem motivated by turbulence modelling has the following form

$$\frac{dc(t)}{dt} = P(t, c(t)) - Q(t, c(t))c(t) \tag{4.16}$$

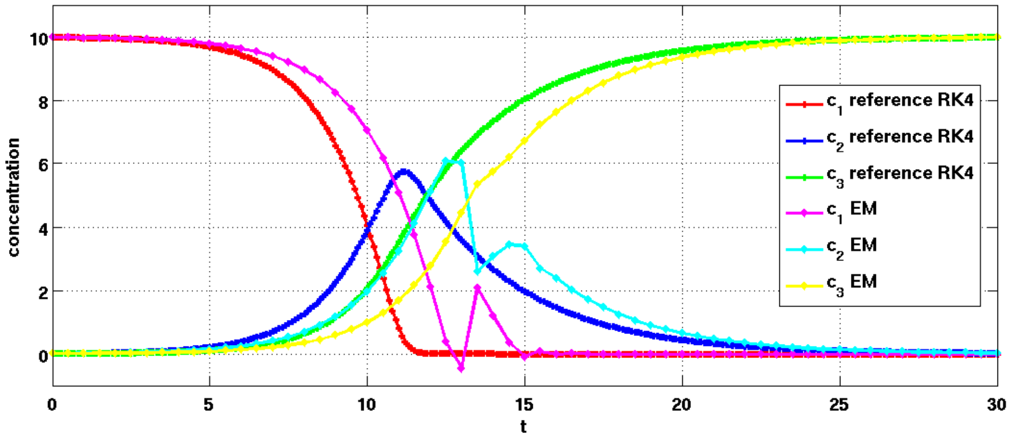


Figure 4.7: The EM with step size $h = 0.5$ applied to test case 2 gives the shifted lines (pink for c_1^n, cyan for c_2^n and yellow for c_3^n). Negative values appear for the approximated solution at e.g. $t = 13, 15$. The reference solution (RK4) is plotted in red (c_1^n), blue (c_2^n) and green (c_3^n) lines.

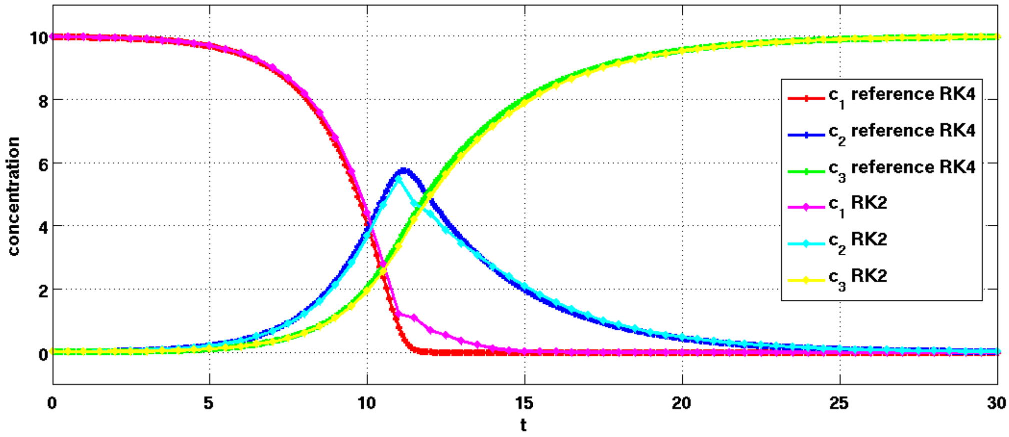


Figure 4.8: The RK2 with step size $h = 0.5$ applied to test case 2 is plotted together with the reference solution (RK4). The red, blue and green lines show the reference solution of $c_1(t), c_2(t), c_3(t)$ and in pink, cyan and yellow lines the approximated solutions of $c_1(t), c_2(t), c_3(t)$ are presented.

where c denotes an arbitrary non-negative quantity, P and Qc the non-negative source and sink terms, respectively and t denotes the time. As in *Burchard* [2002] the straight-forward in time discretisation of equation (4.16) is given by

$$\frac{c^{n+1} - c^n}{h} = P^n(t, c(t)) - Q^n(t, c(t))c^n \tag{4.17}$$

with h denoting the time step.

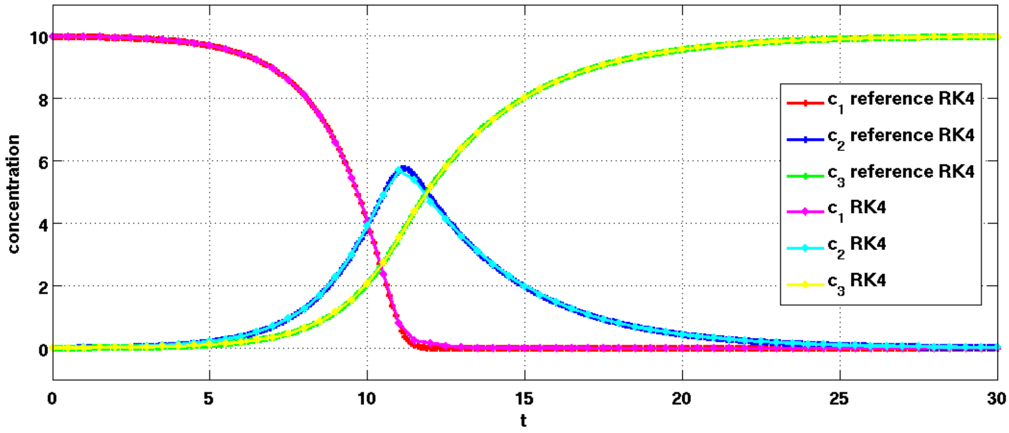


Figure 4.9: The RK4 with step size $h = 0.5$ applied to test case 2. The pink line shows the concentration of c_1^n, the cyan lines the one of c_2^n and the yellow line the one of c_3^n. The reference solution (RK4) of $c_1(t)$, $c_2(t)$ and $c_3(t)$ is depicted in red, blue and green, respectively.

Rearranging equation (4.17) gives the solution at the new time step

$$c_i^{n+1} = c_i^n\big(1 - hQ_i^n(t, c(t))\big) + hP_i^n(t, c(t)), \tag{4.18}$$

which is negative if the right hand side of equation (4.17) is negative and the time step is large with

$$h > \frac{c^n}{Q^n(t, c(t))c^n - P^n(t, c(t))}. \tag{4.19}$$

Restricting h to avoid equation (4.19) is computational unreasonable and hence the following *quasi-implicit* numerical scheme is generally used (e.g. in turbulence modelling):

$$\frac{c^{n+1} - c^n}{h} = P^n(t, c(t)) - Q^n(t, c(t))c^n \cdot \frac{c^{n+1}}{c^n} \tag{4.20}$$

This results in an always non-negative solution

$$c^{n+1} = \frac{c^n + hP^n(t, c(t))}{1 + hQ^n(t, c(t))}. \tag{4.21}$$

Motivated by this the *Patankar Euler method (P1)* of order one

$$c_i^{n+1} = c_i^n + h\left(P_i(c^n) - D_i(c^n)\frac{c_i^{n+1}}{c_i^n}\right) \tag{4.22}$$

for production-destruction equation systems was proposed by Patankar (1980), see *Patankar* [1980].

The *P1*, as well as its extended version of order two, the *Patankar Runge Kutta method (P2)*, developed by *Burchard et al.* [2003],

$$\begin{aligned} c_i^{(1)} &= c_i^n + h\left(P_i(c^n) - D_i(c^n)\frac{c_i^{(1)}}{c_i^n}\right) \\ c_i^{n+1} &= c_i^n + \frac{h}{2}\left(P_i(c^n) + P_i(c^{(1)}) - D_i(c^n) - D_i(c^{(1)})\frac{c_i^{n+1}}{c_i^{(1)}}\right) \end{aligned} \tag{4.23}$$

is not conservative, due to the fact that they are developed for turbulence modelling, where conservation is not essential and hence source and sinks terms are numerically treated in different ways. The equal numerical treatment of source and sink terms was introduced by *Burchard et al.* [2003]. The authors developed the *modified Patankar scheme* of first order, the so-called *modified Patankar Euler method (MP1)*

$$c_i^{n+1} = c_i^n + h\left[\sum_{j=1}^{N}\left(p_{ij}(c^n)\frac{c_j^{n+1}}{c_j^n}\right) - \sum_{j=1}^{N}\left(d_{ij}(c^n)\frac{c_i^{n+1}}{c_i^n}\right)\right] \tag{4.24}$$

and of second order, the so-called *modified Patankar Runge Kutta method (MP2)*

$$\begin{aligned} c_i^{(1)} &= c_i^n + h\left(\sum_{j=1}^{N} p_{ij}(c^n)\frac{c_j^{(1)}}{c_j^n} - \sum_{j=1}^{N} d_{ij}(c^n)\frac{c_i^{(1)}}{c_i^n}\right) \\ c_i^{n+1} &= c_i^n + \frac{h}{2}\left[\sum_{j=1}^{N}\left(p_{ij}(c^n) + p_{ij}(c^{(1)})\frac{c_j^{n+1}}{c_j^{(1)}}\right)\right] \\ &\quad - \frac{h}{2}\left[\sum_{j=1}^{N}\left(d_{ij}(c^n) + d_{ij}(c^{(1)})\frac{c_i^{n+1}}{c_i^{(1)}}\right)\right]. \end{aligned} \tag{4.25}$$

The MP1 is based on the traditional EM and the MP2 on the RK2. The MP2 gives accurate results for test cases 1 and 2, see Figure 4.10 and 4.12 and additionally is suitable for solving stiff problems as can be seen in Figure 4.14. There the results of the MP2 applied to test case 3 are depicted.

Additionally, Figure 4.10 depicts the advantages of the MP2 - the numerical stability as well as the positivity and conservativity. For test case 2, see Figure 4.12 the numerical solution is also positive and conservative, but the results of the RK2 shows an overall higher accuracy. The drawbacks of the MP2 are on the one hand the comparatively high computational effort due to the necessity of solving a linear equation system and on the other hand that conservativity does not hold in biochemical sense, because conservation in biochemical context refers to the conservation of atoms as well as of energy, cp. *Bruggeman et al.* [2007]. The authors addressed the problem of conservativity and developed the *extended modified Patankar Euler method (EMP1)* of first order

$$c_i^{n+1} = c_i^n + h\Big(P_i(c^n) - D_i(c^n)\Big)\prod_{j\in L^n}\frac{c_j^{n+1}}{c_j^n}, \tag{4.26}$$

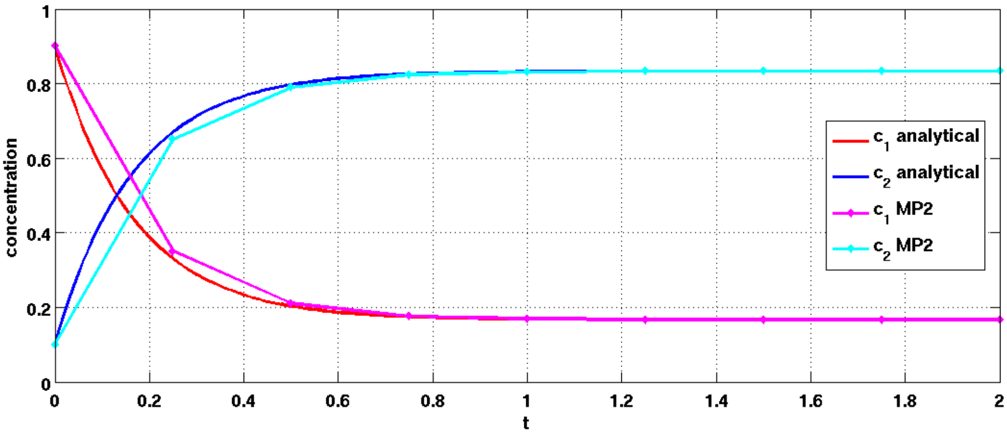


Figure 4.10: The MP2 applied to test case 1 in pink (c_1^n) and cyan (c_2^n) lines, together with the analytical solution of $c_1(t)$ (red) and $c_2(t)$ (blue). The step size h is chosen as 0.25.

where $L^n = \{i : P_i(c^n) - D_i(c^n) < 0,\ i \in \{1, ..., N\}\}$ and the
extended modified Patankar Runge Kutta method (EMP2) of second order

$$\begin{aligned} c_i^{(1)} &= c_i^n + h\Big(P_i(c^n) - D_i(c^n)\Big) \prod_{j\in L^n} \frac{c_j^{(1)}}{c_j^n}, \\ c_i^{n+1} &= c_i^n + \frac{h}{2}\Big(P_i(c^n) + P_i(c^{(1)})\Big) \prod_{k\in K^n} \frac{c_k^{n+1}(t)}{c_k^{(1)}(t)} \\ &\quad - \Big(D_i(c^n) - D_i(c^{(1)})\Big) \prod_{k\in K^n} \frac{c_k^{n+1}(t)}{c_k^{(1)}(t)}, \end{aligned} \tag{4.27}$$

where $L^n = \{i : P_i(c^n) - D_i(c^n) < 0,\ i \in \{1, ..., N\}\}$ and
$K^n = \{i : P_i(c^n) + P_i(c^{(1)}) - D_i(c^n) - D_i(c^{(1)}) < 0,\ i \in \{1, ..., N\}\}$.
Though the EMP1 and the EMP2 seem to be schemes in which n non-linear implicit equations have to be solved, they can be reduced to a polynomial equation in one single variable as it is shown in *Bruggeman et al.* [2007]. Thus the problem to be solved is just a polynomial one. The EMP2 is a conservative and unconditionally positive numerical scheme and hence no negative concentrations appear for any time step. The results are more accurate for test case 1 and 2, see Figure 4.11 and 4.13, compared to those given by the MP2. Furthermore, the EMP2 is not suitable for solving stiff problems, because a very large negative relative derivative occurs.

