A New Numerical Reduction Scheme for Fully Coupled Multicomponent Transport–Reaction Problems in Porous Media

S. Kräutle, P. Knabner

kraeutle@am.uni-erlangen.de, knabner@am.uni-erlangen.de

Institute for Applied Mathematics, University of Erlangen-Nuremberg, Erlangen, Germany.

Abstract

A new systematic approach for the efficient computation of the transport and reaction of a multi-species multi-reaction system is developed. The objective of this approach is to reduce the number of coupled nonlinear differential equations drastically. The reduction mechanism is able to handle both equilibrium and kinetic reactions and mobile and immobile species. Applications to biodegradation problems, modelled by the Monod approach, are given.

A numerical example demonstrates the speed-up of the presented reduction mechanism.

Keywords. porous media, reactive flow, multispecies

AMS Classification. 1899, 1832, 1875, 1894

1 Introduction

This article is concerned with the efficient and accurate numerical solution of a large system of partial differential equations (PDEs) and ordinary differential equations (ODEs) as they occur in the modeling of transport, chemical reactions and biodegradation below the earth’s surface. A rather general approach for a wide class of such ‘biogeochemical’ problems is proposed in this article.

The chemical species which are considered are divided into two classes: Mobile (dissolved) species and immobile (sorbed chemical species, minerals, immobile bacteria) ones. The system of equations for the concentrations consists of PDEs for the mobile species and ODEs for the immobile ones, all of them coupled through the reaction terms. The characteristic timescale of the different reactions may cover a large range making it desirable to model some reactions as equilibrium-controlled and others as kinetically controlled, leading to an algebro-differential system.

Mathematical formulations of such multi-species reaction or transport-reaction problems can be found for example in the work of Yeh and Tripathi [1989,1991], Saaltink, Ayora, Carrera [1998], where a survey on several approaches is given, in Holstad [2000], Chilakapati et al. [1998, 2000], Friedly and Rubin [1992], Robinson, Viswanathan and Valocchi [2000], and Fang, Yeh and Burgos [2003]. A main purpose of several formulations is to enable an efficient handling of equilibrium and kinetic reactions.

The computation of timesteps of such a system as a whole is called global implicit approach (GIA) or one-step method. In 3d and even in 2d, the computational

solution of such a system of equations is, concerning cpu time and also memory
requirements, arduous [Yeh and Tripathi, 1989]. Therefore, operator splitting
schemes which decompose each timestep of the full system into the subproblems
of transport and of reaction, are frequently considered as an alternative to the
GIA [Marchuk, 1990]). These schemes are divided into sequential iterative (SIA)
and sequential non-iterative approaches (SNIA). A prominent representative of the
SNIA splitting schemes is the Strang splitting [Strang, 1968]). The disadvantage of
SNIA schemes is the introduction of additional consistency errors (splitting errors,
see, e.g., Berkvens, Verwer et al., [2002], Valocchi and Malmstead [1992]), since the
processes of transport and of reaction are considered to take place one after the
other, whereas a correct handling should consider them as to be coupled. To over-
come this problem, SIA methods are used. They iterate between the subproblems
to avoid the splitting error. Beside the additional computational costs of such an
iterative process, poor stability properties are sometimes a problem [Steefel and
MacQuarrie, 1996, pp. 110–124, and references therein].

With the increase of computational power during the last years, GIA methods
became again more popular. Nevertheless, any possibility to reduce the computa-
tional costs of a GIA should be used. Therefore attempts were made to split the
full problems into subproblems without introducing splitting errors. This can be
achieved by reformulation of the equations and replacement of the concentration
vector by new variables, the so-called reaction invariants and the reaction extents,
as it is done by Friedly [1991], Friedly and Rubin [1992]. A new system of equations,
which is equivalent of the given one (i.e, there are a priori no additional consistency
errors), is derived. The resulting equations for the new variables can be treated by
a GIA or by a SNIA/SIA.

In this paper, a new approach for the reformulation of the transport-reaction
system and the introduction of new variables is proposed and discussed. The
objective of this approach is to construct a new problem, which is equivalent to the
given one, such that

(a) some of the resulting PDEs are linear and decoupled from the other equations,
and

(b) the number of coupled nonlinear PDEs for the transformed system is smaller
than for the original one.

The reduction of the number of equations and the decoupling is achieved by

(1.) linear combination of given equations,

(2.) introduction of new unknowns, and

(3.) resolution of local equations (AEs, ODEs) for certain variables.

Concerning (1.), the algorithm is partially based on ideas by Friedly [Friedly, 1991,
Friedly and Rubin, 1992]. For the treatment of batch problems, the idea of intro-
ducing new variables is very classical, though such methods are still subject to
scientific investigation and optimization (see Fang, Yeh and Burgos [2003] for a
modern algorithm for batch problems). The development of such techniques for
problems with coupled mobile and immobile species is more challenging. Concerning (3.), the resolution of local AEs and ODEs is similar to the direct substitution approach (DSA; see, e.g., Yeh and Tripathi [1989]). We will lay special emphasis on the following features of the method:

1. The existence of the local resolution functions can be guaranteed under rather general assumptions.

2. Different to other GIA with DSA techniques, the substitution process does not cause nonlinearly coupled species concentrations under the transport operator.

3. The number of coupled equations for the reduced system is for typical applications indeed clearly smaller than for the given problem. If the reaction network contains homogeneous equilibrium reactions (i.e., aqueous complexation reactions), the number of remaining coupled PDEs for our approach is smaller than for other general purpose reduction mechanism (section 4.3.2, 6.1).

The article is structured as follows: In section 2, we present our mathematical model of multispecies reactive transport problems. We will assume that all mobile species are convected by the same flow field, and that the equilibrium reactions are linearly independent and governed by the mass action law. No special assumptions will be required for the type of kinetically controlled reaction rates.

In section 3, the algorithm of Friedly is recalled. Friedly and Rubin consider to solve their reduced scheme with a splitting technique (SNIA/SIA) or as a GIA. In section 3.3, some difficulties are discussed which can occur when this method is used as a GIA to motivate our new approach.

In section 4, the new approach is proposed and discussed. In the present formulation of the method, homogeneous equilibrium reactions (aqueous complexation) and arbitrary kinetically-controlled reactions are considered.

In the sections 4 and 5, several examples of chemical transport-dispersion-reaction problems including biodegradation problems modelled by the Monod model are given.

A first numerical simulation showing the speed-up of the reduction method is presented in section 6. In that example, 4 mobile and one immobile species, and 2 equilibrium reactions and 1 kinetic reaction are considered. The reduction algorithm reduces the number of coupled nonlinear partial and ordinary differential equations from 5 to 1, and we observe a total cpu time of approximately 20% compared to the GIA for the nonreduced scheme.

To guarantee the applicability of the proposed method, the existence of the resolution functions is formulated as a theorem. The proofs are given in Appendix B. The general idea of a Newton method applied to a reduced set of equations with eliminated variables is recalled in Appendix 7.
2 Problem Formulation

Let us consider \( I \) mobile species \( X_1, ..., X_I \) and \( \bar{I} \) sorbed species \( X_{I+1}, ..., X_{I+\bar{I}} \). Let us denote its time- and space dependent concentrations by
\[
\begin{align*}
  \mathbf{c} &= (c_1, ..., c_I), \\
  \bar{\mathbf{c}} &= (\bar{c}_{I+1}, ..., \bar{c}_{I+\bar{I}})
\end{align*}
\]
The mobile species are convected by a given flow field and are subject to diffusion and dispersion. Let us assume that the underlying transport operator \( L \) is linear and the same for all mobile species. This assumption is justified if the species-dependent diffusion is negligible compared to dispersion.

The \( J \) chemical reactions can be written as
\[
\begin{align*}
  \sum_{i=1}^{I} s_{ij} X_i + \sum_{i=1+I}^{I+\bar{I}} s_{ij} \bar{X}_i & \xrightarrow{R_j} 0, \quad j = 1, ..., J,
\end{align*}
\]
where the stoichiometric coefficients \( s_{ij} \) form a \((I + \bar{I}) \times J\) matrix \( S \) with entries in \( \mathbb{Z} \) or in \( \mathbb{R} \) and \( R_j = R_j(c, \bar{c}) \) is the rate expression for reaction \( j \).

The \( I + \bar{I} \) mass balance equations are
\[
\begin{align*}
  \frac{d}{dt}(\theta c_i) + Lc_i &= \sum_{j=1}^{J} s_{ij} R_j(c, \bar{c}), \quad i = 1, ..., I \\
  \rho \frac{d}{dt} \bar{c}_i &= \sum_{j=1}^{J} s_{ij} R_j(c, \bar{c}), \quad i = I+1, ..., I+\bar{I}
\end{align*}
\]
where \( \theta \) represents the time and space dependent local fraction of the mobile fluid-phase volume, and \( \rho \) is the immobile phase bulk density, the \( c_i \) are the molar concentrations in the volume of the fluid phase and the \( \bar{c}_i \) are the moles per unit solid mass. We assume that the \( s_{ij} \) are independent of \( \theta, \rho, x \), while the rates \( R_j \) may depend on these parameters.

We can eliminate \( \theta \) and \( \rho \) on the left-hand side of the equations if we divide both sets of equations by \( \theta \) and substitute \( L'c_i := \frac{1}{\theta}(Lc_i + \frac{\partial \theta}{\partial t} c_i) \), \( \bar{c}'_i = \frac{\rho}{\theta} \bar{c}_i \), and \( R'(c, \bar{c}') = R_j(c, \bar{c}') \). For the simplicity of the notation, we denote \( \bar{c}_i, L', R'_j \) by \( \bar{c}_i, L, R_j \) again. We get
\[
\begin{align*}
  \frac{d}{dt} c_i + Lc_i &= \sum_{j=1}^{J} s_{ij} R_j(c, \bar{c}), \quad i = 1, ..., I \\
  \frac{d}{dt} \bar{c}_i &= \sum_{j=1}^{J} s_{ij} R_j(c, \bar{c}), \quad i = I+1, ..., I+\bar{I}.
\end{align*}
\]
\( S \) consists of a \( I \times J \) block \( S_1 \) of stoichiometric coefficients for the mobile species and a \( I+\bar{I} \times J \) block \( S_2 \) of coefficients for the immobile species:
\[
S = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}
\]
Using the vector notation \( \mathbf{R} = (R_1, ..., R_J) \) we get the compact notation
\[
\begin{align*}
  \frac{d}{dt} \mathbf{c} + L \mathbf{c} &= S_1 \mathbf{R}(c, \bar{c}) \\
  \frac{d}{dt} \bar{c} &= S_2 \mathbf{R}(c, \bar{c})
\end{align*}
\]
(2)
for (1).

If the rates are given by the law of mass action, they read

\[ R_j(c, \bar{c}) = k_f^j \prod_{i=1}^{I+\bar{I}} c_i^{-s_{ij}} - k_b^j \prod_{i=1}^{I+\bar{I}} c_i^{+s_{ij}} \]  

(3)

where for the sake of simplicity we have omitted the bar atop the \( \bar{c}_i \). Other rate laws (empirical, mechanistic, or mass action with non-ideal activities) can be considered. For technical reasons we will assume that \( \frac{\partial R_j}{\partial c_i}, \frac{\partial R_j}{\partial \bar{c}_i} \) are bounded all over the computational domain for all species \( c, \bar{c}_i \) and all kinetic reactions \( R_j \).

If a reaction is assumed to be at local equilibrium,

\[ R_j(c, \bar{c}) = 0 \]  

(4)

holds. Application of \( \ln \) to (4) in the case of mass action (3) leads to the well known relation

\[ Q_j(c, \bar{c}) := \sum_{i=1}^{I} s_{ij} \ln c_i + \sum_{i=I+1}^{I+\bar{I}} s_{ij} \ln \bar{c}_i - k_j = 0, \]  

(5)

where the equilibrium constants \( k_j \) are defined by \( k_j = \ln(k_f^j / k_b^j) \). If all reactions are at equilibrium, the vector \( Q = (Q_1, ..., Q_J)^t \) of algebraic equilibrium conditions reads

\[ Q(c, \bar{c}) = S_1^t \ln c + S_2^t \ln \bar{c} - K = 0 \]  

(6)

with obvious definition of vector \( K \).

3 Friedly’s method

In this section we recall the method by Friedly [1991], Friedly and Rubin [1992], and we will discuss some problems which occur when this method is used as a GIA method. This consideration will lead to the new reduction scheme.

3.1 The operational equations

To solve the system (2), Friedly and Rubin construct an \((I + \bar{I}) \times (I + \bar{I} - J)\)-matrix \( S^T \) of linearly independent columns which are orthogonal to the columns of \( S \). Denoting by \( J_1 \) the maximum number of linearly independent columns of \( S_1 \), they construct an \( I \times J_1 \)-matrix \( S_1^* \) consisting of linearly independent columns of \( S_1 \), and an \( J_1 \times J \)-matrix \( A \) such that

\[ S_1 = S_1^* A \]  

(7)

holds. \( S_1^* \) is an \( I \times (I - J_1) \)-matrix constructed from columns which are orthogonal to the columns of \( S_1 \). Then, problem (2) (with certain initial and boundary
The system is equivalent to the system

\[
\begin{align*}
\frac{d}{dt} \eta &= 0 \\
\frac{d}{dt} \eta_m + L \eta_m &= -L (S_1^T S_1)^{-1} S_1^T (I_0) S_1^T \eta \\
\frac{d}{dt} \xi &= R \left( S \xi + S^T \eta + \begin{pmatrix} S_1^T \xi_m + S_1^T \eta_m \\ 0 \end{pmatrix} \right) \\
\frac{d}{dt} \xi_m + L(\xi_m + A \xi) &= -L(S_1^T S_1)^{-1} S_1^T (I_0) S_1^T \eta
\end{align*}
\]

where the new unknowns are the vectors \( \eta, \eta_m, \xi, \xi_m \) of size \( I + I - J, I - J, J, J_1 \), respectively, and where \( (I_0) \) is an \( I \times (I + I) \)-matrix. The initial and boundary conditions for the new unknowns can be extracted from the initial and boundary conditions for \( c, \bar{c} \).

\( \eta \) is constant during the whole simulation. The second block of (8) consists of uncoupled linear equations. The third and fourth block of (8) form a coupled nonlinear problem. Friedly and Rubin [1992, p. 1947f] consider to solve the remaining coupled nonlinear blocks 3-4 globally implicitly (GIA) or sequentially (SIA) or iteratively (SNIA). Since we want to avoid the possible disadvantages of splitting methods, let us consider (8) treated by the GIA: Each timestep consists of solving the linear scalar equations for \( \eta_m \) followed by a Newton or modified Newton method to solve blocks 3-4. The computational costs are governed by the size of the nonlinear system, which is \( J + J_1 \). The solution of (8) instead of (2) can be expected to be efficient (only) if \( J + J_1 \) is smaller than \( I + I \), which is the size of the original problem. Note that there may be more nonzero entries in the Jacobian for blocks 3-4 of (8) than in the Jacobian for (2), as each reaction term \( R_j \) usually depends on all species \( \xi_i, \xi_m, \bar{i}, \bar{m} \), but only on a limited number of species \( c_i, \bar{c}_i \). This influences the computational costs for iterative linear solvers used within the (modified) Newton scheme. A possible further reduction of blocks 3-4 (in the sense of a DSA) seems desirable and is discussed in section 3.3.

In Friedly’s scheme, each reaction \( R_j \) occurs only once. This important feature enables the introduction of equilibrium reactions. The following section 3.2 deals with Friedly’s scheme for the case where both equilibrium and kinetic reactions take place.

### 3.2 Friedly’s method for equilibrium-controlled reactions

The property that each reaction term appears only once in the scheme (8) makes it simple to handle equilibrium-controlled reactions: Suppose that \( J_{eq} \) of the reactions are so fast that they can be regarded as equilibrium reactions, and the other \( J_{neq} = J - J_{eq} \) are slow. We sort the vector of reaction rates, beginning with the fast ones:

\[
R = \begin{pmatrix} R_{eq} \\ R_{neq} \end{pmatrix}
\]

Similarly, we devide the vector \( \xi \) into a part \( \xi_{eq} \) of size \( J_{eq} \) and a part \( \xi_{neq} \) of size \( J_{neq} \), and we devide the matrices \( S \) and \( A \) into a left part of \( J_{eq} \) columns and a
3.3 Further reduction of the scheme

right part of $J_{\text{neq}}$ columns:

$$\xi = \begin{pmatrix} \xi_{\text{eq}} \\ \xi_{\text{neq}} \end{pmatrix}, \quad S = \begin{pmatrix} S_{\text{eq}} & S_{\text{neq}} \end{pmatrix}, \quad A = \begin{pmatrix} A_{\text{eq}} & A_{\text{neq}} \end{pmatrix}$$

Like $S$, also $S_{\text{eq}}$ and $S_{\text{neq}}$ are decomposed into a block of $I$ and a block of $\bar{I}$ rows, one corresponding to the mobile and one to the immobile species:

$$S_{\text{eq}} = \begin{pmatrix} S_{11} \\ S_{21} \end{pmatrix}, \quad S_{\text{neq}} = \begin{pmatrix} S_{12} \\ S_{22} \end{pmatrix}$$

Obviously, $S_{\text{eq}} \xi_{\text{eq}} + S_{\text{neq}} \xi_{\text{neq}}$, $A \xi = A_{\text{eq}} \xi_{\text{eq}} + A_{\text{neq}} \xi_{\text{neq}}$ hold, and equation (8) becomes

$$0 = R_{\text{eq}} \begin{pmatrix} c \\ \bar{c} \end{pmatrix}$$

$$\frac{d}{dt} \xi_{\text{neq}} = R_{\text{neq}} \begin{pmatrix} c \\ \bar{c} \end{pmatrix}$$

$$\frac{d}{dt} \xi_m + L(\xi_m + A_{\text{eq}} \xi_{\text{eq}} + A_{\text{neq}} \xi_{\text{neq}})
= -L(S_1^T S_1)^{-1} S_1^T (I \ 0) S_1^\top \eta$$

where we have omitted the decoupled equations for $\eta$ and $\eta_m$ and where we have used the identity

$$\begin{pmatrix} c \\ \bar{c} \end{pmatrix} = \left( S_{\text{eq}} \xi_{\text{eq}} + S_{\text{neq}} \xi_{\text{neq}} + S_1^\top \eta + \begin{pmatrix} S_1^* \xi_m + S_1^* \eta_m \\ 0 \end{pmatrix} \right)$$

for abbreviation.

Hence, the resulting coupled nonlinear problem consists of $J_{\text{eq}}$ AEs (9), $J_{\text{neq}}$ ODEs (10) and $J_1$ PDEs (11).

### 3.3 Further reduction of the scheme

The scheme (8), where all reactions are kinetically controlled, is a special case of the mixed equilibrium/kinetic scheme (9)-(11) for $J_{\text{neq}} = 0$. Hence, it is sufficient to consider only scheme (9)-(11) in the following.

Since it is not obvious that the new coupled problem (9)-(11) of size $J + J_1$ is (much) smaller than the original problem of size $I + \bar{I}$, it is desirable to further reduce the scheme. This could be done by solving the local equations (9) or (10) or both (9) and (10) for the variable $\xi_{\text{eq}}$ and/or $\xi_{\text{neq}}$ and eliminate these variables in the remaining equations (see Appendix 7). In general, the resolution functions are only given implicitly. Using the implicit function theorem, we can state the following results:
Theorem 1

(i) If the equilibrium reactions are governed by mass action kinetics, there is a resolution function $\xi_{eq}(\xi_{neq}, \xi_m)$ for block (9).

(ii) If (10) is discretized with the implicit Euler method and if the timestep size $\Delta t$ is sufficiently small, there is a resolution function $\xi_{neq}(\xi_{eq}, \xi_m)$ for block (10).

(iii) If the equilibrium reactions are controlled by mass action law and if $\Delta t$ is sufficiently small, there is a resolution function $\xi(\xi_m) = (\xi_{eq}(\xi_m), \xi_{neq}(\xi_m))$ for blocks (9)-(10).

Note that we have omitted any dependence on $\eta$, since these variables are computed in advance and are considered to be known.

Proof: See Appendix B.

Let us discuss the suitability of the Friedly and Rubin method for the implicit resolution of local equations proposed in this section. Unfortunately, if we introduce the resolution function $\xi_{eq} = \xi_{eq}(\xi_{neq}, \xi_m)$, or $\xi_{neq} = \xi_{neq}(\xi_{eq}, \xi_m)$, or $\xi = \xi(\xi_m)$, in block (11), the nonlinear expressions occur under the transport operator $L$. This has the following two disadvantages:

1. The matrix of the linear system arising from (11) has much more non-zero entries in the blocks beside the diagonal blocks. Especially in 2d and even more in 3d computations, the effect on the cpu time is not negligible.
2. The numerical behaviour of such a system is difficult to predict.

Note that the same difficulties occur for the method by Chilkapati et al. [1998]: If the method is used as a GIA, and the AEs are substituted into to PDEs [p. 1774, equation (31)], nonlinear coupling terms occur under the transport operator.

To circumvent such problems one might substitute

$$\tilde{\xi}_m := \xi_m + A\xi$$

in Friedly’s scheme (9)-(11). We get, using (13) and (7) in (12),

$$S_{eq}\xi_{eq} + S_{neq}\xi_{neq} + \begin{pmatrix} S_1^* \xi_m + S_1^\top \eta_m \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} S_1 \xi \\ S_2 \xi \end{pmatrix} + \begin{pmatrix} S_1^* \xi_m + S_1^\top \eta_m \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} S_1 \xi \\ S_2 \xi \end{pmatrix} + \begin{pmatrix} S_1^* \tilde{\xi}_m - S_1^* A\xi + S_1^\top \eta_m \\ 0 \end{pmatrix}$$

$$= \begin{pmatrix} S_1^* \tilde{\xi}_m + S_1^\top \eta_m \\ S_2 \xi \end{pmatrix}$$
and the resulting equations are

\[
0 = R_{eq} \left( S^T \eta + \begin{pmatrix} S^*_m + S^T_1 \eta_m \\ S_{21} \xi_{eq} + S_{22} \xi_{neq} \end{pmatrix} \right)
\]

\[
\frac{d}{dt} \xi_{neq} = R_{neq} \left( S^T \eta + \begin{pmatrix} S^*_m + S^T_1 \eta_m \\ S_{21} \xi_{eq} + S_{22} \xi_{neq} \end{pmatrix} \right)
\]

\[
\frac{d}{dt} (\tilde{\xi}_m - A_{eq} \xi_{eq} - A_{neq} \xi_{neq}) + L \tilde{\xi}_m = -L(S^T_1 S_1)^{-1} S^T_1 (I_0) S^T \eta
\]

where the PDE block has the same size \( J_1 \) as (11). Now, different from (8) or (9)-(11), the introduction of the resolution equation of block 1-2 of (14) in block 3 does not destroy the sparsity of the off-diagonal blocks of the linear system of equations, and there is no nonlinear transport term. Instead, we get a kind of generalized sorption equation.