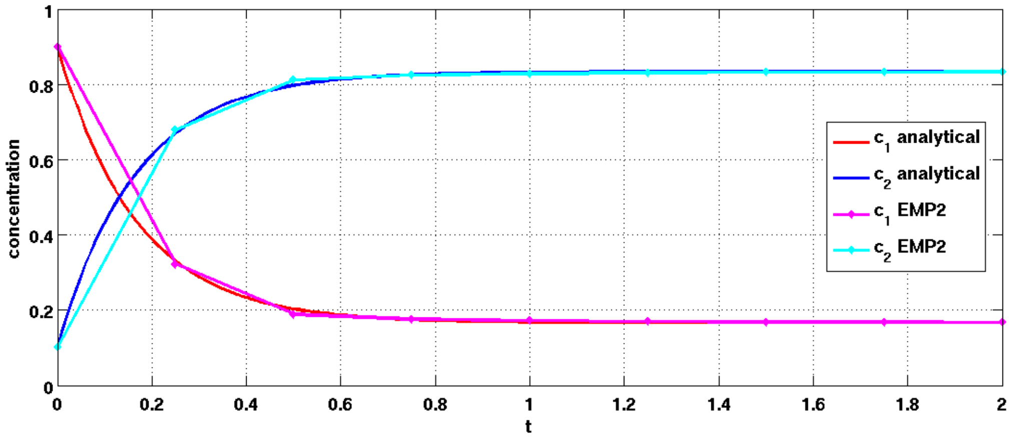


Figure 4.11: The EMP2 applied to test case 1 together with the analytical solution. The pink and cyan lines show the approximated solutions of $c_1(t)$ and $c_2(t)$ and the red and blue lines show the corresponding analytical solutions. The step size h equals 0.25.

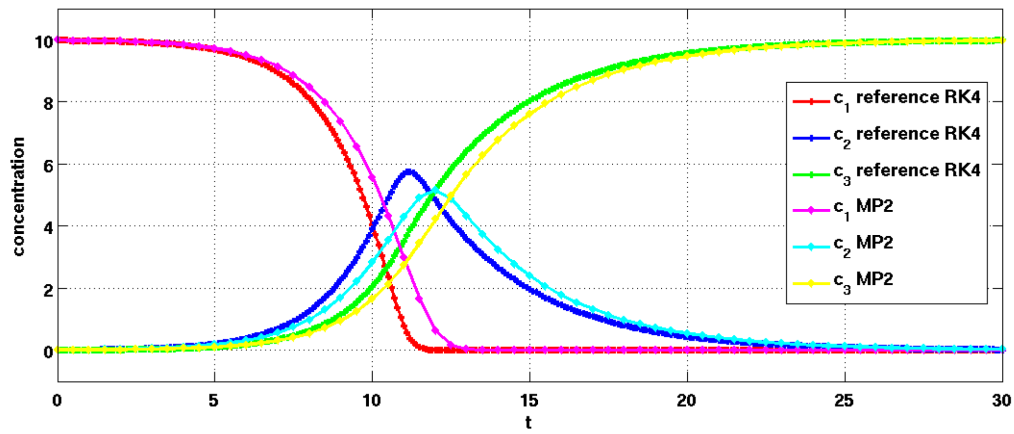


Figure 4.12: The MP2 applied to test case 2 in pink (c_1^n), cyan (c_2^n) and yellow (c_3^n) together with the reference solution (RK4) for $c_1(t)n$ (red), $c_2(t)$ (blue) and $c_3(t)$ (green). The step size h is chosen as 0.5.

4.3.3 Semi-implicit schemes with adapted time steps

Given that the RBMs are derived from diagonal implicit RKMs, see chapter 3, they preserve exact conservation properties and furthermore, they are suitable for solving stiff problems. Additionally, the positivity can be achieved by choosing the tolerance values *atol* and *rtol* sufficient small, but the smaller the tolerances, the more time steps are needed and hence the computing time increases. The maximum values for *atol* and *rtol* have been prescribed at 10^{-2} and through trial and error the best results have been achieved by the

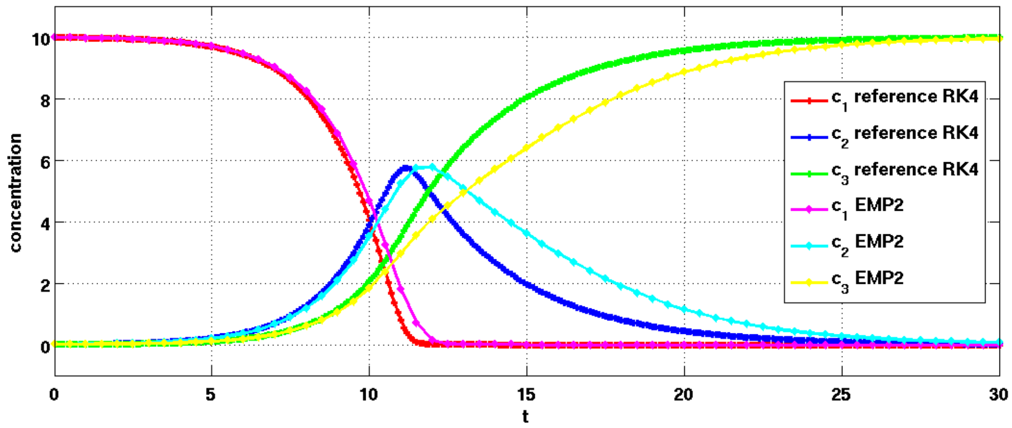


Figure 4.13: The EMP2 applied to test case 2 together with the reference solution (RK4). The pink, cyan and yellow lines show the approximated solutions of $c_1(t)$, $c_2(t)$ and $c_3(t)$. The red, blue and green lines show the corresponding reference solutions (RK4). The step size h is chosen as 0.5.

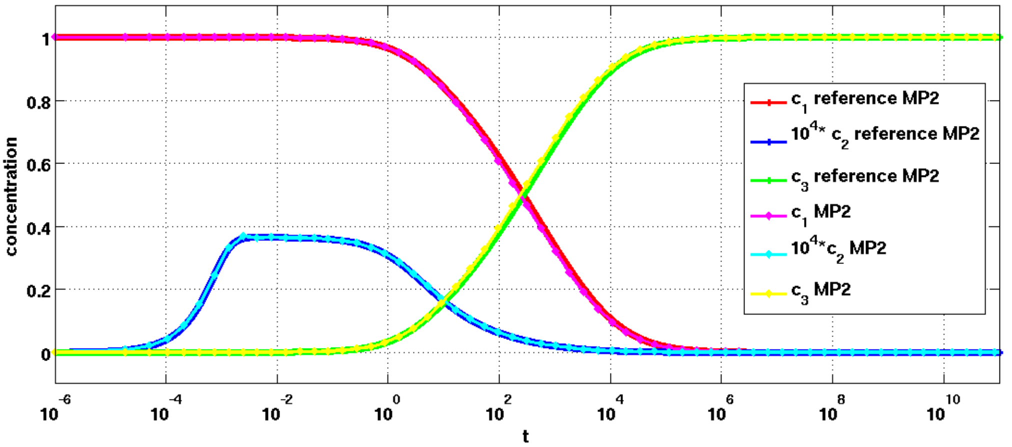


Figure 4.14: The MP2 applied to test case 3 in pink (c_1^n), cyan (c_2^n) and yellow (c_3^n) lines together with the reference solution (MP2) of $c_1(t)$ (red), $c_2(t)$ (blue) and $c_3(t)$ (green). As in *Burchard et al.* [2003] an exponential growing step size $h^n = 10^{-6} \cdot 1.8^n$ is used to compute the approximated solution, resulting in 63 time steps. The reference solution is also computed with exponentially increasing time steps $h^n = 2 \cdot 10^{-14} \cdot 1.0002^n$. For clarity, the results of c_2^n have been multiplied by a factor of 10^4.

denoted values, see Table 4.3.3. In this Table also the starting step size h and the number of accepted and rejected steps are listed.
The two Rosenbrock solvers ROS3 and ROS4 that are used for comparison, are presented in an rearranged form of equation (3.16) and for this derivation the starting point is the general form of the 4-stage RBM, defined on page 18

$$\begin{aligned}
k_1 &= \frac{h}{\mathbf{I} - h\gamma J_i} f(c^n) \\
k_2 &= \frac{h}{\mathbf{I} - h\gamma J_i} (f(c^n + \alpha_{21}k_1) + J_i\gamma_{21}k_1) \\
k_3 &= \frac{h}{\mathbf{I} - h\gamma J_i} (f(c^n + \alpha_{31}k_1 + \alpha_{32}k_2 + J_i(\gamma_{31}k_1 + \gamma_{32}k_2) \\
k_4 &= \frac{h}{\mathbf{I} - h\gamma J_i} (f(c^n + \alpha_{41}k_1 + \alpha_{42}k_2 + \alpha_{43}k_3) + \\
&\quad + \frac{h}{\mathbf{I} - h\gamma J_i} (J_i(\gamma_{41}k_1 + \gamma_{42}k_2 + \gamma_{43}k_3)) \\
c^{n+1} &= c^n + b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4
\end{aligned} \tag{4.28}$$

where J_i denotes the i-th component of the Jacobian matrix

$$J_i := \sum_{j=1}^{N} \frac{\partial f_i(c^n)}{\partial c_j^n} \tag{4.29}$$

and γ_{ij} and $\alpha_{i,j}$ are the determining coefficients identical to that of the defined RBM on page 18.
Inserting $k_l,\ l = 1, \ldots, 4$ into the equation for c^{n+1}, given in system 4.3.3, and using the following substitution

$$\begin{aligned}
c_i^{(1)} &:= c_i^n + \tilde{h}\cdot \alpha_{21} f_i(c^n) \\
c_i^{(2)} &:= c_i^n + \tilde{h}\cdot \left[\alpha_{31} f_i(c^n) + \alpha_{32}(f(c^{(1)}) + J_i\gamma_{21} f_i(c^n))\right] \\
c_i^{(3)} &:= c_i^n + \tilde{h}\cdot \left[\alpha_{41} f_i(c^n) + \alpha_{42}(f_i(c^{(1)}) + J_i\gamma_{21} f(c^n))\right] + \\
&\quad + \tilde{h}\cdot \left[\alpha_{43}(f_i(c^{(2)}) + J_i\tilde{h}(\gamma_{31} f_i(c^n) + \gamma_{32}(f_i(c^{(1)}) + J_i\gamma_{21} f(c^n))))\right]
\end{aligned} \tag{4.30}$$

where

$$\tilde{h} := \frac{h}{\mathbf{I} - h\gamma J_i} \tag{4.31}$$

results in

$$\begin{aligned}
c^{n+1} = c^n &+ \tilde{h}\cdot \left[b_1 f_i(c^n) + b_2 \left\{ f_i(c^{(1)}) + J_i\gamma_{21} f_i(c^n) \right\} + b_3 f_i(c^{(2)})\right] + \\
&+ \tilde{h}^2 b_3 J_i \left\{ \gamma_{31} f_i(c^n) + \gamma_{32}(f_i(c^{(1)}) + J_i\gamma_{21} f_i(c^n)) \right\} + \tilde{h} b_4 f_i(c^{(3)}) + \\
&+ \tilde{h}^2 b_4 J_i \left\{ \gamma_{41} f_i(c^n) + \gamma_{42}(f_i(c^{(1)}) + J_i\gamma_{21} f_i(c^n)) \right\} + \\
&+ \tilde{h}^2 b_4 J_i\gamma_{43} f(c^{(3)}) + \\
&+ \tilde{h}^2 b_4 J_i\gamma_{43} \left\{ J_i(\gamma_{31} f_i(c^n) + \gamma_{32}(f_i(c^{(1)}) + J_i\gamma_{21} f_i(c^n)) \right\}.
\end{aligned} \tag{4.32}$$

For further simplification the following two abbreviations are used

$$\begin{aligned} p_1(t) &:= f_i(c^{(1)}) + J_i\gamma_{21} f_i(c^n) \\ p_2(t) &:= f_i(c^{(3)}) + J_i(\gamma_{31} f_i(c^n) + \gamma_{32}(f_i(c^{(1)}) + J_i\gamma_{21} f_i(c^n)) \end{aligned} \tag{4.33}$$

which convert equation (4.32) into

$$\begin{aligned} c^{n+1} = c^n &+ \tilde{h}\cdot [b_1 f_i(c^n) + b_2 p_1] + \\ &+ \tilde{h}\cdot b_3 \left\{ f(c^{(2)}) + J_i\tilde{h}(\gamma_{31} f_i(c^n) + \gamma_{32} p_1) \right\} + \\ &+ \tilde{h}\cdot b_4 \left\{ f(c^{(3)}) + J_i\tilde{h}\left[\gamma_{41} f(c^n) + \gamma_{42} p_1 + \gamma_{43} p_2\right] \right\}. \end{aligned} \tag{4.34}$$

Additionally, using the notations

$$\begin{aligned} \gamma_1 &= \tilde{h}(\gamma_{31} f_i(c^n) + \gamma_{32} p_1) \\ \gamma_2 &= \tilde{h}(\gamma_{41} f_i(c^n) + \gamma_{42} p_1 + \gamma_{43} p_2) \end{aligned} \tag{4.35}$$

and finally applying the equations (4.33) and (4.35) to equation (4.30) and substitute

$$f_i(c^n) = P_i(c^n) - D_i(c^n), \tag{4.36}$$

the rearranged Rosenbrock formula for the ROS4 solver of fourth order is obtained:

$$\begin{aligned} c_i^{(1)} &= c_i^n + \tilde{h}\cdot \alpha_{21}(P_i(c^n) - D_i(c^n)) \\ c_i^{(2)} &= c_i^n + \tilde{h}\cdot [\alpha_{31}(P_i(c^n) - D_i(c^n)) + \alpha_{32} p_1] \\ c_i^{(3)} &= c_i^n + \tilde{h}\cdot [\alpha_{41}(P_i(c^n) - D_i(c^n)) + \alpha_{42} p_1 + \alpha_{43} p_2] \\ c_i^{n+1} &= c_i^n + \tilde{h} b_1\cdot [P_i(c^n) - D_i(c^n)] + \tilde{h} b_2 p_1 + \\ &\quad + \tilde{h} b_3 \left[P_i(c^{(2)}) - D_i(c^{(2)}) + J_i\gamma_1 \right] + \\ &\quad + \tilde{h} b_4 \left[P_i(c^{(3)}) - D_i(c^{(3)}) + J_i\gamma_2 \right]. \end{aligned} \tag{4.37}$$

In addition, the formula for the third order Rosenbrock solver ROS3 is obtained, if the equation for $c_i^{(3)}$ is let out and b_4 is set to zero.

Remark 4.3.1.
In this study

1. the *L-stable* versions of both solvers are used, see *Table 7.2* in *Hairer and Wanner* [1991], where the values of the coefficients α_{lj}, b_l and γ_{lj} are taken from *Hairer and Wanner* [1991] and *Sandu et al.* [1997b].

2. a modified version of ROS3 is used, according to the origin one, which has been developed by *Sandu et al.* [1997b].

		ROS3			ROS4	
test cases	1	2	3	1	2	3
rtol	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-5}	10^{-2}
atol	10^{-3}	10^{-4}	10^{-7}	10^{-3}	10^{-5}	10^{-7}
h	0.25	0.5	10^{-3}	0.25	0.5	10^{-3}
accepted steps	9 (9)	50 (60)	38 (63)	9 (9)	57 (60)	43 (63)
rejected steps	1	10	2	0	7	0

Table 4.1: The tolerance values *atol* and *rtol*, starting step size h and the number of accepted and rejected steps for ROS3 and ROS4, respectively. In brackets: the number of steps needed by the fixed step methods.

Applying ROS3 and ROS4 without any step size restriction to test case 1, see 4.15 and 4.16, results in 9 time steps with positive values. Both schemes show highly accurate results even for the initial phase of the problem. For test case 2, see the Figures 4.17 and 4.18, both methods give unconditional positive and conservative results in all phases of the model problem, too. Nevertheless, ROS4 shows more accurate results than ROS3, due to the fact that it makes seven more steps and hence it can higher resolve the approximated solution. Also the approximated solutions of test case 3, depicted in the Figures 4.19 and 4.20, are positive, conservative and accurate for both Rosenbrock solvers. However, both RBMs have some problems in accurately approximating the initial phase (up to $t = 10^4$) of $c_2(t)$, though they rapidly adapt the step size.

From a mathematical point of view numerical methods with a fixed step size and with adaptive step size cannot be compared, because the underlying criteria - computing time and global method error - are not comparable. Similar accurate results can be obtained for fixed step methods compared to schemes with adaptive step size by choosing the step size small enough. However, this leads to an increase of computing time, due to the fact that they have to solve a higher number of equations, than the adaptive size methods. Thus, in order to directly compare all numerical schemes, the step size of the RBMs is set to 0.25 for the 1st and 0.5 for the 2nd test case (similar to the schemes with the fixed step size), by restricting the maximum step size. Furthermore the tolerances values *rtol* and *atol* have been chosen to be large (10^{-1}). Thus, each step is accepted and the next time step size is set to the maximum. This restriction is only possible for the first two test cases, because in test case 3 an exponential growing step size is used, which has not been transfered to the RBMs in this study. Please note, that an alternative could be to include adaptive step size mechanisms in the fixed step methods (using e.g. the embedded RKM). As expected, ROS3 and ROS4 are less accurate in representing the initial phase, see Figure 4.21 and 4.22, compared to the model run with the RBMs using an adaptive step size for integration, see Figure 4.15 and 4.16. After $t = 0.5$, where the reaction between the

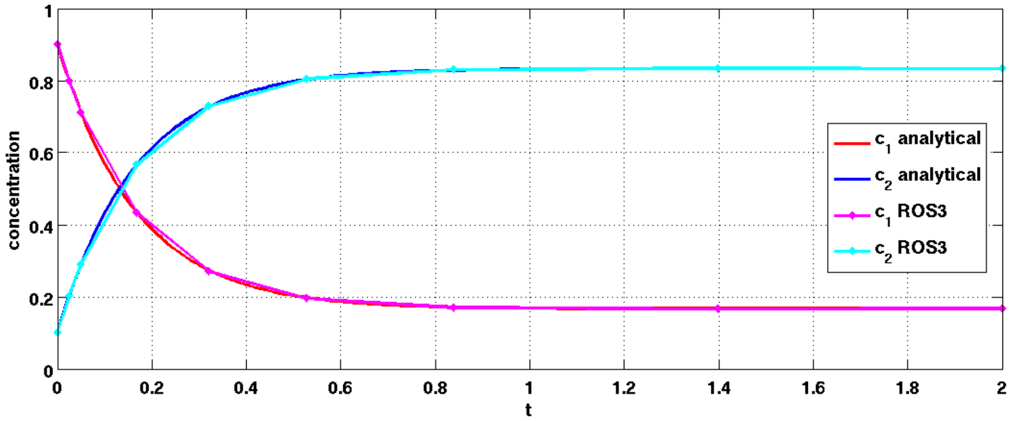


Figure 4.15: ROS3 applied to test case 1 together with the analytical solution that is shown in the red ($c_1(t)$) and blue ($c_2(t)$) lines. The approximated solutions are plotted in pink (c_1^n)and cyan (c_2^n) lines. The starting step size is chosen as 0.25, the relative error tolerance *rtol* as 10^{-4} and the absolute error tolerance *atol* as 10^{-3}.

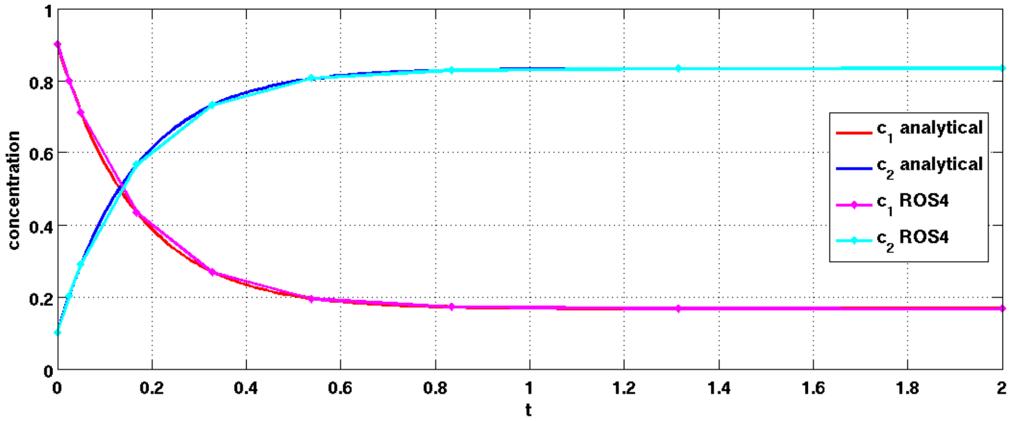


Figure 4.16: ROS4 applied to test case 1 together with the analytical solution that is shown in the red ($c_1(t)$) and blue ($c_2(t)$) lines. The approximated solutions are plotted in pink (c_1^n)and cyan (c_2^n) lines. The starting step size is chosen as 0.25, the relative error tolerance *rtol* as 10^{-4} and the absolute error tolerance *atol* as 10^{-3}.

temporal changes of the two constituents cease, the approximations are highly accurate. For test case 2, similar accurate results are obtained for the whole integration interval, as shown in the Figures 4.23 and 4.24 with RBMs and fixed time step compared to RBMs with adapted time step. Overall, the differences between ROS3 and ROS4 are marginal and their approximations are positive and conservative for both test cases.

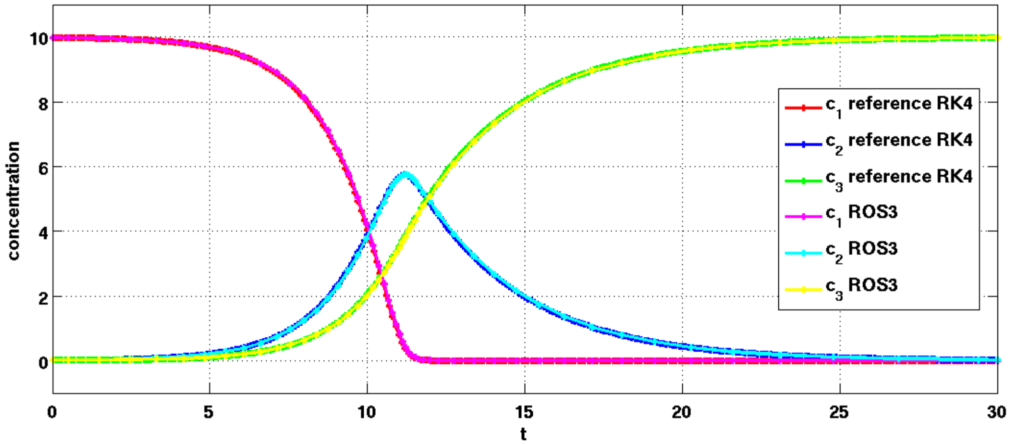


Figure 4.17: ROS3 applied to test case 2 together with the reference solution (RK4) that is shown in the red (c_1^n), blue (c_2^n) and green (c_3^n) lines. The pink, cyan and yellow lines show the approximated solutions of $c_1(t)$, $c_2(t)$ and $c_3(t)$. The starting step size is chosen as 0.5, the relative error tolerance *rtol* as 10^{-6} and the absolute error tolerance *atol* as 10^{-4}.