The severe problem with formulation (14) is that a resolution equation of block 1 with respect to \( \xi_{eq} \) only exists if

matrix \( S_{21} \) consists of linearly independent columns.

(For a proof see Appendix B.) This is a rather restrictive assumption: If there is, for example, just one equilibrium reaction in the mobile phase, \( S_{21} \) has a zero column, and (15) is violated.

### 4 A New Method of Reduction

Let us summarize the problems with the method of section 3: If it is used as a global implicit method, the coupled nonlinear system consists of \( J + J_1 \) equations, i.e., it might be not significantly smaller (or even bigger) than the original system of size \( I + \bar{I} \). To further reduce the number of coupled equations, the local equations can be solved and substituted into the remaining equations. Beside a reduction of the number of equations, this procedure leads to many more non-zero entries in the system matrix making the solution rather time consuming. A substitution (13) can avoid this problem, but then an elimination of local equations is possible only for situations, where (15) holds.

In this section, another reduction mechanism is developed. The derived system has no coupling terms under the transport operator, and the number of coupled PDEs is less or equal to the number of nonequilibrium reactions and independent of the number of equilibrium reactions in the mobile phase.

#### 4.1 The derivation of the operational equations

The idea of the new method is to transform the two blocks of (2) separately: Let \( J_1, J_2 \) be the number of linearly independent columns in the matrix \( S_1, S_2 \), respectively. For \( S_i, i = 1, 2 \), we define \( S^*_i \) as a matrix consisting of a maximum system of linearly independent columns of \( S_i \) and \( A_i \) such that

\[
S_i = S^*_i A_i
\]
holds. $A_1$ is a matrix of size $J_1 \times J$ and $A_2$ of size $J_2 \times J$. For $S_i^*$ we define $S_i^T$ consisting of a maximum set of linearly independent columns that are orthogonal to each column of $S_i^*$. By construction,

$$S_i^T S_i^* = 0, \quad S_i^* S_i^T = 0, \quad i = 1, 2,$$

(17)

hold. Since the columns of $S_i^*$ are linearly independent, the matrix $S_i^* S_i^T$ is regular. By using (16) in (2) and then multiplication of the $i$-th block of (2), $i = 1, 2$, by

$$(S_i^T S_i^*)^{-1} S_i^T$$

and $(S_i^T S_i^*)^{-1} S_i^T$

we derive the following four blocks:

$$
\begin{align*}
(S_1^T S_1^*)^{-1} S_1^T (\frac{d}{dt} c + Lc) &= 0 \\
(S_2^T S_2^*)^{-1} S_2^T \frac{d}{dt} \bar{c} &= 0 \\
(S_1^* S_1^*)^{-1} S_1^T (\frac{d}{dt} c + Lc) &= A_1 R (c, \bar{c}) \\
(S_2^* S_2^*)^{-1} S_2^T \frac{d}{dt} \bar{c} &= A_2 R (c, \bar{c})
\end{align*}
$$

(19)

Since the matrices and the differential operators in system (19) commute, we can substitute

$$
\begin{align*}
\eta &= (S_1^T S_1^*)^{-1} S_1^T c, \\
\bar{\eta} &= (S_2^T S_2^*)^{-1} S_2^T \bar{c}, \\
\xi &= (S_1^* S_1^*)^{-1} S_1^T c, \\
\bar{\xi} &= (S_2^* S_2^*)^{-1} S_2^T \bar{c}.
\end{align*}
$$

(20)

One can show that

$$
\begin{align*}
c &= S_1^T \xi + S_1^T \eta \\
\bar{c} &= S_2^T \bar{\xi} + S_2^T \bar{\eta}
\end{align*}
$$

(21)

(22)

hold (see Appendix B). The vectors $(\xi, \eta) \in \mathbb{R}^J$, $(\bar{\xi}, \bar{\eta}) \in \mathbb{R}^\bar{J}$ are representations of the vectors $c, \bar{c}$, respectively, with respect to another basis. $\xi$ and $\eta$ are linear combinations of mobile species, and $\bar{\xi}$, $\bar{\eta}$ are linear combinations of immobile species.

System (19) is equivalent to

$$
\begin{align*}
\frac{d}{dt} \eta + L\eta &= 0 \quad (I - J_1 \text{ equ.}) \\
\frac{d}{dt} \bar{\eta} &= 0 \quad (I - J_2 \text{ equ.}) \\
\frac{d}{dt} \xi + L\xi &= A_1 R \left( \begin{array}{c} S_1^T \xi + S_1^T \eta \\ S_2^T \bar{\xi} + S_2^T \bar{\eta} \end{array} \right) \quad (J_1 \text{ equ.}) \\
\frac{d}{dt} \bar{\xi} &= A_2 R \left( \begin{array}{c} S_1^T \xi + S_1^T \eta \\ S_2^T \bar{\xi} + S_2^T \bar{\eta} \end{array} \right) \quad (J_2 \text{ equ.})
\end{align*}
$$

(23)

In the formulation (23) we have two vectors of ‘reaction invariants’ $\eta, \bar{\eta}$, one consisting of mobile species and one consisting of immobile species. Corresponding to $\eta, \bar{\eta}$, there are two vectors $\xi, \bar{\xi}$ consisting of mobile/immobile concentrations governed
by the progress of the chemical reactions. The initial and boundary conditions for (23) are computed from the initial and boundary conditions for \(c, \bar{c}\) through (20).

The number of coupled nonlinear equations for this method is \(J_1 + J_2\) which is, especially if there are homogeneous reactions in the mobile phase, usually smaller than \(J_1 + J\) in section 3 \((J_2 \leq J)\). On first view, a disadvantage of this formulation seems to be that in a general setting, it is not obvious how to introduce equilibrium reactions, as some reaction expressions can appear in several equations of (23). At least under certain restrictions on the stoichiometry of equilibrium reactions, the introduction of equilibrium reactions is possible. Let us develop the technique by an example:

### 4.2 Example 1

Let us consider the chemical reaction network

\[
\begin{align*}
2A & \xrightarrow{R_1} B \\
B + C & \xrightarrow{R_2} D \\
\bar{D} & \xrightarrow{R_3} \bar{E}
\end{align*}
\]

consisting of 3 mobile species \(A, B, C\) and 2 immobile species \(\bar{D}, \bar{E}\) and 3 chemical reactions \(R_1, R_2, R_3\). The corresponding nonlinear system of PDEs reads

\[
\begin{align*}
A + LA &= -2R_1(A, B) \\
\dot{B} + LB &= R_1(A, B) - R_2(B, C, \bar{D}) \\
\dot{C} + LC &= -R_2(B, C, \bar{D}) \\
\bar{D} &= R_2(B, C, \bar{D}) - R_3(B, \bar{D}, \bar{E}) \\
\dot{\bar{E}} &= R_3(B, \bar{D}, \bar{E})
\end{align*}
\]

Linear combinations of the equations within the block of mobile species and within the block of immobile species leads to a system similar to (19):

\[
\begin{align*}
(d/dt + L)(-\frac{1}{2}A) &= R_1(A, B) \\
(d/dt + L)(-C) &= R_2(B, C, \bar{D}) \\
(d/dt + L)(\frac{1}{2}A + B - C) &= 0 \\
\frac{d}{dt}(\bar{D} + \bar{E}) &= R_2(B, C, \bar{D}) \\
\frac{d}{dt} \bar{E} &= R_3(B, \bar{D}, \bar{E})
\end{align*}
\]

Substituting

\[
\begin{align*}
\eta_1 &= \frac{1}{2}A + B - C, \\
\xi_1 &= (\frac{1}{2}A), \\
\xi_2 &= -C,
\end{align*}
\]

we get

\[
\begin{align*}
(d/dt + L)\xi_1 &= R_1 \\
(d/dt + L)\xi_2 &= R_2 \\
(d/dt + L)\eta_1 &= 0 \\
\frac{d}{dt} \xi_1 &= R_2 \\
\frac{d}{dt} \xi_2 &= R_3
\end{align*}
\]
Note that the transformation of problem (25) into a formulation of type (28) is not unique; addition of a multiple of the third line of (26) to the first or second line would lead to the same shape (28).

Now let us apply the systematic approach of section 4.1 to (24)/(25): The stoichiometric matrix corresponding to (24) is

$$S = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = \begin{pmatrix} -2 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}.$$ 

The extraction of the linearly independent columns yields

$$S_1^* = \begin{pmatrix} -2 \\ 1 \\ 0 \end{pmatrix}, \quad S_2^* = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix},$$

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

and for the orthogonal complement of $S_1^*$ we may choose

$$S_1^\top = \begin{pmatrix} 1/2 \\ 1 \\ -1 \end{pmatrix},$$

and $S_2^\top = ()$ is the empty matrix. Hence, the algorithm of section 4.1 yields a system of shape (28). However, due to the nonuniqueness of the transformation from (25) to (26), the relation between $\xi_1, \xi_2, \eta_1$ and the original concentrations is not exactly the same:

$$\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \frac{1}{9} \begin{pmatrix} -4 & 1 & -1 \\ -2 & -4 & -5 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix},$$

$$\eta_1 = \frac{4}{9} \begin{pmatrix} 1/2 & 1 & -1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix},$$

$$\begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} D \\ E \end{pmatrix}.$$

The systematic approach for this example shows under which conditions and how to introduce equilibrium reactions: Since all nonzero columns in $S_i$, $i = 1, 2$, are linearly independent, $A_i$ consists of a unit matrix and possibly some zero columns. This has the consequence that each reaction $R_j$ occurs in each of the last two blocks of (23) at most once. Those reactions which are heterogeneous ($R_2$ in the example) occur exactly once in each block, i.e., twice in the whole system. Those
4.3 Equilibrium reactions in the mobile phase

reactions which are homogeneous in the mobile or in the immobile phase \((R_1, R_3)\) occur exactly once in the whole system. Therefore, homogeneous fast reactions \(R_j\) can easily be considered as equilibrium reactions by replacing the one equation containing \(R_j\) by the corresponding equilibrium condition \(R_j = 0\). We conclude that the introduction of equilibrium reactions is straightforward with our approach, if we assume that

all equilibrium reactions are homogeneous reactions. \((29)\)

The systematic handling of the case \((29)\) is described in section 4.3.

4.3 Equilibrium reactions in the mobile phase

4.3.1 Replacement of rate expressions by equilibrium expressions

As in section 3.2, we decompose

\[
R = \begin{pmatrix} R_{eq} \\ R_{neq} \end{pmatrix}, \quad \xi = \begin{pmatrix} \xi_{eq} \\ \xi_{neq} \end{pmatrix},
\]

\[
S_i = \begin{pmatrix} S_i^{eq} \\ S_i^{neq} \end{pmatrix}, \quad i = 1, 2.
\]

Hence,

\[
S = \begin{pmatrix} S_1 \\ S_2 \end{pmatrix} = \begin{pmatrix} S_1^{eq} & S_1^{neq} \\ S_2^{eq} & S_2^{neq} \end{pmatrix}
\]

For the considerations of the rest of section 4.3, we suppose that all equilibrium reactions are homogeneous reactions in the mobile phase, i.e.,

\[
S_{eq}^2 = 0, \quad S = \begin{pmatrix} S_{eq} \\ 0 \end{pmatrix} \begin{pmatrix} S_{neq}^{eq} \\ S_{neq}^{neq} \end{pmatrix}.
\]

This restriction may be justified by the observation that reactions involving sorbed species are often slower than homogeneous reactions in the mobile phase. Note that fast reactions involving sorbed species are still possible; however, they have to be modelled as kinetic reactions (with large reaction coefficients \(k_{fj}^{eq}, k_{bj}^{eq}\)).

If the assumption \((30)\) is met, the columns of \(S_{eq}^1\) are linearly independent, i.e., \(S_{eq}^1 = S_{eq}^{eq}\) holds. Without loss of generality we can choose the linearly independent columns of \(S_1\) for the construction of \(S_1^*\) such that \(S_1^*\) has the structure

\[
S_1^* = \begin{pmatrix} S_{eq}^{1} \\ S_{neq}^{1' eq} \end{pmatrix},
\]

with \(S_{neq}^{1'}\) being a submatrix of \(S_{neq}^{1}\). Then \(A_1, A_2\) have the block structure

\[
A_2 = \begin{pmatrix} 0 & \tilde{A}_2 \end{pmatrix}, \quad A_1 = \begin{pmatrix} I & A_{12} \\ 0 & A_{22} \end{pmatrix}
\]

with \(J_{eq}\) columns in the left blocks and \(J_{neq}\) columns in the right blocks.
The system (23) is then equivalent to

\[
\frac{d}{dt} \eta + L \eta = 0 \quad (I - J_1 \text{ equ.})
\]
\[
\frac{d}{dt} \bar{\eta} = 0 \quad (I - J_2 \text{ equ.})
\]
\[
\frac{d}{dt} \xi_{eq} + L \xi_{eq} = R_{eq} \begin{pmatrix} c \\ \bar{c} \end{pmatrix} + A_{12} R_{neq} \begin{pmatrix} c \\ \bar{c} \end{pmatrix} \quad (J_{eq} \text{ equ.})
\]
\[
\frac{d}{dt} \xi_{neq} + L \xi_{neq} = A_{22} R_{neq} \begin{pmatrix} c \\ \bar{c} \end{pmatrix} \quad (J_1 - J_{eq} \text{ equ.})
\]
\[
\frac{d}{dt} \bar{\xi} = \tilde{A}_2 R_{neq} \begin{pmatrix} c \\ \bar{c} \end{pmatrix} \quad (J_2 \text{ equ.})
\]

where

\[
\begin{pmatrix} c \\ \bar{c} \end{pmatrix} = \begin{pmatrix} S_{eq}^1 \xi_{eq} + S_{neq}^1 \xi_{neq} + S_{1}^T \eta \\ S_2^\xi + S_2^T \bar{\eta} \end{pmatrix}
\]

is set for abbreviation. Hence, thanks to the restriction (30), each equilibrium expression occurs only once in our system (32), and can be replaced by the corresponding equilibrium condition \( R_j = 0 \):

System (32) then reads

\[
0 = R_{eq} \begin{pmatrix} c \\ \bar{c} \end{pmatrix} \quad (J_{eq} \text{ equ.})
\]
\[
\frac{d}{dt} \xi_{neq} + L \xi_{neq} = A_{22} R_{neq} \begin{pmatrix} c \\ \bar{c} \end{pmatrix} \quad (J_1 - J_{eq} \text{ equ.})
\]
\[
\frac{d}{dt} \bar{\xi} = \tilde{A}_2 R_{neq} \begin{pmatrix} c \\ \bar{c} \end{pmatrix} \quad (J_2 \text{ equ.})
\]

were we have omitted the decoupled equations for \( \eta, \bar{\eta} \).

If we assume that all equilibrium reactions are controlled by the law of mass action, the application of representation (5) to each equilibrium controlled reaction and denoting the vector \((Q_1, ..., Q_{J_{eq}})\) by \(Q\), the first block of (33) reads

\[
Q \begin{pmatrix} c \\ \bar{c} \end{pmatrix} = Q \begin{pmatrix} \text{c}^1_{eq} \xi_{eq} + \text{c}^1_{neq} \xi_{neq} + \text{c}^1_1 \eta \\ \text{c}^2_2 \xi + \text{c}^2_2 \bar{\eta} \end{pmatrix}
\]
\[
= \left( \text{c}^1_{eq} \right)^{t} \ln \left( \text{c}^1_{eq} \xi_{eq} + \text{c}^1_{neq} \xi_{neq} + \text{c}^1_1 \eta \right) - K
\]
\[
= \left( \text{c}^1_{eq} \right)^{t} \ln \left( \text{c}^1_{eq} \xi_{eq} + \text{c}^1_{neq} \xi_{neq} + \text{c}^1_1 \eta \right) - K = 0.
\]
4.3 Equilibrium reactions in the mobile phase

So (33) becomes

\[
(S_{eq}^1)^t \ln (S_{eq}^1 \xi_{eq} + S_{eq}^1 \xi_{neq} + S_1^T \eta) = K
\]

\[
\frac{d}{dt} \xi_{neq} + L \xi_{neq} = A_{22} R_{neq} \left( \frac{c}{c} \right)
\]

(35)

\[
\frac{d}{dt} \bar{\xi} = \bar{A}_2 R_{neq} \left( \frac{c}{c} \right)
\]

(36)

Note that a transformation similar to that of this section can be done in the (less likely) case that all equilibrium reactions are homogeneous reactions in the immobile phase, i.e., \(S_{eq}^1 = 0\) instead of (30).

4.3.2 The implicit elimination process for local equations

We are considering the DSA-like reduction mechanism of Appendix A for the following three systems:

(i) In system (23) (where all reactions are kinetically controlled), we want to solve the last block, after a time-discretization, e.g., by the implicit Euler method, for \(\bar{\xi}\) as a function of \(\xi\). We remain with the decoupled blocks

\[
\frac{d}{dt} \eta + L \eta = 0
\]

\[
\frac{d}{dt} \bar{\eta} = 0
\]

\[
\frac{d}{dt} \xi + L \xi = A_1 R \left( \frac{S_1^\xi \xi + S_1^\eta \eta}{S_2^\xi \xi + S_2^\eta \eta} \right)
\]

where \(\bar{\xi}(\xi)\) is defined as the resolution function of

\[
\frac{1}{\Delta t} (\bar{\xi} - \bar{\xi}_{old}) = A_2 R \left( \frac{S_1^\xi \xi + S_1^\eta \eta}{S_2^\xi \xi + S_2^\eta \eta} \right).
\]

(37)

which corresponds to the block \(f_2 = 0\) of local equation in Appendix A. The block \(f_1 = 0\) of Appendix A, for which the Newton iteration has to be done, corresponds to block 3 of (36).

(ii) In the presence of equilibrium reactions, let us assume that no immobile species participates in the equilibrium reactions (condition (30)), and that the mass action law holds for all equilibrium reactions.

(a) Under these conditions, we want to solve block 1 in (33)/(35) for \(\xi_{eq}\) as a function of \(\xi_{neq}, \xi\). (33)/(35) becomes

\[
\frac{d}{dt} \xi_{neq} + L \xi_{neq} = A_{22} R_{neq} \left( \frac{S_{eq}^1 \xi_{eq}(\xi_{neq}, \bar{\xi}) + S_{eq}^1' \xi_{eq} + S_1^T \eta}{S_2^\xi \xi + S_2^\eta \eta} \right)
\]

(38)

where \(\xi_{eq}(\xi_{neq}, \bar{\xi})\) is defined as the resolution function of

\[
(S_{eq}^1)^t \ln (S_{eq}^1 \xi_{eq} + S_{eq}^1' \xi_{eq} + S_1^T \eta) = K.
\]

(39)
(39) corresponds to \( f_1 = 0 \) and (39) to \( f_2 = 0 \) in Appendix A. Note that, since the immobile species do not occur in the mass action equilibrium reactions, \( \xi_{eq} \) is in fact independent of \( \bar{\xi} \).

(ii) (b) We want to solve simultaneously the first and the last block of (33) for \( \xi_{eq}, \bar{\xi} \) as functions of \( \xi_{neq} \). The resulting system is

\[
\frac{d}{dt} \xi_{neq} + L \xi_{neq} = A_{22} R_{neq} \left( \frac{S_{eq}^1 \xi_{eq} + S_{neq}^1 \xi_{neq} + S_1^T \eta}{S_2^2 \xi_{eq} + S_2^T \bar{\eta}} \right) \tag{40}
\]

where \( \bar{\xi}(\xi_{neq}), \xi_{eq}(\xi_{neq}) \) are defined as the resolution functions of

\[
\frac{1}{\Delta t} (\bar{\xi} - \bar{\xi}_{old}) = A_2 R_{neq} \left( \frac{S_{eq}^1 \xi_{eq} + S_{neq}^1 \xi_{neq} + S_1^T \eta}{S_2^2 \xi + S_2^T \bar{\eta}} \right) \tag{41}
\]

The block \( f_1 = 0 \) corresponds to (40) and is of size \( J_1 - J_{eq} \), which is, if \( J_{eq} > 0 \), smaller than the size of the PDE block of (11) and (14). By definition of \( J_1 \), \( J_1 \leq J_{eq} + J_{neq} \) and therefore the number of coupled PDEs is less or equal to the number of nonequilibrium reactions:

\[ 0 \leq J_1 - J_{eq} \leq J_{neq} \]

One can verify that the resolution equations \( \bar{\xi}(\xi) \) for system (37), \( \xi_{eq}(\xi_{neq}) \) for system (39) and \( \xi_{eq}(\xi_{neq}), \bar{\xi}(\xi_{neq}) \) for system (41) exist:

**Theorem 2** Suppose that there is a positive lower bound for all concentrations \( c_i, \bar{c}_i \), uniform with respect to \( x \).

(i) For (37) there is a local resolution equation \( \bar{\xi} = \bar{\xi}(\xi) \) if \( \Delta t \) is small enough.

(ii) Suppose that all equilibrium reaction are governed by mass action, and the condition \( S_{eq}^2 = 0 \) of section 4.3 holds, i.e., all equilibrium reactions are homogeneous reactions in the mobile phase.

(a) Then for (39) there is always a local resolution equation for \( \xi_{eq} = \xi_{eq}(\xi_{neq}) \).

(b) Then for (41) there is a local resolution equation for \( \xi_{eq}(\xi_{neq}), \bar{\xi}(\xi_{neq}) \), if \( \Delta t \) is small enough.

**Proof:** See Appendix B.

### 4.3.3 Exact or approximate Jacobian?

In this section the Jacobian of the nonlinear problems of section 4.3.2 is considered. For the sake of brevity we focus on system (36):

The Jacobian of the nonlinear block reads (after time discretization)

\[
J = I + \Delta t L - \Delta t A_1 \left( \frac{\partial R}{\partial c} S^1 + \frac{\partial R}{\partial c} S^2 \frac{\partial \bar{\xi}}{\partial \xi} \right) \tag{42}
\]
In order to eliminate the term \( \frac{\partial \bar{\xi}}{\partial \xi} \), we differentiate (37) with respect to \( \xi \):

\[
\frac{\partial \bar{\xi}}{\partial \xi} = \Delta t A_2 \left( \frac{\partial R}{\partial c} S_1^* + \frac{\partial R}{\partial \bar{c}} S_2^* \frac{\partial \bar{\xi}}{\partial \xi} \right),
\]

hence

\[
\frac{\partial \bar{\xi}}{\partial \xi} = \left( I - \Delta t A_2 \frac{\partial R}{\partial \bar{c}} S_2^* \right)^{-1} \Delta t A_2 \frac{\partial R}{\partial c} S_1^*.
\]

The exact Jacobian (42) with (43) requires the explicit computation of the inverse of \( I - \Delta t A_2 \frac{\partial R}{\partial \bar{c}} S_2^* \), which is a block matrix of size \( J_2 \times J_2 \) at each grid point of the computational domain.