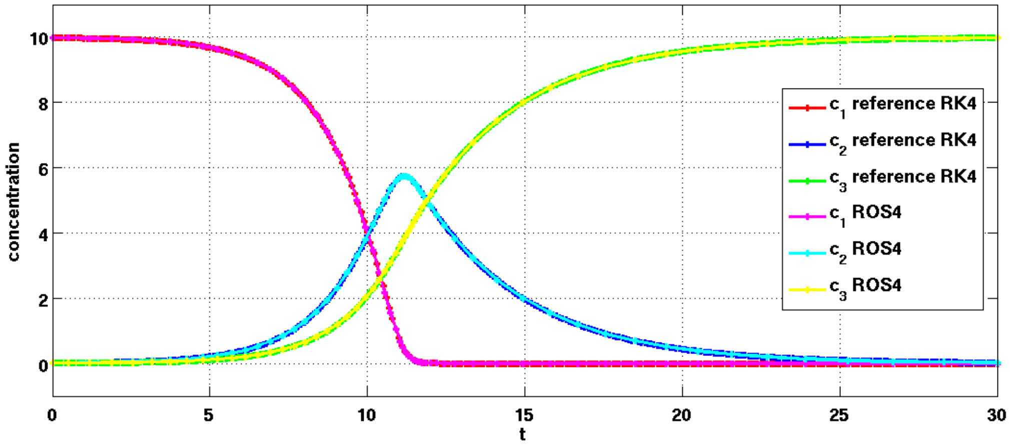


Figure 4.18: ROS4 applied to test case 2 together with the reference solution (RK4) that is shown in the red (c_1^n), blue (c_2^n) and green (c_3^n) lines. The pink, cyan and yellow lines show the approximated solutions of $c_1(t)$, $c_2(t)$ and $c_3(t)$. The starting step size is chosen as 0.5, the relative error tolerance *rtol* as 10^{-5} and the absolute error tolerance *atol* as 10^{-5}.

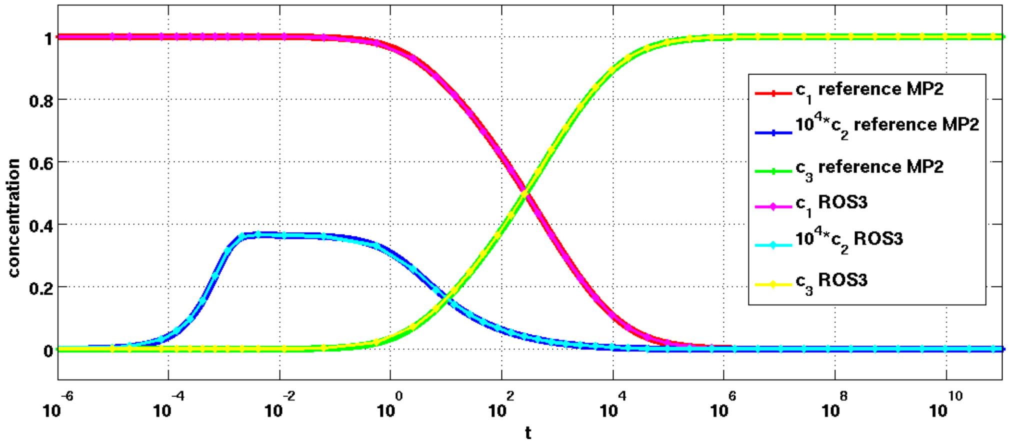


Figure 4.19: ROS3 applied to test case 3 together with the reference solution (MP2), computed with exponentially increasing time steps $h^n = 2\cdot 10^{-14}\cdot 1.0002^n$, shown in the red (c_1^n), blue (c_2^n) and green (c_3^n) lines. The pink, cyan and yellow lines show the approximated solutions of $c_1(t)$, $c_2(t)$ and $c_3(t)$. The starting step size h is chosen as 10^{-3}, the relative error tolerance $rtol$ as 10^{-2} and the absolute error tolerance $atol$ as 10^{-7}. For clarity the results of $c_2 n$ again have been multiplied by a factor of 10^4.

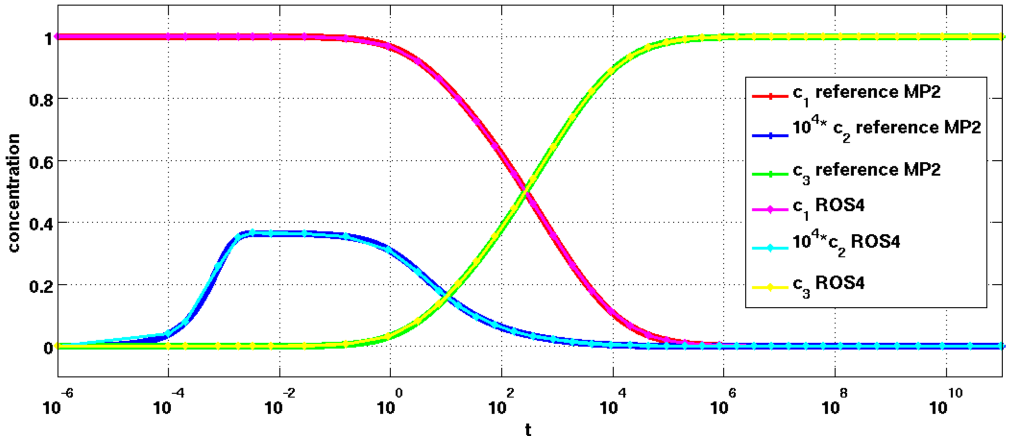


Figure 4.20: ROS4 applied to test case 3 together with the reference solution (MP2), computed with exponentially increasing time steps $h^n = 2\cdot 10^{-14}\cdot 1.0002^n$, shown in the red (c_1^n), blue (c_2^n) and green (c_3^n) lines. The pink, cyan and yellow lines show the approximated solutions of $c_1(t)$, $c_2(t)$ and $c_3(t)$. The starting step size h is chosen as 10^{-3}, the relative error tolerance $rtol$ as 10^{-2} and the absolute error tolerance $atol$ as 10^{-7}. For clarity the results of c_2^n again have been multiplied by a factor of 10^4.

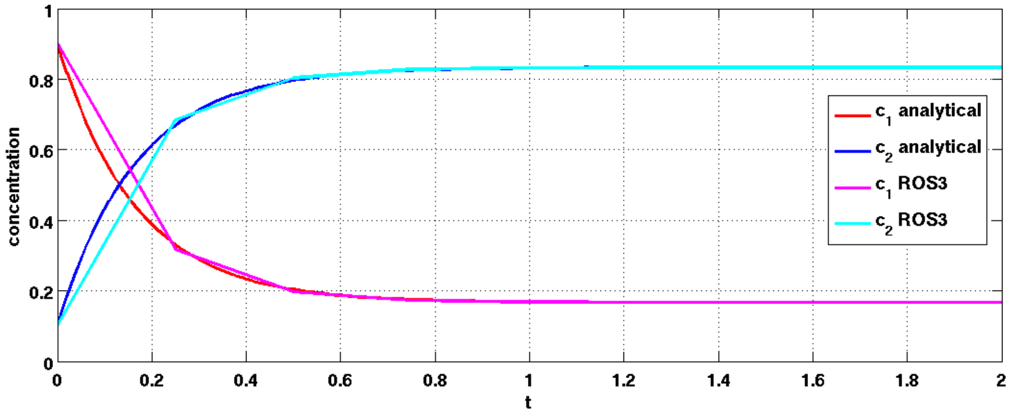


Figure 4.21: Step size restricted ROS3 applied to test case 1 together with the analytical solution that is shown in the red ($c_1(t)$) and blue ($c_2(t)$) lines. The approximated solutions are plotted in pink (c_1^n)and cyan (c_2^n) lines. The step size $h = 0.25$ is fixed for each step, the relative and absolute error tolerances *rtol* and *atol* are set to 10^{-1}.

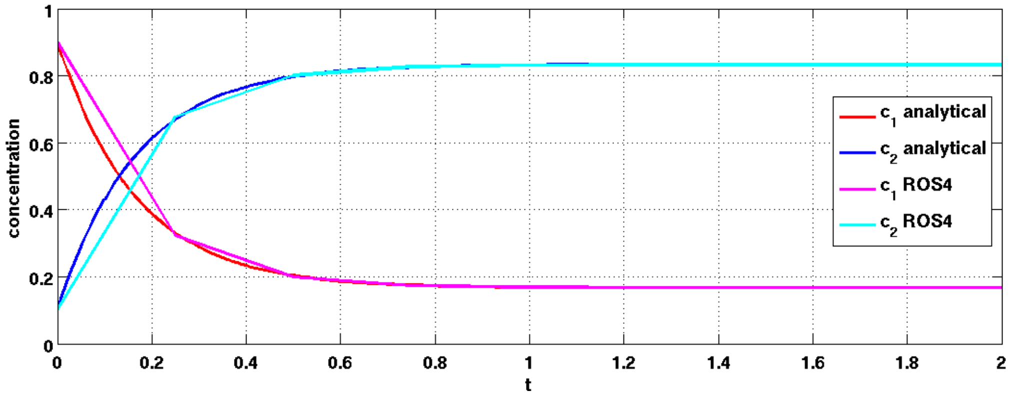


Figure 4.22: Step size restricted ROS4 applied to test case 1 together with the analytical solution that is shown in the red ($c_1(t)$) and blue ($c_2(t)$) lines. The approximated solutions are plotted in pink (c_1^n)and cyan (c_2^n) lines. The step size $h = 0.25$ is fixed for each step, the relative and absolute error tolerances *rtol* and *atol* are set to 10^{-1}.

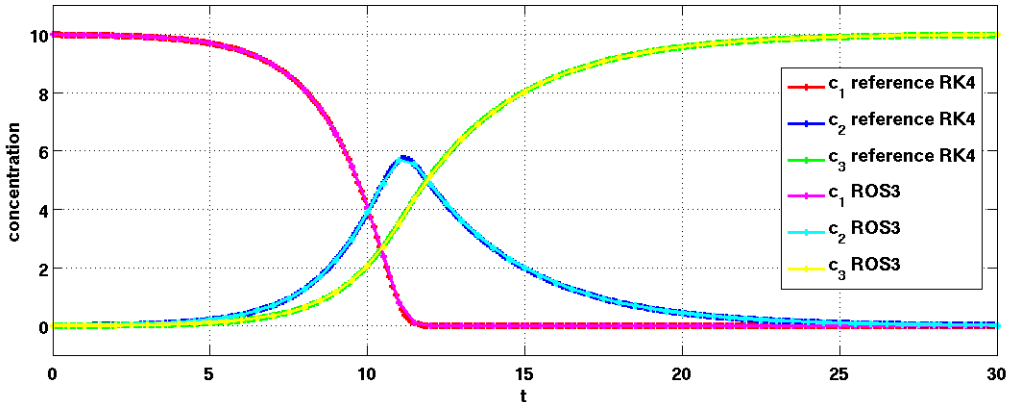


Figure 4.23: Step size restricted ROS3 applied to test case 2 together with the reference solution (RK4) that is shown in the red (c_1^n), blue (c_2^n) and green (c_3^n) lines. The pink, cyan and yellow lines show the approximated solutions of $c_1(t)$, $c_2(t)$ and $c_3(t)$. The step size $h = 0.5$ is fixed for each step, the relative and absolute error tolerances $rtol$ and $atol$ are set to 10^{-1}.

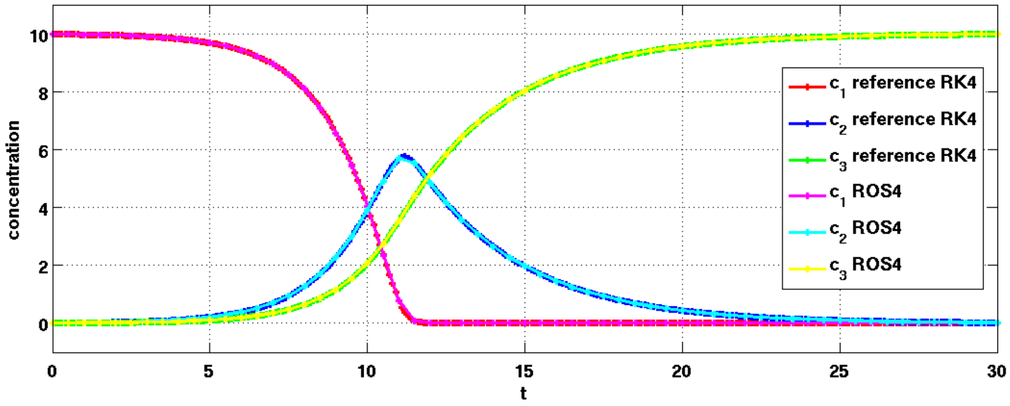


Figure 4.24: ROS4 applied to test case 2 together with the reference solution (RK4) that is shown in the red (c_1^n), blue (c_2^n) and green (c_3^n) lines. The pink, cyan and yellow lines show the approximated solutions of $c_1(t)$, $c_2(t)$ and $c_3(t)$. The step size $h = 0.5$ is fixed for each step, the relative and absolute error tolerances $rtol$ and $atol$ are set to 10^{-1}.