To avoid this computation, one can approximate the Jacobian \( J \) by omitting all terms that are (A) of order \( \Delta t^2 \) or (B) of order \( \Delta t^3 \). In case (A) we get the approximate Jacobian

\[
\tilde{J}_1 = I + \Delta t L - \Delta t A_1 \frac{\partial R}{\partial c} S_1^*,
\]

in case (B) we get

\[
\tilde{J}_2 = I + \Delta t L - \Delta t A_1 \left( \frac{\partial R}{\partial c} S_1^* + \Delta t \frac{\partial R}{\partial \bar{c}} A_2 \frac{\partial R}{\partial \bar{c}} S_2^* \right).
\]

The so-called inexact Newton method using the approximate Jacobian increases the overall efficiency in the numerical example of section 6. Note that, for larger reaction networks, all the matrices in (44), (45) are sparse. The sparsity should be used for the implementation of the matrix-matrix multiplications.

If we apply the simplification of type (A) to the system (40)-(41), we get the approximate Jacobian

\[
\tilde{J}_3 = I + \Delta t L - \Delta t A_{22} \frac{\partial R_{\text{neq}}}{\partial c} \left[ I - S_{eq}^1 (S_{eq}^1 t D_c S_{eq}^1)^{-1} S_{eq}^1 t D_c \right] S_{neq}^1,
\]

where \( D_c \) is the \( I \times I \)-matrix diagonal matrix with entries \( \frac{1}{c_i} \). Depending on the present species concentrations in \( D_c \), further simplifications might be possible.

5 The Monod model

For the simulation of biodegradation of pollutants, the Monod model (also called Michaelis–Menten kinetics) turned out to be an efficient tool. In this section we show how to apply the reduction mechanism of section 4 to the Monod model. We consider two examples of different complexity:

**Example 2: The binary Monod model for one biodegradation reaction**

The binary Monod model describes the reaction of an electron donator \( D \) and an electron acceptor \( A \) in the presence of a biospecies \( X \). The reaction is usually considered to be ‘slow’ (kinetic). The model equations are

\[
\begin{align*}
\dot{c}_A + L c_A &= -\frac{R(c_A, c_D, c_X)}{c_X}, \\
\dot{c}_D + L c_D &= -\frac{\alpha_D/A}{R(c_A, c_D, c_X)} R(c_A, c_D, c_X), \\
\dot{c}_X &= \gamma (1 - \frac{c_X}{c_{X,max}}) R(c_A, c_D, c_X)
\end{align*}
\]

(46)
with the reaction rate

\[ R(c_A, c_D, c_X) = \alpha_A \frac{c_A}{K_A + c_A} \frac{c_D}{K_D + c_D} c_X, \]

(47)

\( \alpha_A, \alpha_D/A, \gamma, c_{X,\text{max}} > 0 \). The factor \( \gamma (1 - \frac{c_X}{c_{X,\text{max}}}) \) limits the concentration of the biospecies by \( c_{X,\text{max}} \) and can be justified by limiting factors such as a limited pore space or toxic metabolism products.

Note that unfortunately this system is of different type than our model equations (2), since the 'stoichiometric' coefficient \( \gamma (1 - \frac{c_X}{c_{X,\text{max}}}) \) depends on the concentration \( c_X \). The independence of the stoichiometric coefficients from the concentrations (and from the location) was required in the step from (19) to (23). However, it is possible to transform system (46)-(47) into the shape of (2):

Instead of one reaction \( R \) we introduce the two reactions

\[ R_1 := R, \quad R_2 := (1 - \frac{c_X}{c_{X,\text{max}}}) R. \]

Now the system can be written as

\[
\begin{align*}
\dot{c}_A + L c_A &= -R_1(c_A, c_D, c_X) \\
\dot{c}_D + L c_D &= -\alpha_D/A R_1(c_A, c_D, c_X) \\
\dot{c}_X &= \gamma R_2(c_A, c_D, c_X)
\end{align*}
\]

The stoichiometric matrix for this system,

\[ S = S_{\text{neq}} = \left( \begin{array}{c} S_1 \\ S_2 \end{array} \right) = \left( \begin{array}{cc} -1 & 0 \\ -\alpha_D/A & 0 \\ 0 & \gamma \end{array} \right), \]

is independent of the concentration vector. To this system, the reduction mechanism of section 4 can be applied. It is \( S_1^* = (-1, -\alpha_A/D)^t \), \( S_2^* = (\gamma) \), and we choose \( S_1^T = (-\alpha_D/A, 1)^t \). We obtain

\[
\begin{align*}
\left( \begin{array}{c} \frac{d}{dt} + L \\ \frac{d}{dt} + L \end{array} \right) \eta &= 0 \\
\left( \begin{array}{c} \frac{d}{dt} + L \\ \frac{d}{dt} + L \end{array} \right) \xi &= R_1(c_A, c_D, c_X) \\
\frac{d}{dt} \xi &= R_2(c_A, c_D, c_X)
\end{align*}
\]

(48)

with the 3 scalar unknowns \( \eta, \xi, \bar{\xi} \) and \( c_A = -\xi - \alpha_D/A \eta, \quad c_D = -\alpha_D/A \xi + \eta, \quad c_X = \gamma \bar{\xi} \). After discretization in time, the third equation of (48) is solved for \( \bar{\xi} \), and \( \bar{\xi} \) is eliminated in the right-hand side of the second equation. The reduction mechanism yields a completely decoupled set of one linear equation for \( \eta \) and one nonlinear equation for \( \xi \) remaining.

**Example 3: A biodegradation chain**

Let us consider a chain of 2 biodegradation reactions with two biospecies \( \bar{X}_1, \bar{X}_2 \), where the product \( C \) of the first reaction serves as an educt for the second.

\[
\begin{align*}
A + B &\rightarrow \bar{X}_1 \rightarrow C \\
C + D &\rightarrow \bar{X}_2 \rightarrow E
\end{align*}
\]
The corresponding coupled system consists of 5 PDEs and 2 ODEs:

\[
\begin{align*}
\dot{c}_A + Lc_A &= -R_1(c_A, c_B, c_{X_1}) \\
\dot{c}_B + Lc_B &= -\alpha_{B/A}R_1(c_A, c_B, c_{X_1}) \\
\dot{c}_C + Lc_C &= \beta_1 R_1(c_A, c_B, c_{X_1}) - R_2(c_C, c_D, c_{X_2}) \\
\dot{c}_D + Lc_D &= -\alpha_{D/C}R_2(c_C, c_D, c_{X_2}) \\
\dot{c}_E + Lc_E &= \beta_2 R_2(c_C, c_D, c_{X_2}) \\
\dot{c}_{X_1} &= \gamma_1 (1 - \frac{c_{X_1}}{c_{X_{1,\text{max}}}})R_1(c_A, c_B, c_{X_1}) \\
\dot{c}_{X_2} &= \gamma_2 (1 - \frac{c_{X_2}}{c_{X_{2,\text{max}}}})R_2(c_C, c_D, c_{X_2}),
\end{align*}
\]

(49)

\[
R_1(c_A, c_B, c_{X_1}) = \alpha_A \frac{c_A}{K_A + c_A} \frac{c_B}{K_B + c_B} c_{X_1},
\]

\[
R_2(c_C, c_D, c_{X_2}) = \alpha_C \frac{c_C}{K_C + c_C} \frac{c_D}{K_D + c_D} c_{X_2}.
\]

After introduction of

\[
R_3 := (1 - \frac{c_{X_1}}{c_{X_{1,\text{max}}}})R_1, \quad R_4 := (1 - \frac{c_{X_2}}{c_{X_{2,\text{max}}}})R_2,
\]

the stoichiometric matrix reads

\[
S = \begin{pmatrix}
S_1 \\
S_2
\end{pmatrix} = \begin{pmatrix}
-1 & 0 & 0 & 0 \\
-\alpha_{B/A} & 0 & 0 & 0 \\
+\beta_1 & -1 & 0 & 0 \\
0 & -\alpha_{D/C} & 0 & 0 \\
0 & +\beta_2 & 0 & 0 \\
0 & 0 & \gamma_1 & 0 \\
0 & 0 & 0 & \gamma_2
\end{pmatrix}
\]

The column rank of $S_1$ and $S_2$ is 2, the orthogonal complement matrix $S_1^\top$ has column rank $5 - 2 = 3$. Hence, the reduced system

\[
\begin{align*}
\frac{d}{dt} + L \eta_1 &= 0 \\
\frac{d}{dt} + L \eta_2 &= 0 \\
\frac{d}{dt} + L \eta_3 &= 0 \\
\frac{d}{dt} + L \xi_1 &= R_1 \\
\frac{d}{dt} + L \xi_2 &= R_2 \\
\frac{d}{dt} \xi_1 &= R_3 \\
\frac{d}{dt} \xi_2 &= R_4
\end{align*}
\]

consists, after elimination of the equations for $\xi_1, \xi_2$, of 3 linear decoupled equations for $\eta_1, \eta_2, \eta_3$ and 2 nonlinear coupled equations for $\xi_1, \xi_2$.

Friedly’s method for this problem from section 3.1 yields $J_1 + J = 2 + 4 = 6$ coupled nonlinear equations and $I - J_1 = 5 - 2 = 3$ linear decoupled PDEs. Combined with the substitution approach of section 3.3, $J_1 = 2$ of the nonlinear equations remain.
6 Numerical results

In this section we show by a numerical example that the reduction algorithm is applicable. We compare the cpu time spent for the reduced and for the non-reduced problem and analyse different simplifications of the algorithm in regard of cpu time reduction.

6.1 The test problem

Numerical test runs were performed for a one-dimensional problem with the program Richy\textsuperscript{2} on the domain $\Omega = (0, 20)$. The chemical reactions are

\[
\begin{align*}
2A & \xrightarrow{R_1} B \\
B + C & \xrightarrow{R_2} D \\
B + D & \xrightarrow{R_3} E
\end{align*}
\]

where species $A, B, C, D$ are mobile and species $E$ is sorbed to the soil matrix. Reactions $R_1, R_2$ were considered as equilibrium reactions with equilibrium constants $k_{f1}/k_{b1} = 1.0, k_{f2}/k_{b2} = 1.0$, and $R_3$ as a kinetic reaction with the rate parameters $k_{f3} = 100, k_{b3} = 1$. We consider a constant water flux $q = 0.2$, a porosity $\theta = 0.33$ and a dispersion-diffusion $D = 0.39$. Nonconstant initial concentrations for species $A-E$ are chosen.

The application of the algorithm of section 4 yields a system of 1 linear scalar decoupled equation for $\eta$ and 1 nonlinear scalar equation for $\xi_{neq}$ and 3 local equations which are implicitly solved for $\xi_{eq1}, \xi_{eq2}, \bar{\xi}$:

\[
\begin{align*}
(1) \quad \dot{\eta} + L\eta & = 0 \\
(2) \quad \xi_{neq} + L\xi_{neq} & = R_3(\xi_{neq}, \xi_{eq1}, \xi_{eq2}, \bar{\xi}) \\
(2a) \quad R_2(\xi_{neq}, \xi_{eq1}, \xi_{eq2}, \bar{\xi}) & = 0 \\
(2b) \quad R_3(\xi_{neq}, \xi_{eq1}, \xi_{eq2}, \bar{\xi}) & = 0 \\
(2c) \quad \dot{\bar{\xi}} & = R_3(\xi_{neq}, \xi_{eq1}, \xi_{eq2}, \bar{\xi})
\end{align*}
\]

So the algorithm yields a complete decoupling with one nonlinear scalar PDE remaining. Let us compare this to the algorithm of section 3: The coupled last two blocks of (8) consist of $J + J_1 = 3 + 3 = 6$ equations. In formulation (9)-(11), $J_1 = 3$ coupled nonlinear PDEs are left. The version of section 3.3 is not available since condition (15) is not met.

We use 500 elements and 50 timesteps with $\Delta t = 10^{-3}$ for our simulation.

6.2 Comparison of the decoupled problem to the original problem

We present the computational time (in seconds) for the reduced system (50) in comparison to the primordial system for $A, B, C, D, E$.

For the reduced system the cpu time and the average number of Newton steps is listed in Table 1. The cpu time used for assembling and for the linear solver for the

\footnote{\textsuperscript{2}cf. http://www.am.uni-erlangen.de/am1/software/RichyDocumentation}
6.2 Comparison of the decoupled problem to the original problem

Table 1: Computational time, and average number of Newton steps per timestep for the reduced problem

<table>
<thead>
<tr>
<th></th>
<th>$\eta$</th>
<th>$\xi_{neq}$</th>
<th>$\xi_{eq}, \bar{\xi}$</th>
<th>tot. cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu for assembl.</td>
<td>0.80</td>
<td>2.86</td>
<td>-</td>
<td>3.66</td>
</tr>
<tr>
<td>cpu for lin. solver</td>
<td>0.80</td>
<td>2.21</td>
<td>-</td>
<td>3.01</td>
</tr>
<tr>
<td>cpu for loc. probl.</td>
<td>-</td>
<td>-</td>
<td>5.46</td>
<td>5.46</td>
</tr>
<tr>
<td>tot. cpu</td>
<td>1.60</td>
<td>5.07</td>
<td>5.46</td>
<td>12.13</td>
</tr>
<tr>
<td>Newton steps</td>
<td>1.00</td>
<td>2.02</td>
<td>2.62</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Computational time and number of Newton steps for the original problem, in dependence of the rate coefficient for the fast reactions $R_1, R_2$.

<table>
<thead>
<tr>
<th>rate coeff.</th>
<th>$10^2$</th>
<th>$10^3$</th>
<th>$10^4$</th>
<th>$10^5$</th>
<th>$3\cdot10^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu for assembl.</td>
<td>6.41</td>
<td>6.36</td>
<td>6.43</td>
<td>13.42</td>
<td>49</td>
</tr>
<tr>
<td>cpu for lin. solver</td>
<td>22.02</td>
<td><strong>40.87</strong></td>
<td>111.06</td>
<td>483.39</td>
<td>2041</td>
</tr>
<tr>
<td>total cpu</td>
<td>28.43</td>
<td><strong>47.41</strong></td>
<td>117.49</td>
<td>496.81</td>
<td>2090</td>
</tr>
<tr>
<td>Newton steps</td>
<td>2.40</td>
<td><strong>2.40</strong></td>
<td>2.44</td>
<td>3.58</td>
<td>7.82</td>
</tr>
</tbody>
</table>

For the $\eta$-equation and for the $\xi_{neq}$-equation is displayed. The average number of Newton steps per timestep is 1.00 for the $\eta$-problem, since it is linear, and 2.02 for the nonlinear $\xi_{neq}$-problem. The nested local Newton iterations to recover the values for $\xi_{eq}, \bar{\xi}$ require 5.46 seconds cpu time. The average number of Newton steps for the local problems (per grid point and timestep and global Newton step) is 2.62. As long as the global Newton iterate has a large residual, it seems unnecessary to solve the local problems with full accuracy. So in order to use the cpu time efficiently, the convergence criterion of the nested local Newton iteration (i.e., the bound for the residual for the local problem) is chosen in dependence of the residual of the present iterate of the global Newton sequence.

Let us consider the computational results for the nonreduced formulation of the problem. For the nonreduced formulation consisting of 5 coupled nonlinear PDEs and ODEs, the equilibrium reactions have to be implemented as 'fast' kinetic reactions. Table 2 shows the cpu time for kinetic rates $k_f, k_b$ varying from $10^2$ to $3\cdot10^5$. Since the rate $R_2$ is nonlinear, the magnitude of the kinetic rate influences the nonlinearity of the problem. Note the strong increase of the number of Newton steps and of the cpu time if the rates are high. Numerical tests showed that for $k_f = k_b \approx 10^3$ the error caused by the finite reaction rate is not bigger than the temporal discretization error. Hence, if the true rate of the fast reactions is a priori not known it seems reasonable to choose $k_f = k_b \approx 10^3$. Even if these rates are known and larger than $10^3$, it could be considered to set the rates to $\approx 10^3$ to reduce the number of Newton steps, since the effect on the accuracy of the result is negligible.

Comparing Table 1 with Table 2 we see that the total cpu time was reduced from 47.41 (for the reaction rate $10^3$) to 12.13 seconds, which is approximately a
factor of 4. The cpu time for the solution of the global linear systems of equations (without assembling time) was even reduced from 40.87 to 3.01 seconds.

Let us remark on the choice of the linear solver on the speed-up factor of the reduction algorithm: For the computational test we have chosen the GMRES(k) solver with SSOR relaxation, although for 1d problems, a direct band-LU solver is usually faster. This choice was made in order to get results that are more representative also for 2d/3d computations: If a 1d problem is solved with a very fast direct band matrix solver, the cpu time for the solver is rather small compared to the cpu time for assembling the system matrix, no matter if a reduction algorithm is used or not. For 2d and 3d problems, direct solvers cannot compete with efficient iterative solvers.

6.3 Further reduction of cpu time, inexact Newton

The reduction algorithm offers several possibilities for the acceleration of the computation, for example

1. The simplification of the Jacobian matrix for the global problem:

   The terms $\partial \bar{\xi}/\partial \xi_{neq}$ (and/or $\partial \xi_{neq}/\partial \xi_{eq}$) could be omitted in order to reduce the cpu time for assembling the matrix of the global problem. The cancelling of the term $\partial \bar{\xi}/\partial \xi_{neq}$ in the Jacobian means to ignore the terms $O(\Delta t^2)$ (cf. section 4.3.3). It is desirable to check if and how much the number of Newton steps may increase by this procedure since the Newton method is replaced by an inexact Newton method then.

2. The simplification of the Newton step for the nested local problems at each grid point:

   Instead of solving the $3 \times 3$ nonlinear local problem for $\xi_{eq}, \bar{\xi}$, one can try to apply Newton to the $2 \times 2$-block of equilibrium conditions for $\xi_{eq}$ with fixed $\bar{\xi}$ and after that to apply Newton to the scalar ODE for $\xi$ with fixed $\xi_{eq}$. This sequential approach has the advantage that the problems for $\xi_{eq}$ and for $\bar{\xi}$ are of smaller size than the coupled problem and are faster to solve.

3. Another simplification of the solution process of the local problems is to iterate between one Newton step for $\xi_{eq}$ with fixed $\bar{\xi}$ and one Newton step for $\bar{\xi}$ with fixed $\xi_{eq}$. As in 2., the problems to solve are smaller than for the coupled local problems, and there should be no effect on the performance of the global Newton iteration.

   Note that all the simplification methods 1.-3. can only influence the computational time and the number of Newton steps, but they cannot deteriorate the accuracy of the solution.

The simplification method 1 is investigated in Table 3. If both the terms for $\xi_{eq}$ and for $\bar{\xi}$ are omitted in the Jacobian of the global Newton, the number of Newton steps increases from 2.02 to 3.28, and the total cpu time increases from 12.13 to 17.28 (right part of the table). If, instead, only the $\bar{\xi}$-term, i.e., only the $O(\Delta t^2)$-term is omitted (as proposed in section 4.3.3), the number of Newton steps is still 2.02, and the total cpu time decreases a bit to 11.73 (middle part of
Table 3: Reduction method 1: Computational time and number of Newton steps for the decoupled problem for the full global Jacobian, for the global Jacobian with the $\xi$-terms omitted, and for the global Jacobian with both $\xi_{eq}$-, $\xi$-terms omitted.

<table>
<thead>
<tr>
<th></th>
<th>exact Jac.</th>
<th>$\xi$-terms omitted</th>
<th>$\xi_{eq}$-, $\xi$-terms omitted</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\eta$ $\xi, \bar{\xi}$ tot. cpu</td>
<td>$\eta$ $\xi, \bar{\xi}$ tot. cpu</td>
<td>$\eta$ $\xi, \bar{\xi}$ tot. cpu</td>
</tr>
<tr>
<td>cpu for ass.</td>
<td>0.80 2.86 3.66</td>
<td>0.75 2.43 3.18</td>
<td>0.79 2.72 3.51</td>
</tr>
<tr>
<td>cpu for lin. solver</td>
<td>0.80 2.21 3.01</td>
<td>0.80 2.24 3.04</td>
<td>0.84 3.68 4.52</td>
</tr>
<tr>
<td>cpu for loc. probl.</td>
<td>- 5.46 5.46</td>
<td>- 5.51 5.51</td>
<td>- 9.25 9.25</td>
</tr>
<tr>
<td>tot. cpu</td>
<td>1.60 10.53 12.13</td>
<td>1.55 10.18 11.73</td>
<td>1.63 15.65 17.28</td>
</tr>
<tr>
<td>glob. N. steps</td>
<td>1.00 2.02 -</td>
<td>1.00 2.02 -</td>
<td>1.00 3.28 -</td>
</tr>
<tr>
<td>loc. N. steps</td>
<td>- 2.62 -</td>
<td>- 2.62 -</td>
<td>- 2.77 -</td>
</tr>
</tbody>
</table>

Table 4: Reduction methods 2 and 3: Computational time and number of Newton steps for the decoupled problem with inexact Jacobian matrix for the local problems (columns 2 and 3) compared to the exact handling of Jacobian the local problems (column 1).

<table>
<thead>
<tr>
<th></th>
<th>coupled</th>
<th>sequential</th>
<th>alternating</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\eta$ $\xi, \bar{\xi}$ tot. cpu</td>
<td>$\eta$ $\xi, \bar{\xi}$ tot. cpu</td>
<td>$\eta$ $\xi, \bar{\xi}$ tot. cpu</td>
</tr>
<tr>
<td>cpu for ass.</td>
<td>0.80 2.86 3.66</td>
<td>0.81 2.88 3.69</td>
<td>0.82 2.90 3.72</td>
</tr>
<tr>
<td>cpu for lin. solver</td>
<td>0.80 2.21 3.01</td>
<td>0.83 2.22 3.05</td>
<td>0.82 2.20 3.02</td>
</tr>
<tr>
<td>cpu for loc. probl.</td>
<td>- 5.46 5.46</td>
<td>- 3.32 3.32</td>
<td>- 3.43 3.43</td>
</tr>
<tr>
<td>tot. cpu</td>
<td>1.60 10.53 12.13</td>
<td>1.64 8.42 10.06</td>
<td>1.64 8.53 10.17</td>
</tr>
<tr>
<td>glob. N. steps</td>
<td>1.00 2.02 -</td>
<td>1.00 2.02 -</td>
<td>1.00 2.02 -</td>
</tr>
<tr>
<td>loc. N. steps</td>
<td>- 2.62 -</td>
<td>- 3.75 -</td>
<td>- 4.01 -</td>
</tr>
</tbody>
</table>

the table) thanks to the simpler assembling of the Jacobian. A closer look at the Jacobian shows that indeed the dependence of the entries on $\bar{\xi}$ is rather small since the kinetic reactions are well resolved by the timestepping, but the dependence on $\xi_{eq}$ is not negligible. Of course, the decision if further entries can be omitted depends on the problem.