5 Comparison of the schemes

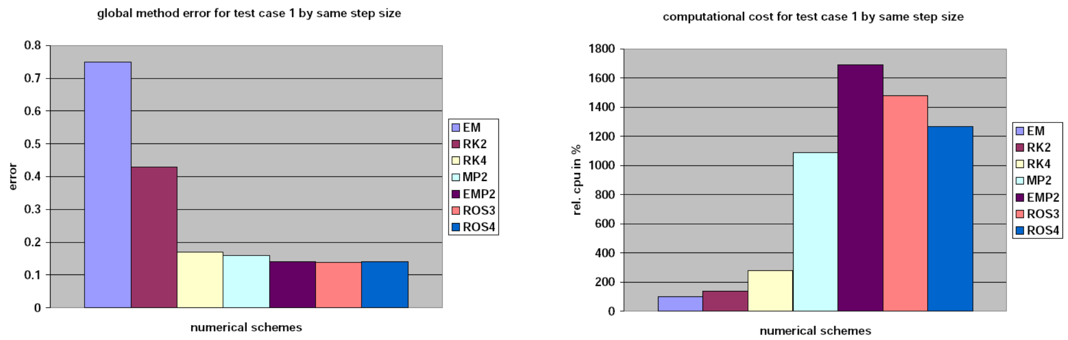
In this chapter the RBMs are compared with traditional (Euler and Runge Kutta) and advanced numerical schemes (modified and extended modified Patankar methods) used in ecosystem modelling.
In order to evaluate the schemes and to define whether a numerical method is suitable for applications to biogeochemical models

1. the global method error
$$\varepsilon = \max_{l=0,\dots,n} \|c^l(t) - c(t_l)\|_2 \tag{5.1}$$
and
2. the computational cost

are determined for each scheme and compared.
Each numerical method was computed four times for 100000 times and the average of these four results was taken for evaluation to get significant results of the computing time. For obtaining the global method error of the schemes for test case 1, the results are compared to the analytical solution. For test case 2 and 3 there is no analytical solution and hence reference solutions (RK4 for test case 2 and MP2 for test case 3 with very small step size) have been taken for computing the error. This requires, that the computed data of all applied methods have to be interpolated to 300 steps of size $h = 0.1$ for test case 2 and to 241795 steps of size $h = h^n = 2\cdot 10^{-14}\cdot 1.002^n$ for test case 3, respectively. Afterwards, the global method error and the computational cost of the schemes are determined for each method and the results are plotted. As mentioned above, the explicit and quasi-implicit schemes use fixed time steps. Thus, to compare the schemes with the RBMs, the step size of the latter has been restricted, as mentioned in chapter 4. Due to the fact that the explicit schemes as well as the EMP2 are not suitable for solving stiff ordinary differential equations, the results of the MP2 and the RBMs are compared.
In the first section all explicit, quasi-implicit and semi-implicit methods are compared using a fixed time step. Additionally, experiments are performed with RBMs using adaptive step size. The last section gives an overview of the influence of the tolerance values on the performance of the RBMs.

5.1 Comparison between explicit, quasi- and semi-implicit methods with fixed step size

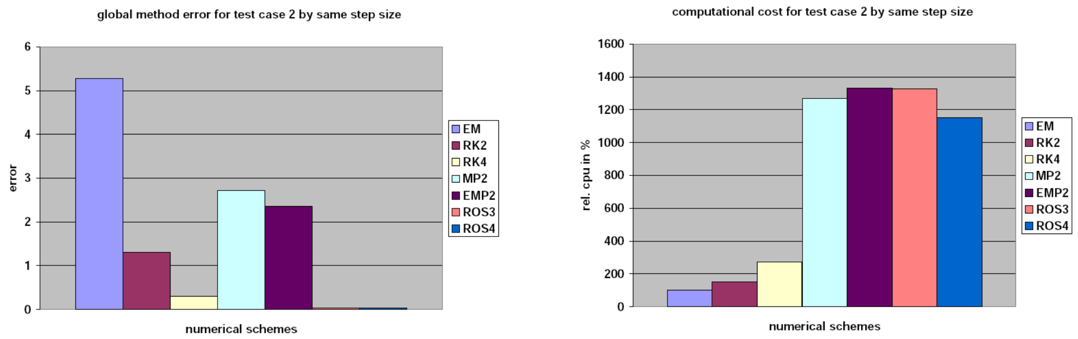
First, the results of all numerical schemes with fixed step size are compared. Note however, an essential feature of the RBMs is eliminated.
The results of test case 1 show that the better the performance of the methods, in terms of



(a) Global method error

(b) Execution time

Figure 5.1: Global method error (left) and execution time (right) of all seven numerical schemes with fixed steps of size 0.25 applied to test case 1 in the following order: EM, RK2, RK4, MP2, EMP2, ROS3, ROS4.

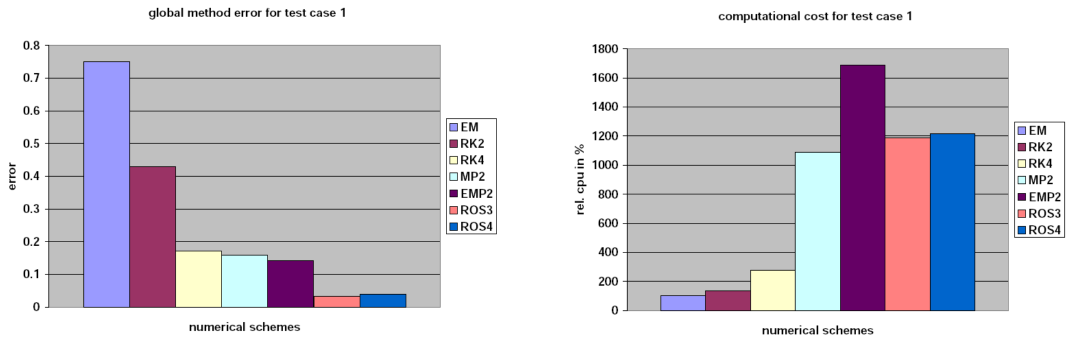


(a) Global method error

(b) Execution time

Figure 5.2: Global method error (left) and execution time (right) of all seven numerical schemes with fixed steps of size 0.5 applied to test case 2 in the following order: EM, RK2, RK4, MP2, EMP2, ROS3, ROS4.

accuracy (see Figure 5.1(a)) the more cost expensive are the schemes (see Figure 5.1(b)). However, the trend is not as such visible for the RBMs. Both solvers - ROS3 and ROS4 - give accurate results similar to the RK4, MP2 and EMP2. However, their computing time is higher as that of the MP2 but less than that of the EMP2. In contrast, the RKMs have the lowest computational effort (six times less than the other tested schemes).
For test case 2 the general trend in the relationship of the global method error and the computing time cannot be found. The RBMs give the most accurate results of all tested schemes, see Figure 5.2(a), because the fixed step size forces the solvers to compute solutions even at times where the temporal changes of the constituents are relatively small. Thus, their global method error is about ten times smaller than that of the RK4 and even 100 times smaller than that of the MP2 and EMP2. The RBMs need similar computing time as the latter two. That means, the RBMs are about six times more expensive than the RK4, see Figure 5.2(b)



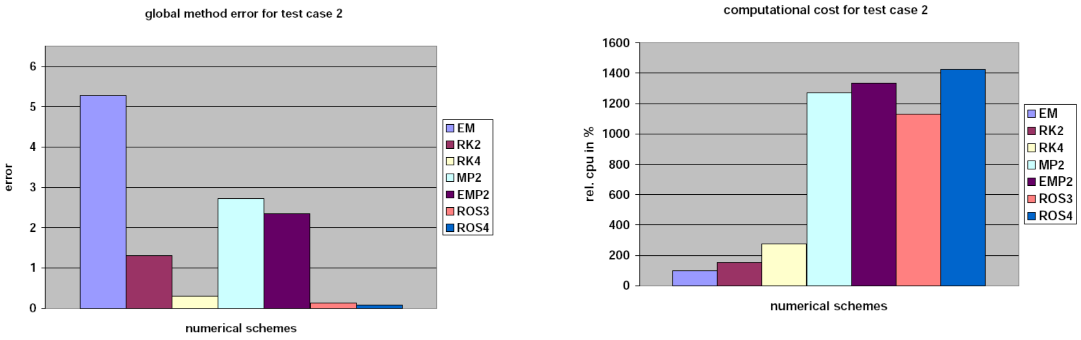
(a) Global method error

(b) Execution time

Figure 5.3: Global method error (left) and execution time (right) of all seven numerical schemes applied to test case 1 in the following order: EM, RK2, RK4, MP2, EMP2, ROS3, ROS4.

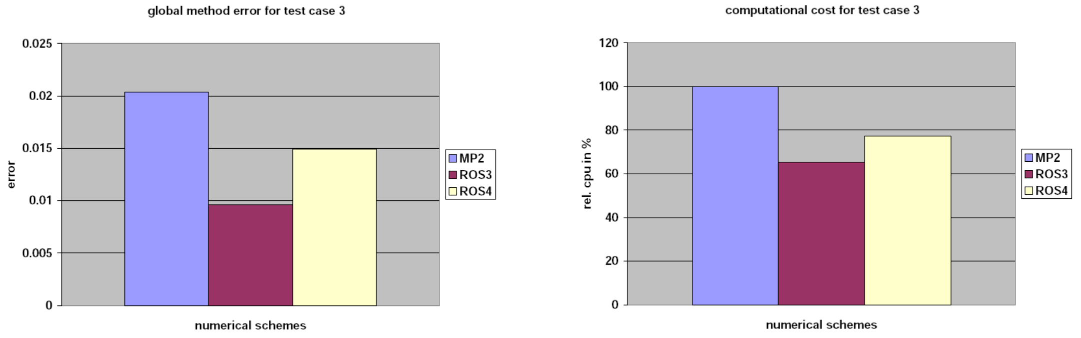
5.2 Comparative analysis between the fixed step methods and the Rosenbrock methods with adapted step size

While for the former experiments a fixed time step for the RBMs is used for a direct comparison with the explicit and quasi-implicit schemes, in these experiments the RBMs with adaptive time stepping (in their original form) are applied. As expected, the performance of the RBMs is influenced when this feature is included. The results for the first test case are more accurate and less expensive (Figure 5.3) and again, the better the performance in terms of accuracy (Figure 5.3(a)), the higher is the computing time, except the EMP2 (Figure 5.3(b)). The RBMs give the most accurate results of all schemes. The global error of the other tested schemes is about one order of magnitude higher than that of the RBMs. The EMP2 requires the highest computational effort, while the RBMs have similar execution times as the MP2. The most effective schemes are the explicit RKM (five times less computing time than RBMs).
For test case 2 again the RBMs are the most accurate schemes of all tested numerical methods, see Figure 5.4(a). In contrast to the other explicit and quasi-implicit schemes, the global error of the RK4 has the same order of magnitude as the RBMs. The quasi-implicit schemes are about ten times less accurate than the RBM, but they are as expensive as both Rosenbrock solvers and hence about four times more expensive than the RKMs, as can be seen in Figure 5.4.
Similar to test case 2, there is no correlation between the global method error and the execution time of MP2 and the Rosenbrock solvers in test case 3. The ranges of global method error and the computational cost are smaller for test case 3, than for test case 1 and 2. Compared to the MP2, the RBMs give better results, with respect to error and computational effort. ROS3 is two times more accurate and requires a computing time that is 40% less than that of the MP2.



(a) Global method error (b) Execution time

Figure 5.4: Global method error (left) and execution time (right) of all seven numerical schemes applied to test case 2 in the following order: EM, RK2, RK4, MP2, EMP2, ROS3, ROS4.



(a) Global method error (b) Execution time

Figure 5.5: Global method error (left) and execution time (right) for test case 3 of the three numerical schemes in the following order: MP2, ROS3, ROS4.

5.2.1 Comparison of the efficiency of the numerical schemes

In order to quantify the costs of the RBMs, all methods are implemented in such a way that the global error ε is in the range of

$$1.1\cdot 10^{-1} \leq \varepsilon \leq 1.4\cdot 10^{-1}, \tag{5.2}$$

i.e. their step size is chosen sufficiently small, see Table 5.2.1. In the following, their runtime was measured in the same way as described above on page 47. The Figures 5.6, 5.7 and 5.8 show the results of these simulations. Only the explicit methods, particularly the second and fourth order RKMs, need less time than the RBMs for the first two test cases. Both schemes require only half of the time to give results of the same order of magnitude of accuracy. The MP2 is slightly more expensive, than the RBMs, for the same order of magnitude of accuracy for test case 1, although it requires twice as many steps as the RBMs. Both quasi-implicit schemes - MP2 and EMP2 - need more time for computing test case 2 than the RBMs, while the EMP2 is the most expensive method.

test case	method	step size	number of steps
1	EM	0.025	80
	RK2	0.08	25
	RK4	0.1	20
	MP2	0.9	20
	EMP2	0.1	20
2	EM	0.01	3000
	RK2	0.2	150
	RK4	0.4	75
	MP2	0.08	375
	EMP2	0.06	500
3	MP2	$10^{-14} \cdot 1.2^j$	168

Table 5.1: required step size of the fixed step methods to give an global error in the range of equation 5.2.1; $j = 1, \ldots, 168$.

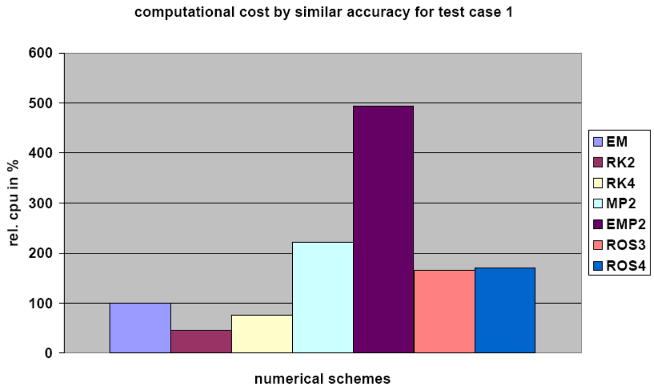


Figure 5.6: Execution time of the seven numerical schemes EM, RK2, RK4, MP2, EMP2, ROS3 and ROS4 applied to test case 1 by the similar dimension of the global method error.

A highly accurate result has already been obtained for the MP2 applied to test case 3 before in the experiment without error restriction. After introducing the limits the computational effort increases about four times.

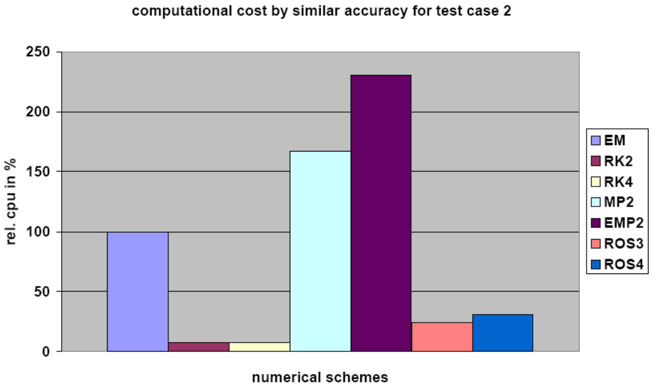


Figure 5.7: Execution time of the seven numerical schemes EM, RK2, RK4, MP2, EMP2, ROS3 and ROS4 applied to test case 2 by the similar dimension of the global method error.

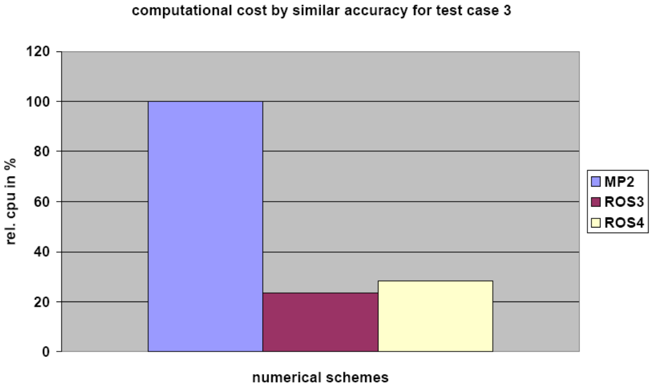


Figure 5.8: Execution time of the three numerical schemes MP2, ROS3 and ROS4 applied to test case 3 by the similar dimension of the global method error.

5.3 Effect of the tolerance values

The performance of the RBMs strongly depends on the choice of the tolerance values *rtol* and *atol*, because they determine the error value *err* that controls the step size of the scheme by regulating the size of the difference between the solution and its embedded solution, as described in chapter 3.4. For test case 1 and 2 the values for the absolute tolerance *atol* has been set to 10^{-2} and 10^{-4} while the relative tolerance *rtol* has been set to 10^{-2} and 10^{-4}, respectively, for each value of *atol*. Furthermore *atol* has been set to 10^{-6} while the relative tolerance *rtol* has been set to 10^{-2}, 10^{-4} and 10^{-6}, respectively. This yields seven pairs of tolerances *rtol* and *atol*. For test case 3 *rtol* equals 10^{-2}, while *atol* was set to 10^{-4}, 10^{-6} and 10^{-8}. It was not necessary to downscale the absolute and relative tolerances, because highly accurate results are achieved by using these values and hence three pairs of tolerances are obtained.

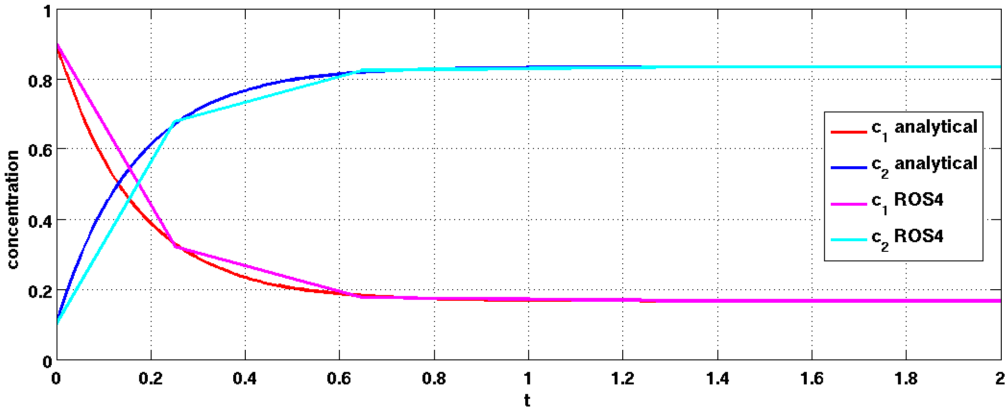


Figure 5.9: ROS4 applied to test case 1 with large tolerance values $rtol = atol = 10^{-2}$. The red ($c_1(t)$) and blue ($c_2(t)$) lines depict the analytical solution, where the pink and cyan lines show the approximated solution of $c_1(t)$ and $c_2(t)$.

In the Figures 5.9, 5.10, 5.12, 5.13, 5.15 and 5.16 the influence of the tolerance values on the performance only of the ROS4 in terms of accuracy is shown, because the results of ROS4 and ROS3 are similar. The Figures 5.9, 5.12 and 5.15 present the approximated solution of all test cases with the pair of large tolerance values, while the Figures 5.10, 5.13 and 5.16 present those with pairs of small tolerance values on the right. A higher accuracy is obtained by choosing *rtol* and *atol* sufficiently small (Figures 5.10, 5.13 and 5.16). The Figures 5.11, 5.14 and 5.17 present the tolerances against the global method error for ROS4 for all test cases.

The global method error decreases rapidly in all test cases (see Figures 5.11, 5.14 and 5.17), if the tolerance values have been chosen small enough. For test case 1 and 2 that means, even if the values of the pair ($atol$,$rtol$) are smaller than (10^{-2}, 10^{-4}) and (10^{-4}, 10^{-4}), the error is below 0.03 and 0.01, respectively and thus more accurate results are obtained. In order to achieve a global error below 0.015 for test case 3, the pair of tolerances must be smaller than (10^{-6}, 10^{-2}). Overall, the experiments with ROS3 and ROS4 show that with decreasing tolerance values the execution time is prolonged. However, this increase is rather marginal and still less time is needed compared to the MP2.

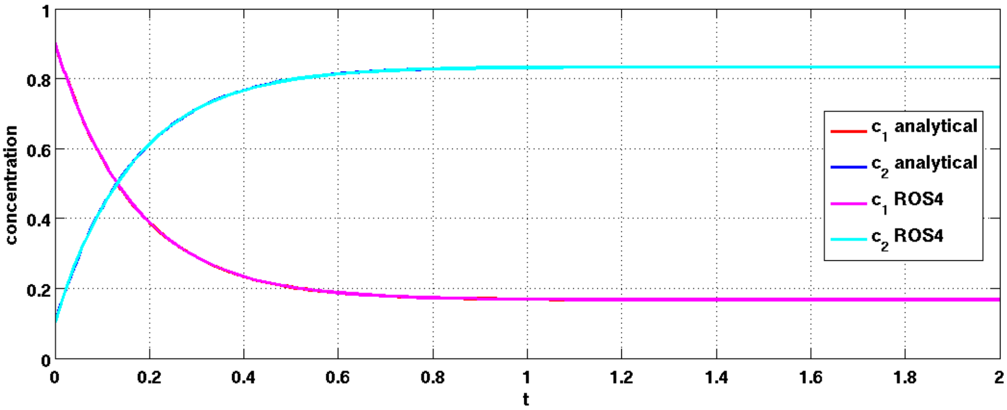


Figure 5.10: ROS4 applied to test case 1 with small tolerance values $rtol = atol = 10^{-6}$. The red ($c_1(t)$) and blue ($c_2(t)$) lines depict the analytical solution, where the pink and cyan lines show the approximated solution of $c_1(t)$ and $c_2(t)$.

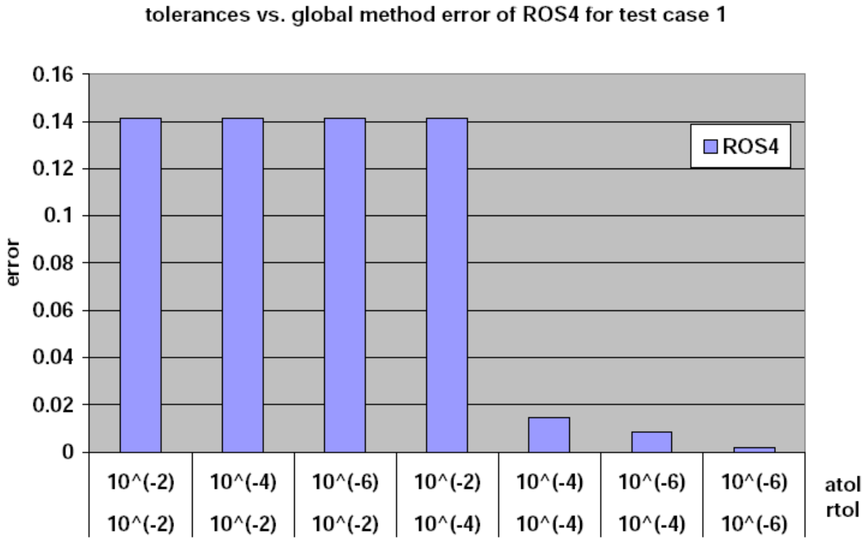


Figure 5.11: Effect of different values for $rtol$ and $atol$ on the global method error of ROS4 applied to test case 1. The x-axis shows the tolerances written below each other and the y-axis shows the computed error of the ROS4 solver for test case 1.

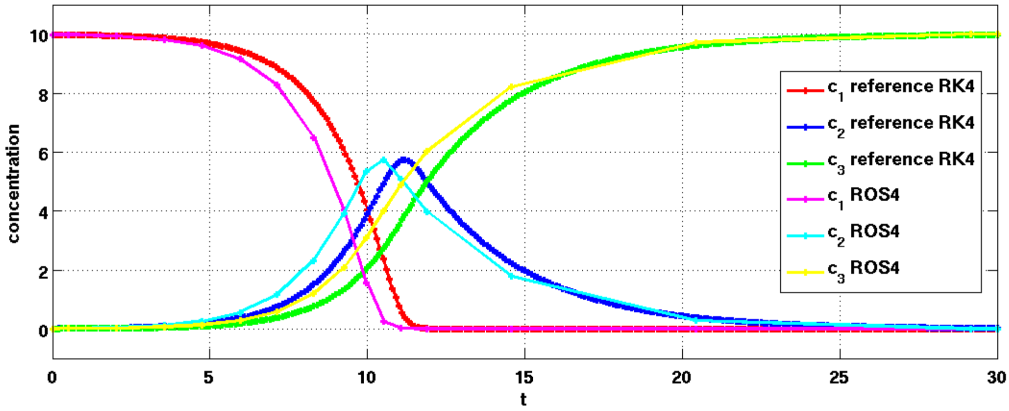


Figure 5.12: ROS4 applied to test case 2 with large tolerance values $rtol = atol = 10^{-2}$. The red ($c_1(t)$), blue ($c_2(t)$) and green ($c_3(t)$) lines depict the analytical solution, where the pink, cyan and yellow lines show the approximated solution of $c_1(t)$, $c_2(t)$ and $c_3(t)$.