Table 5: Reduction method 1 and 2 combined: Computational time and number of Newton steps.

<table>
<thead>
<tr>
<th></th>
<th>$\xi$-terms omitted in glob. Jac.; loc. probl. sequential</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\eta$ $\bar{\xi}$ tot. cpu</td>
</tr>
<tr>
<td>Ass.</td>
<td>0.77 2.39 3.16</td>
</tr>
<tr>
<td>Lin. solver</td>
<td>0.81 2.23 3.04</td>
</tr>
<tr>
<td>Local. probl.</td>
<td>- 3.30 3.30</td>
</tr>
<tr>
<td>tot. cpu</td>
<td>1.58 7.92 9.50</td>
</tr>
<tr>
<td>glob. N. steps</td>
<td>1.00 2.02 -</td>
</tr>
<tr>
<td>loc. N. steps</td>
<td>- 3.75 -</td>
</tr>
</tbody>
</table>
Table 4 shows the speed-up for the inexact handling of the local Newton iteration due to method 2 and method 3. Especially the splitting of the local problems by method 2 ('sequential') shows a reduction of the cpu time by 17%.

Finally, Table 5 shows the combination of the simplification of the global Jacobian by omitting the $\bar{\xi}$-terms with the sequential handling of the local problems for $\xi_{eq}, \bar{\xi}$. The gain is 22% in terms of cpu time: The total computational time is 9.50, while the exact Newton for the reduced systems requires 12.13, and the nonreduced system requires 47.41.

7 Summary and Outlook

We have proposed a new reduction scheme for multicomponent reactive transport problems, which is able to handle coupled mobile and immobile species and mixed equilibrium and nonequilibrium reactions. A special property of the scheme is that it allows to eliminate local equations resulting from equilibrium or nonequilibrium reactions, and that this elimination process does not lead to nonlinear coupling terms under the transport operator, which would cause many nonzero entries in the Jacobian. Under rather general conditions, assuming mass action for the equilibrium reactions, the applicability of the reduction method and the existence of the resolution functions can be guaranteed.

The number of coupled nonlinear PDEs of the reduced scheme is smaller than the number of given species, and also usually smaller than the number of reactions: It is always less or equal to the number of those nonequilibrium reactions, in which mobile species participate.

In the present formulation, the method is restricted to reaction networks with homogeneous equilibrium reactions and arbitrary kinetic reactions. The generalization to arbitrary reaction networks containing heterogeneous equilibrium reactions would be desirable.

The presented 1d numerical simulation of a moderate-sized reaction network shows that the gain of efficiency by the reduced size of the system outweighs the additional costs of the more complex matrix assembling and the nested local iterations, at least in the example under consideration. Since linear solvers for 2d and 3d problems differ from 1d solvers, it is not obvious that the efficiency results of the 1d computations of section 6 apply in the same way for 2d/3d problems. It is possible that the gain of efficiency is even larger for those problems. The impact of the more complex matrix assembling and the local iterations on the total computational time should be investigated carefully, especially for larger reaction networks. It is planned to implement the reduction method for 2d problems and to compare the efficiency with the global implicit method for the original non-reduced set of equations. An implementation of the method on parallel computers should be considered, since the reduction algorithm reduces the size of the PDE system (which is hard to distribute on several processors) and introduces local problems (which are easily distributed among several processors). So on parallel machines, the application of the reduction scheme might gain even more efficiency.
Appendix A: Reduction by Using Implicit Elimination

The systems of equations derived in the sections 3.1, 3.2, 4.1, 4.3.1 can be further reduced to smaller systems if the involved AEs and/or ODEs (after a time-discretization) are solved for certain variables and these variables are eliminated in the remaining PDEs. The Newton iteration is only performed for this reduced set of unknowns/equations.

It is sometimes believed that this procedure can only be performed if the resolution of the equations can be done *explicitely*. Since the AEs are in general nonlinear, they often cannot be resolved explicitly for the ‘new’ variables $\xi_i, \bar{\xi}_i$. Nevertheless, it is possible to show the existence of a local resolution equation, even if this equation is in general not known explicitly. The existence of the resolution equation and the possibility to compute the partial derivatives of the nonlinear equations is enough to perform the Newton step for the (now smaller) system of PDEs. In the remainder of this section, we lay out the general idea of such a reduced Newton method. The application of this method to the systems of section 3 and of section 4 is pointed out in the sections 3.3 and 4.3.2 (see theorem 2). In these sections, special emphasis is laid on the observation that under rather general assumptions the existence of the resolution functions is guaranteed.

All the methods of section 3 and of 4 have in common that, after a time discretization, a timestep for the coupled nonlinear system of equation consists of a block of PDEs and a block of local AEs. We describe such a system by

\[
\begin{align*}
    f_1(u, v) &= 0 \\
    f_2(u, v) &= 0
\end{align*}
\]  

where $f_1 = 0$ is the block of PDEs (coupling the unknowns at all grid points of the domain) and $f_2 = 0$ is the block of local AEs. $u$ and $v$ are vectors of unknowns where the number of degrees of freedom of vector $u$ ($v$, resp.) coincides with the number of equations in $f_1 = 0$ ($f_2 = 0$, resp.).

A straightforward Newton step $(u_n, v_n) \rightarrow (u_{n+1}, v_{n+1})$ for (51) would read

\[
\begin{align*}
    \frac{\partial f_1}{\partial u} \bigg|_{(u_n, v_n)} \Delta u_n + \frac{\partial f_1}{\partial v} \bigg|_{(u_n, v_n)} \Delta v_n &= f_1(u_n, v_n), \\
    \frac{\partial f_2}{\partial u} \bigg|_{(u_n, v_n)} \Delta u_n + \frac{\partial f_2}{\partial v} \bigg|_{(u_n, v_n)} \Delta v_n &= f_2(u_n, v_n),
\end{align*}
\]

\[
\begin{align*}
    u_{n+1} &= u_n \Delta u_n, \\
    v_{n+1} &= v_n \Delta v_n.
\end{align*}
\]

If there is a resolution function $v = v(u)$ for equation $f_2(u, v) = 0$, then system (51) can be written in the equivalent form

\[
f_1(u, v(u)) = 0,
\]

for which the Jacobian reads

\[
J = \frac{\partial f_1}{\partial u} + \frac{\partial f_1}{\partial v} \frac{\partial v}{\partial u} = \frac{\partial f_1}{\partial u} - \frac{\partial f_1}{\partial v} \left( \frac{\partial f_2}{\partial v} \right)^{-1} \frac{\partial f_2}{\partial u}.
\]
and a Newton step reads
\[ J \Delta u_n = f_1(u_n, v_n) \]
\[ u_{n+1} = u_n - \Delta u_n. \]  \hspace{1cm} (57)

Note that the computation of the inverse matrix of \( \partial f_2/\partial v \) has to be done explicitly in order to assemble the Jacobian. Here it is essential that due to the lack of spatial derivatives in the ODEs/AEs, problem \( f_2(u, v) = 0 \) consists of equations at each node of the computational domain without any coupling between the nodes. So \( \partial f_2/\partial v \) has a diagonal block structure with one block per node which enables us to perform the explicit inversion of the matrix. To save cpu time, an approximate Jacobian may be considered (cf. section 4.3.3) instead of \( J \).

After \( u_{n+1} \) is computed, \( v_{n+1} \) is required before the next iteration step (57) can be done. If the resolution function \( v(u) \) is explicitly known, the value \( v_{n+1} \) can be computed by
\[ v_{n+1} = v(u_{n+1}). \]  \hspace{1cm} (58)

Both Newton iterations (52)-(55) and (57)-(58) are not equivalent in the sense that they produce the same sequence \( (u_n, v_n) \) for the same starting point \( (u_0, v_0) \). The fixed points of both series are the same.

In the general setting which is considered throughout this article, we cannot assume that the resolution function is explicitly known, since this would require very simple, mostly linear reaction rates. In the general case of a not explicitly known resolution function, the following method to recover \( v_{n+1} \) is proposed:

Solve
\[ f_2(u_{n+1}, v) = 0 \]  \hspace{1cm} (59)
for \( v \) by a (nested) Newton iteration. The system matrix \( \partial f_2/\partial v \) of this nested iteration has again block structure, i.e., the solution of (59) decouples into small 'local' nonlinear problems. The sequence \( (u_{n+1}, v_{n+1}) \) then is the same as for the Newton iteration (57)-(58) for problem (56).

To save cpu time, another approach could be the following: After the Newton step (57) for \( u_{n+1} \) is performed, \( v_{n+1} \) is computed through (53), (55). Note that this procedure recovers neither the Newton sequence of (52)-(55) nor the Newton sequence of (57)-(58). However, the computational costs per step are lower than for the previous procedure.

If for some reason it is desired to recover the Newton sequence (52)-(55), one can proceed as follows: Eliminate \( \Delta v_n \) in (52) by solving formally the local problems (53) for \( \Delta u_n \). Solve numerically the resulting global problem for \( \Delta u_n \). With \( \Delta u_n \) known, solve numerically the local problems (53) for \( \Delta v_n \).

**Appendix B: Proofs**

**Proof of (21), (22):**

(21) is an equation in \( R^I \), and since the columns of the block matrix \( (S_1^* \mid S_1^\top) \) form a basis of \( R^I \), it is sufficient to prove that
\[ S_1^* c = S_1^t(S_1^* \xi + S_1^\top \eta), \quad S_1^\top c = S_1^t(S_1^* \xi + S_1^\top \eta) \]  \hspace{1cm} (60)
hold. This is done by using the definition (20) of $\xi, \eta$ and the orthogonality relations (17) on the right-hand sides of (60).

Similarly to (21), the equality (22) is proved. □

**Proof of theorem 1.**

(i) Following (3)-(6) and using (12), equation (9) is equivalent to

$$Q_{eq}(\xi_{eq}, \xi_{neq}, \xi_{m}) := S_{eq}^t \ln \left( \frac{c}{\bar{c}} \right)$$

$$= S_{eq}^t \ln \left( S_{eq}\xi_{eq} + S_{neq}\xi_{neq} + S^\top \eta + \left( S^*_m \xi_{m} + S^\top_1 \eta_{m} \right) \right) = K.$$

We have to check that the matrix $\frac{\partial Q_{eq}}{\partial \xi_{eq}}$ is regular: We get

$$\frac{\partial Q_{eq}}{\partial \xi_{eq}} = S_{eq}^t DS_{eq},$$

where $D$ is the $(I+\bar{I}) \times (I+\bar{I})$ diagonal matrix with diagonal entries $1/c_i$, $1/\bar{c}_i$.