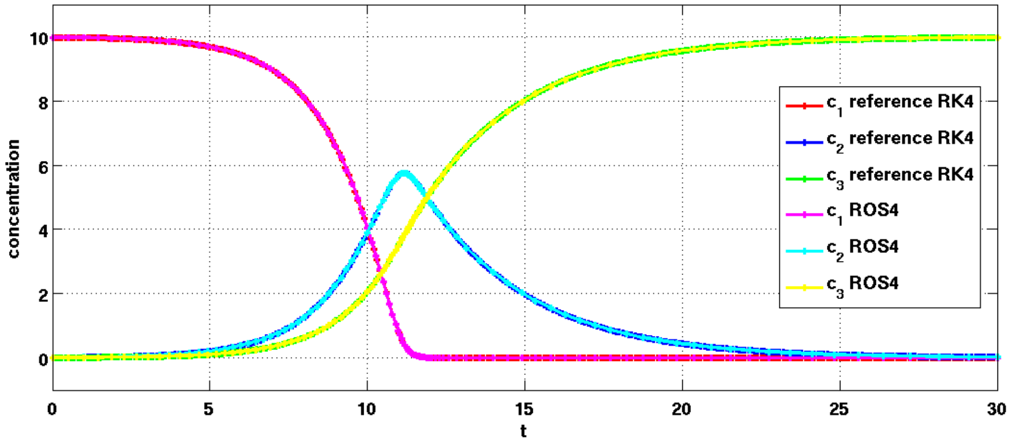


Figure 5.13: ROS4 applied to test case 2 with small tolerance values $rtol = atol = 10^{-6}$. The red ($c_1(t)$), blue ($c_2(t)$) and green ($c_3(t)$) lines depict the analytical solution, where the pink, cyan and yellow lines show the approximated solution of $c_1(t)$, $c_2(t)$ and $c_3(t)$.

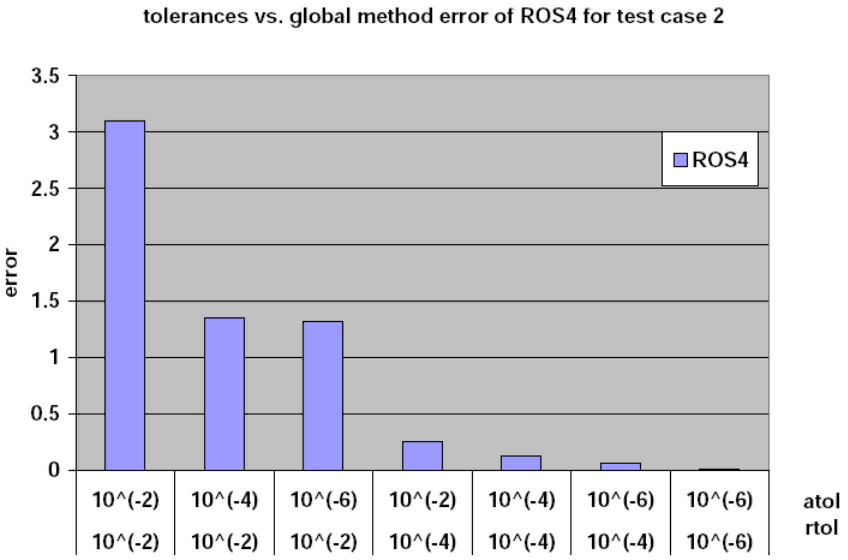


Figure 5.14: Effect of different values for *rtol* and *atol* on the global method error of ROS4 applied to test case 2. The x-axis shows the tolerances written below each other and the y-axis shows the computed error of the ROS4 solver for test case 2.

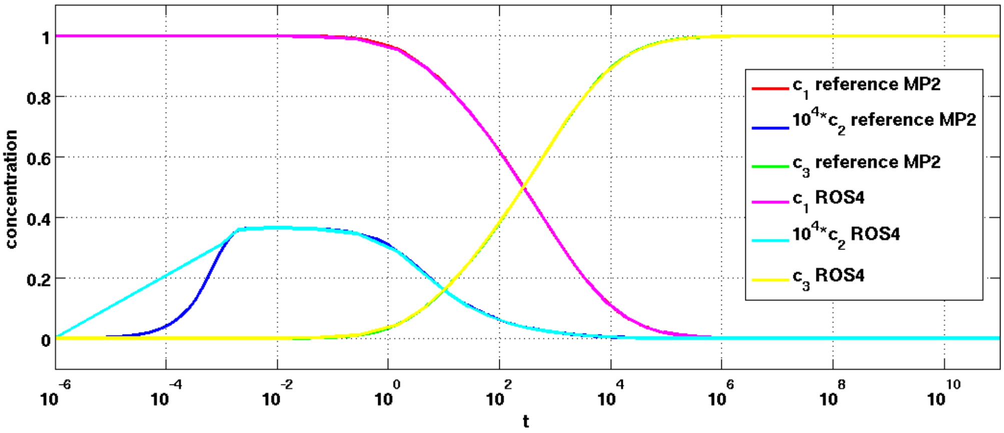


Figure 5.15: ROS4 applied to test case 3 with large tolerance values $rtol = 10^{-2}$ and $atol = 10^{-4}$. The red ($c_1(t)$), blue ($c_2(t)$) and green ($c_3(t)$) lines depict the analytical solution, where the pink, cyan and yellow lines show the approximated solution of $c_1(t)$, $c_2(t)$ and $c_3(t)$.

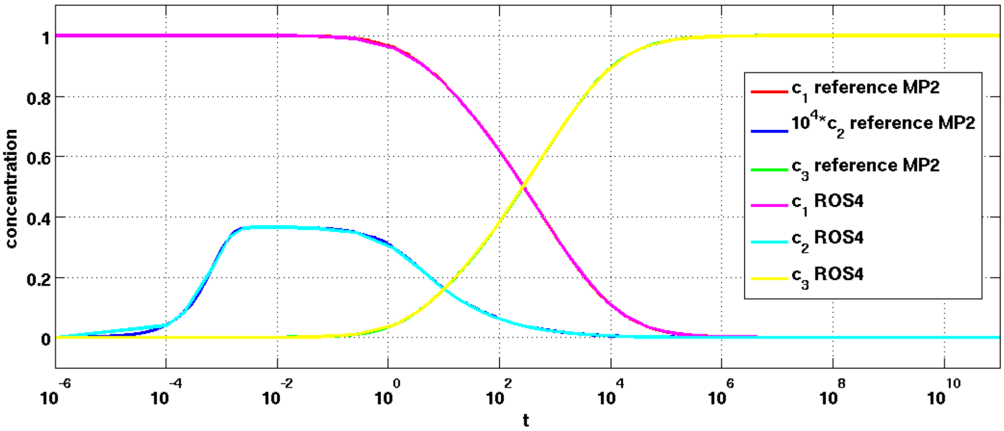


Figure 5.16: ROS4 applied to test case 3 with small tolerance values $rtol = 10^{-2}$ and $atol = 10^{-8}$. The red ($c_1(t)$), blue ($c_2(t)$) and green ($c_3(t)$) lines depict the analytical solution, where the pink, cyan and yellow lines show the approximated solution of $c_1(t)$, $c_2(t)$ and $c_3(t)$.

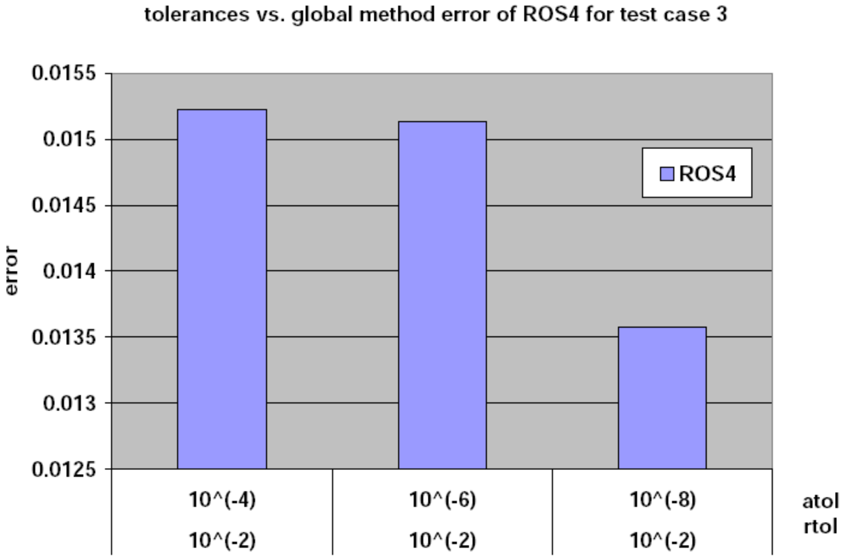


Figure 5.17: Effect of different values for *rtol* and *atol* on the global method error of ROS4 applied to test case 3. The x-axis shows the tolerances written below each other and the y-axis shows the computed error of the ROS4 solver for test case 3.

6 Discussion and conclusions

In this study traditional (explicit Euler and Runge Kutta) and advanced (modified and extended modified Patankar) numerical methods used in biogeochemical modelling are compared with methods commonly applied in numerical models for chemical reactions (Rosenbrock methods).
The comparison is based on three simple test cases:

1. a linear model, describing a one-time reaction between two constituents, for which an analytical solution is available
2. a non-linear model, describing mass exchanges between three constituents
3. a stiff ODE system, describing chemical reactions, running on different time scales, between three constituents.

The traditional numerical schemes are conservative (in sense of definition 4.3.2), where the advanced schemes are positive (in sense of definition 4.3.1) and conservative. These two properties are considered when comparing the numerical schemes. The main focus however, has been put on the accuracy and the computational effort of the numerical methods in this study.

The first test case - a two dimensional model - is taken from *Burchard et al.* [2003]. An example from the field of biogeochemistry is the transformation of iron-III-oxide to iron-II-oxide under anoxic conditions. Biogeochemical models, which address the role of iron chemistry parameterise this process, as e.g. done by *Weber et al.* [2007] (note, that ultimately a number of processes are involved in iron chemistry and hence more than just two variables have been included in the model by *Weber et al.* [2007]).
The second test case - a three dimensional model - is also taken from *Burchard et al.* [2003]. Such a system can be regarded as a simple marine biogeochemical ***N****utrient-****P****hytoplankton-****D****etritus (NPD)* model, where fluxes of elements (generally nitrogen) between the microalgae and dead organic and inorganic elemental (nitrogen) pool are computed. These rather simple NPD or NPZD (including zooplankton) - type models, see e.g. *Fasham et al.* [1990], are still the basis for current complex marine biogeochemical models, e.g. *Neumann et al.* [2002], *Weber et al.* [2007] and *Siddorn et al.* [2007].
The third test case is the so-called stiff Robertson test problem. It describes the kinetics of an auto-catalytic reaction given by *Robertson* [1966] and presents a typical example for chemical reactions that take place on significantly different time scales. As mentioned before, reactions running on different time scales are also included in biogeochemical models, e.g. in the iron model presented by *Weber et al.* [2007].
Hence, the results of the comparison of the three numerical methods, which are applied to all test cases are relevant and important for biogeochemical modelling.

For a direct comparison of the seven numerical schemes the step size of the RBMs has been restricted for test case 1 and 2. While comparing the performance of the methods differences occur depending on the test cases:
The restricted RBMs are as accurate as the other schemes, but more expensive than the explicit methods and the MP2 for test case 1. For test case 2 the RBMs give the most accurate results. Their computing time is similar to those of the Patankar schemes, but twice as high as that of the RK4.
Applying the RBMs in their original form (with adaptive step size), their results are more accurate for test case 1, because they can highly resolve the reactive phase of the problem. In contrast, their results are slightly less accurate for test case 2, due to the chosen tolerance values, which lead to larger steps in phases, where the temporal changes of the constituents are relatively small. For both test cases the adaptive step size mechanism does not decrease the execution time (in comparison to RBMs and fixed time step).
The explicit schemes as well as the EMP2 are not suitable for solving stiff problems, as shown e.g. by *Hairer and Wanner* [1991] and *Bruggeman et al.* [2007], and thus only the modified Patankar scheme of second order, as well as the Rosenbrock solvers with adaptive time stepping are applied to test case 3. The performance of the RBMs clearly shows the advantage of adaptive time stepping. They are significantly more accurate and faster compared to the MP2. The latter uses an exponential growing step size in order to adequately resolve the short term reactions.

The goal of this study was to compare currently used numerical schemes with the Rosenbrock methods and to investigate, whether the Rosenbrock solvers are suitable for application to biogeochemical models. In a case where the underlying problem is a chemical conversion process between two substances (as given in test case 1), the traditional explicit Runge Kutta methods of 2nd and 4th order are the most convenient. The same applies for more sophisticated model problems like the test case 2 presented here. The high computing time of the RBMs in general is caused by the necessity of solving n linear equations in each step of the calculation, although the effort for this computation has been minimised due to the fact that only one LU-decomposition is needed per step.
Even if the RBMs use the adaptive step size mechanism their computing time is higher than that of the explicit RKMs (by the same order of accuracy, see chapter 5) for the first two test cases, because there the reactions run on similar time scales. Thus, the RBMs can only choose larger time steps in the initial and final phase of the process, before the actual reactions start and almost no changes occur, respectively. However, these periods are too short to substantially save computing time.
Applying the RBMs to stiff ordinary differential equations, like the presented test case 3, the demand for computing time and accuracy is different from test case 1 and 2. On the one hand the RBMs save time by choosing the step size large in phases where reactions are slow (small temporal changes). On the other hand, they can highly resolve the short term reactions (large temporal changes), by choosing a small time step. Thus, the RBMs are more appropriate than the MP2 for these kinds of model problems.
Assuming, however, that in the underlying stiff problem short term processes react continuously and simultaneously together with long term processes, the advantage of an adaptive step size is lost. In this case, a small step size is also needed for the RBMs for

the whole integration period.

In summary, compared to the Patankar schemes the Rosenbrock solvers present an alternative for application to biogeochemical models, particularly in those, where the processes run on significantly different time scales. Finally, this study has shown that the differences between ROS4 and ROS3 are marginal; both are suitable for solving biogeochemical model problems.
After considering all results, advantages and disadvantages of the test cases and compared numerical schemes, the outcome of this study can be recapitulated as follows:
The tested Rosenbrock methods

1. give accurate results for all test cases,
2. give unconditional positive results, if the tolerance values are chosen sufficiently small,
3. give more accurate results than the modified Patankar schemes,
4. have higher computational effort than the explicit schemes, but similar to the modified Patankar schemes.

As a next step both tested Rosenbrock solvers will be included into the ***G**eneral **O**cean **T**urbulence **M**odel (GOTM)* to test the schemes within complex ecosystem models, which are coupled to a physical model. GOTM is a one-dimensional model of the water column, where the latter is split into boxes (not necessarily equidistant). Generally, the challenge of this envisaged work is to match the user specified model time step with the adaptive Rosenbrock time step. The former is taken for all reactions (biogeochemical and physical) in the whole water column, whereas the latter is used to compute the biogeochemical part. This will be done in the following way:

- if the model time step is smaller than the recommended Rosenbrock step, the former shall be taken
- vice versa, if the recommended Rosenbrock step is smaller than the model time step, this step size shall be taken, under the restriction that the last Rosenbrock step has to be cut off, if the model step size is overshot in order to ensure that all reactions end at the same time.

A possible solution could be, to split the whole integration interval in subintervals with length equal to the model step size and to solve the problem for each subinterval.

Symbol Index

α_{lj}	coefficient of Rosenbrock methods, Seite 29
γ	coefficient of Rosenbrock methods, Seite 29
γ_{lj}	coefficient of Rosenbrock methods, Seite 29
$\mathcal{O}(\cdot)$	Landau symbol, Seite 19
$\mathbf{J}$	Jacobian matrix, Seite 28
ε	global method error, Seite 19
a_{ij}	scalar variables of the Runge Kutta method, Seite 20
$b, \tilde{b}$	weighting coefficients, Seite 31
c^n	abbreviation of $c^n(t_n)$, Seite 18
$c^n(t_n)$	approximation of the ODE solution c at the new time step t_n, Seite 18
$c^{(k)}$	intermediate step, $k = 1, 2, \ldots$, at the old time step t_n, Seite 18
c^{n+1}	abbreviation of $c^{n+1}(t_{n+1})$, Seite 18
$c^{n+1}(t_{n+1})$	approximation of the ODE solution c at the new time step t_{n+1}, Seite 18
c_i^n	i-th component of the approximation c^n, Seite 18
$c_i^{(1)}$	i-th component of the intermediate step, Seite 18
D_i	destruction terms, Seite 34
d_{ij}	rate at which i-th constitution transforms into the j-th, Seite 34
d_i	scalar variables of the Runge Kutta method, Seite 20
EM	Euler forward method, Seite 19
$EMP1$	extended modified Patankar Euler method, Seite 44
$EMP2$	extended modified Patankar Runge Kutta method, Seite 45
Est	local error estimator of the Rosenbrock method, Seite 31
h	step size, $h \in \mathbb{R}_{\geq 0}$, Seite 18

J_i	i-th component of the Jacobian matrix, Seite 48
$MP1$	modified Patankar Euler method, Seite 44
$MP2$	modified Patankar Runge Kutta method, Seite 44
ODE	ordinary differential equation, Seite 15
p	order of consistency of a numerical method, Seite 19
$P1$	Patankar Euler method, Seite 43
$P2$	Patankar Runge Kutta method, Seite 44
P_i	production terms, Seite 34
p_{ij}	rate at which j-th constitution transforms into the i-th, Seite 34
RBM	Rosenbrock method, Seite 29
$RK2$	2-stage RKM, Seite 21
$RK4$	classical RKM, Seite 21
RKM	Runge Kutta method, Seite 19
$ROS3$	Rosenbrock solver of third order, Seite 48
$ROS4$	Rosenbrock solver of fourth order, Seite 48
s	stage number of numerical method, Seite 20
atol	absolute user-specified error tolerance for Rosenbrock method, Seite 31
NPD	Nutrient - Phytoplankton - Detritus, Seite 69
NPZD	Nutrient-Phytoplankton-Zooplankton-Detritus, Seite 33
rtol	relative user-specified error tolerance for Rosenbrock method, Seite 31
Tol	tolerance occurring in each step of the Rosenbrock method, Seite 32

Bibliography

Baretta, J., W. Ebenhöh, and P. Ruardij, The European regional seas ecosystem model, a complex marine ecosystem model, *Neth. J. Sea Res.*, *33*, 1995, http://www.ifm.zmaw.de/forschung/modelle/ersem/.

Bruggeman, J., H. Burchard, B. W. Kooi, and B. Sommeijer, A second-order, unconditionally positive, mass-conserving integration scheme for biochemical systems, *Applied Numerical Mathematics*, *57*, 2007.

Burchard, H., Applied Turbulence Modelling in Marine Waters, *Lecture Notes in Earth Sciences*, *100*, 2002.

Burchard, H., K. Bolding, and M. Villarreal, GOTM — A General Ocean Turbulence Model. Theory, applications and test cases, *Tech. Rep. EUR 18745 EN, European Commission*, 1999, http://www.gotm.net.

Burchard, H., E. Deleersnijder, and A. Meister, A high-order conservative Patankar-type discretisation for stiff systems of production - destruction equations, *Applied Numerical Mathematics*, *47*, 2003.

Burchard, H., E. Deleersnijder, and A. Meister, Application of modified Patankar schemes to stiff biogeochemical models for the water column, *Ocean Dynamics*, *55*, 2005.

Burchard, H., K. Bolding, W. Kühn, A. Meister, T. Neumann, and L. Umlauf, Description of a flexible and extendable physical-biogeochemical model system for the water column, *Journal of Marine Systems*, *61*, 2006.

Deleersnijder, E., J.-M. Beckers, J.-M. Campin, M. E. Mohajir, T. Fichefet, and P. Luyten, Some mathematical problems associated with the development and use of marine models, *The Mathematics of Models for Climatology and Environmen*, *NATO ASI 48*, 1997.

Evans, G., and J. Parslow, A model of annual plankton cycles, *Biological Oceanography*, *3*, 1985.

Fasham, M., H. Ducklow, and S. McKelvie, A nitrogen-based model of plankton dynamics in the oceanic mixed layer, *Journal of Marine Research*, *48*, 1990.

Forster, O., *Analysis 2- Differentialrechnung im R^n, gewöhnliche Differentialgleichungen*, Vieweg Wiesbaden, 1999.

Hairer, E., and G. Wanner, *Solving ordinary differential equations II*, Springer Berlin Heidelberg, 1991, http://www.unige.ch/ hairer/software.html.

Hairer, E., S. Nørsett, and G. Wanner, *Solving ordinary differential equations I*, Springer Berlin Heidelberg, 1991.

Hairer, E., C. Lubich, and G. Wanner, *Geometric numerical integration: Structure-preserving algorithms for ordinary differential equations*, Springer Berlin Heidelberg, 2006.

Heuser, H., *gewöhnliche Differentialgleichungen*, Teubner Wiesbaden, 1989.

Janelli, A., and R. Fazio, Adaptive stiff solvers at low accuracy and complexity, *Journal of Computational and Applied Mathematics*, *191*, 2006.

Kantha, L. H., A general ecosystem model for applications to primary productivity and carbon cycle studies in the global oceans, 2003.

Kaps, P., and P. Rentrop, Generalized Runge Kutta Methods of order 4 with step size control for stiff ordinary differential equation, *Numerical Mathematica*, *33*, 1979.

Kaps, P., and G. Wanner, A study of rosenbrock-type methods of high order, *Numerical Mathematics*, *38*, 1981.

Neumann, T., W. Fennel, and C.Kremp, Experimental simulations with an ecosystem model of the baltic sea: A nutrient load reduction experiment, *Global biogeochemical cycles*, *16*, 2002.

Patankar, S., *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corp. NY, 1980.

Robertson, H., The solution of a set of reaction rate equations, *Numerical Analysis, An Introduction, Academic Press*, 1966.

Rosenbrock, H. H., Some general implicit processes for the numerical solution of differential equations, *Computer Journal*, *5*, 1963.

Sandu, A., J. G. Verwer, M. V. Loon, G. R. Carmichael, F. A. Potra, D. Dabdub, and J. H. Seinfeld, Benchmarking stiff ode solvers for atmospheric chemistry problems I: implicit vs. explicit, *Atmospheric Environment*, *31*, 1997a.

Sandu, A., J. G. Verwer, M. V. Loon, G. R. Carmichael, F. A. Potra, D. Dabdub, and J. H. Seinfeld, Benchmarking stiff ode solvers for atmospheric chemistry problems II: Rosenbrock solvers, *Atmospheric Environment*, *31*, 1997b.

Seiler, C., Visualisierung in der computergestützten Biochemie, *Dissertation an der Johann-Wolfgang-Goethe-Universität in Frankfurt am Main*, 2006.

Siddorn, J. R., J. I. Allen, J. C. Blackford, F. J. Gilbert, J. T. Holt, M. W. Holt, J. P. Osborne, R. Proctor, and D. K. Mills, Modelling the hydrodynamics and ecosystem of the North-West European continental shelf for operational oceanography, *Journal of Marine Systems*, *65*, 2007.

Simeon, B., Numerik gewöhnlicher Differentialgleichungen, Skriptum zur Vorlesung im Wintersemester 2006/2007 TU München.

Strehmel, K., and R. Weiner, *Numerik gewöhnlicher Differentialgleichungen*, Teubner Wiesbaden, 1995.

Weber, L., C. Völker, A. Oschlies, and H. Burchard, Iron profiles and speciation of the upper water column at the Bermuda Atlantic Time-series Study site: a model based sensitivity study, *Biogeosciences*, *4*, 2007.

Wolfbrandt, A., A study of Rosenbrock processes with respect to order conditions and stiff stability, Ph.D. thesis, Chalmers University of Technology, Göteborg, Sweden, 1977.

Zou, R., and A. Gosh, Automated sensitivity analysis of stiff biochemical systems using fourth-order adaptive step size Rosenbrock integration method, *IEEE Proc.- Syst. Biol*, *153*, 2006.

Erklärung nach §28 Abs. 5 der Prüfungsordnung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig angefertigt und ohne fremde Hilfe verfasst habe, keine außer den von mir angegebenen Hilfsmitteln und Quellen dazu verwendet habe und die den benutzten Werken inhaltlich und wörtlich entnommenen Stellen als solche kenntlich gemacht habe.

Bianca Schippmann Rostock, 02.12.2008