Since the columns of $S_{eq}$ were supposed to be linearly independent, for all $\xi \neq 0$, $S_{eq}\xi \neq 0$ holds. Since all diagonal entries of $D$ are positive,

$$\langle S_{eq}^t DS_{eq}\xi, \xi \rangle = \langle DS_{eq}\xi, S_{eq}\xi \rangle > 0$$

follows. Hence, $S_{eq}^t DS_{eq}$ is a symmetric positive definite matrix and invertible.

(ii) A temporal discretization of (10) with the implicit Euler methods yields

$$Q_{neq}(\xi_{eq}, \xi_{neq}, \xi_{m}) := \xi_{neq} - \xi_{neq}^{old}$$

$$-\Delta t R_{neq} \left( S_{eq}\xi_{eq} + S_{neq}\xi_{neq} + S^\top \eta + \left( S^*_m \xi_{m} + S^\top_1 \eta_{m} \right) \right) = 0$$

$$Q_{neq}(\xi_{eq}, \xi_{neq}, \xi_{m}) :=$$

$$\xi_{neq} - \xi_{neq}^{old} - \Delta t R_{neq} \left( S_{eq}\xi_{eq} + S_{neq}\xi_{neq} + S^\top \eta + \left( S^*_m \xi_{m} + S^\top_1 \eta_{m} \right) \right) = 0$$

where $\xi_{neq}^{old}$ is the numerical solution at the old time level. We obtain

$$\frac{\partial Q_{neq}}{\partial \xi_{neq}} = I - \Delta t \frac{\partial R_{neq}}{\partial (c, \bar{c})} S_{neq}.$$

This matrix is regular at least if

$$\Delta t \left\| \frac{\partial R_{neq}}{\partial (c, \bar{c})} S_{neq} \right\| < 1,$$

i.e., if

$$\Delta t < \left\| \frac{\partial R_{neq}}{\partial (c, \bar{c})} S_{neq} \right\|^{-1}$$
in the whole computational domain, where \( \| \cdot \| \) is an arbitrary submultiplicative matrix norm, and where we have used the assumption on the boundedness of \( \partial R_j/\partial c_i \) made in section 2.

(iii) For the invertibility of the block matrix

\[
\partial(Q_{eq}, Q_{neq})/\partial(\xi_{eq}, \xi_{neq})
\]

we use the formula

\[
\begin{pmatrix}
M_{11} & M_{12} \\
M_{21} & M_{22}
\end{pmatrix}^{-1} = \begin{pmatrix}
M'_{11} & M'_{12} \\
M'_{21} & M'_{22}
\end{pmatrix},
\]

\[
M'_{11} = (M_{11} - M_{12}M^{-1}_{22}M_{21})^{-1},
M'_{12} = -(M_{11} - M_{12}M^{-1}_{22}M_{21})^{-1}M_{12}M^{-1}_{22},
M'_{21} = -(M_{22} - M_{21}M^{-1}_{11}M_{12})^{-1}M_{21}M^{-1}_{11},
M'_{22} = (M_{22} - M_{21}M^{-1}_{11}M_{12})^{-1}.
\]

For the existence of all the terms on the right-hand side it is sufficient that the two blocks on the diagonal \( M_{11} = \partial Q_{eq}/\partial \xi_{eq} \) and \( M_{22} = \partial Q_{neq}/\partial \xi_{neq} \) are invertible, and that one of the two off-diagonal blocks \( M_{12}, M_{21} \) is small enough. The regularity of \( M_{11}, M_{22} \) was already shown in (i), (ii). For \( \Delta t \to 0 \),

\[
M_{21} = \frac{\partial Q_{neq}}{\partial \xi_{eq}} = -\Delta t \frac{\partial R_{neq}}{\partial (c, \bar{c})} S_{neq} \to 0.
\]

Hence, for \( \Delta t \) small enough the block matrix (61) is invertible. \( \square \)

**Proof of the fact that (15) is necessary for the existence of the resolution function of block 1 of (14).**

The local resolution function exists if and only if the derivative of the first block of (14) with respect to \( \xi_{eq} \), i.e., if matrix

\[
M := \frac{\partial R_{eq}}{\partial \bar{c}} S_{21},
\]

is invertible. If the columns of \( S_{21} \) are linear dependent, the nullspace of \( S_{21} \) is nonzero. For any \( \xi_{eq} \neq 0 \) taken from the nullspace of \( S_{21} \), \( M \xi_{eq} = 0 \) holds, i.e., \( M \) is not regular. \( \square \)

**Proof of theorem 2.** (i) We want to solve equation (37)

\[
f(\bar{\xi}, \xi) := \bar{\xi} - \bar{\xi}_{old} - \Delta t A_2 R_{neq} \begin{pmatrix} S^*_{1} \xi + S^*_1 \eta \\ S^*_2 \xi + S^*_2 \eta \end{pmatrix} = 0
\]

for \( \bar{\xi} \). According to the implicit function theorem, we have to check that the matrix \( \frac{\partial}{\partial \bar{\xi}} f(\bar{\xi}, \xi) \) is regular. It holds

\[
\frac{\partial}{\partial \bar{\xi}} f(\bar{\xi}, \xi) = I - \Delta t A_2 \frac{\partial R_{neq}}{\partial \bar{c}} S^*_2.
\]

Obviously, for

\[
\Delta t \| A_2 \frac{\partial R_{neq}}{\partial \bar{c}} S^*_2 \| < 1,
\]
where \(\|\cdot\|\) is an arbitrary submultiplicative matrix norm, the regularity is guaranteed.

(ii) (a) We have to prove regularity for matrix \(\frac{\partial f}{\partial \xi_{eq}}\) for

\[
f(\xi_{eq}, \xi_{neq}) := (S_{eq}^I)^t \ln \left( S_{eq}^I \xi_{eq} + S_{neq}^I \xi_{neq} + S_1^T \eta \right) - K.
\]

It holds

\[
\frac{\partial}{\partial \xi_{eq}} f = (S_{eq}^I)^t D_c S_{eq}^I
\]

where \(D_c\) is a diagonal matrix with diagonal entries being the positive values \(1/c_i\), \(i = 1, \ldots, I\). Since the columns of \(S_{eq}^I\) are linearly independent, \(\frac{\partial}{\partial \xi_{eq}} f\) is symmetric positive definite, thus regular (cf., proof of theorem 1(i)).

(ii) (b) We have to check that the matrix \(\frac{\partial (f_1, f_2)}{\partial (\xi_{eq}, \xi_{neq})}\) is regular, where

\[
f_1(\xi_{eq}, \xi_{neq}) = (S_{eq}^I)^t \ln \left( S_{eq}^I \xi_{eq} + S_{neq}^I \xi_{neq} + S_1^T \eta \right) - K,
\]

\[
f_2(\xi_{eq}, \xi_{neq}, \xi_{neq}, \xi_{eq}) = \xi_{eq} - \xi_{neq} - \Delta t \tilde{A}_2 R_{neq} \left( \begin{array}{c} S_{eq}^I \xi_{eq} + S_{neq}^I \xi_{neq} + S_1^T \eta \\ S_2^* \xi + S_2^T \eta \end{array} \right).
\]

We get

\[
\frac{\partial (f_1, f_2)}{\partial (\xi_{eq}, \xi_{neq})} = \begin{pmatrix}
(S_{eq}^I)^t D_c S_{eq}^I & 0 \\
-\Delta t \tilde{A}_2 \frac{\partial R_{neq}}{\partial \xi_{eq}} S_{eq}^I & I - \Delta t \tilde{A}_2 \frac{\partial R_{neq}}{\partial \xi_{eq}} S_2^*
\end{pmatrix}.
\]

For

\[
\Delta t \left\| \tilde{A}_2 \frac{\partial R_{neq}}{\partial \xi_{eq}} S_2^* \right\| < 1
\]

both diagonal blocks are regular and so is the whole matrix.

Let us remark on the restrictions of the timestep size (65), (66) found in the proof of the theorem: These restriction seem to be pessimistic. Let us consider for case (i) the situation where each reaction involves at most one immobile species (and an arbitrary number of mobile species) and where each immobile species participates in one reaction. By renumbering we can assume that \(S_2\) has the shape \(S_2 = (D 0)\) with \(D\) being a diagonal matrix. By multiplication of the columns of \(S\) by \(\pm 1\) we can assume that all entries of \(D\) are nonnegative. We get \(S_2^* = D\) and \(A_2 = (I 0)\). If mass action kinetics (3) is assumed, \(\frac{\partial R_{i}}{\partial \xi_{eq}} \leq 0\), \(\frac{\partial R_{i}}{\partial \xi_{eq}} = 0\) for all \(i \neq j\). Then (64) is a diagonal matrix with positive entries, i.e., \(\Delta t\) can be chosen arbitrarily large. Note that also for Example 1 (section 4.2), no restriction on \(\Delta t\) is required, since

\[
I - \Delta t \tilde{A}_2 \frac{\partial R_{neq}}{\partial \xi_{eq}} S_2^* = \begin{pmatrix}
1 + \Delta t k_{b_2} & -\Delta t k_{b_2} \\
-\Delta t k_{f_3} & 1 + \Delta t (k_{f_3} + k_{b_3})
\end{pmatrix},
\]

which is for arbitrary \(\Delta t > 0\) a strictly diagonal dominant matrix.
Notation

- $I$: number of mobile species
- $\bar{I}$: number of immobile species
- $J$: number of reactions
- $J_{eq}$: number of equilibrium reactions
- $J_{neq}$: number of kinetically controlled reactions
- $J_{l}$: number of linearly independent reactions, if all immobile species are ignored
- $J_{2}$: number of linearly independent reactions, if all mobile species are ignored
- $c$: concentration vector of mobile species, size $I$
- $\bar{c}$: concentration vector of immobile species, size $\bar{I}$
- $L$: a linear transport (advection-dispersion-diffusion) operator
- $R$: vector of reaction expressions, size $J$
- $R_{eq}, R_{neq}$: vector of equilibrium/non-equilibrium reaction expressions, size $J_{eq}, J_{neq}$
- $Q_{eq}$: vector of equilibrium expressions after log is applied to each component of $R_{eq}$
- $S$: $(I + 1) \times J$-matrix of stoichiometric coefficients $s_{ij}$
- $S_1$: $I \times J$-submatrix of $S$ of stoich. coeff. of mobile species
- $S_2$: $I \times J$-submatrix of $S$ of stoich. coeff. of immobile species
- $S_{1,l}$: $I \times J_{mot}$-submatrix of $S_1$ of linearly independent columns
- $S_{2,l}$: $I \times J_{immo}$-submatrix of $S_2$ of linearly independent columns
- $S_{eq,l}^{eq}, S_{neq}^{eq}$: $J_{eq} \times I-J_{neq}$-submatrix of $S_1$ for the equilibrium/kinetic reactions
- $S_{eq,l}^{eq}, S_{neq}^{eq}$: $J_{eq} \times I-J_{neq}$-submatrix of $S_2$ for the equilibrium/kinetic reactions
- $A_i$: matrix defined by (16)
- $S_{neq}^{eq}$: submatrix of $S_{neq}^{eq}$ defined by (31)
- $M^\top$: a matrix $M^\top$ consists of a maximum system of linearly independent columns that are orthogonal to the columns of matrix $M$.
- $t$: $M^\top$ is the transpose of $M$
- $\| \cdot \|$: any submultiplicative matrix norm
- $\langle \cdot, \cdot \rangle$: the Euclidean scalar product
- $\Delta t$: timestep size
- $x$: an arbitrary point in the computational domain

References


