Reactive Transport in Porous Media

Master of Science Thesis

Applied and Computational Mathematics

Pål Næverlid Sævik

Department of Mathematics
University of Bergen

Center for Integrated Petroleum Research

August 2011

Acknowledgements

First of all, I would like to thank my advisor, Ivar Aavatsmark, for introducing me to the exciting (but challenging!) subject of reactive transport in porous media, and for giving me directions along the way. Thanks to my fellow office-mates Svenn and Kristian, for sharing your knowledge about reservoir simulation techniques over a cup of coffee, time and again. The process of writing a master thesis would not have been as enjoyable without your cheerful company!

I also want to thank my parents, who were the first to introduce me to the fascinating world of mathematics and natural sciences. Thank you for all your support through 25 years, and for always being there for me.

Finally, to my beloved Elisabeth, thank you for everything you are to me. Your patience and support has been of tremendous value, and has kept my motivation strong throughtout the entire period of writing.

Pål,

July 2011

Introduction

Concerns about rising levels of atmospheric greenhouse gases, has lead researchers to propose several strategies for reducing CO_2 emissions. One of these is to capture CO_2 at various industrial point sources, such as coal-fired power plants, and store the gas underground. For instance, it has been suggested to inject captured CO_2 into oil and gas reservoirs, deep saline aquifers, and unminable coal beds. To investigate potential risks associated with geological storage, numerical simulations is a useful tool. However, it is a challenging task to include all relevant physical processes into such simulations. In particular, the inclusion of geochemical reactions introduces a number of modelling challenges and computational difficulties. Nevertheless, important effects can be lost by ignoring the reactive nature of the injected gas. CO_2 dissolves into the aquifer brine and reacts chemically with the ambient rock, either through dissolution or precipitation, possibly changing its porosity and permeability. It has therefore been speculated that chemical reactions may compromise the integrity of cap rock seals, causing leakage from the storage site. On the other hand, mineral precipitation may also provide an additional trapping mechanism, increasing the potensial for geological CO_2 storage.

In this thesis, we show how the common equations for flow in porous media can be expanded to account for geochemical reactions. Furthermore, the complications arising when solving the new equations numerically are described and explained. Specialised methods that alleviate the difficulties are then introduced, and discussed with respect to robustness and convergence properties. Much attention is directed to ways of reformulating the equations, in order to make them more amenable to numerical treatment. Also, the fact that chemical reactions introduce stiffness to the system is adressed. Diagonally implicit Runge-Kutta methods, which are commonly used to combat stiffness, are evaluated with respect to their usefulness in CO_2 sequestration simulations. Finally, we have applied the methods to several test cases, some including complex mineralogies, to illustrate the strengths and weaknesses of the different numerical approaches.

Contents

Chapter 1

Component Modelling

A geological formation suitable for CO_2 sequestration, typically consists of a porous rock layer filled with brine, with an impermeable cap rock seal above. At the site of injection, CO_2 will partially displace the brine inside the rock pores, and slowly start to migrate upwards due to density differences. For nonreactive simulations of the process, it is sufficient to keep track of the gas volume fraction within the pores, i.e., the *saturation* of the gas. Any simulation including reactions, on the other hand, must keep track of the composition of the phases as well. In other words, the concentration of chemical components within each phase, must be included as primary variables. In this chapter, we will present the equations that are typically used in component-based flow models, and show how they can be extended to account for chemical reactions. The model we present will be quite extensive, including, e.g., porosity changes and thermal effects. For many applications, such processes are neglible. However, there are situations where they may be of importance, and we want to show how to incorporate chemical reactions in these cases as well.

1.1 Choice of Primary Variables

The chemical species to be included in a mathematical model for CO_2 storage, will vary according to the effects one wishes to study. The simplest one-phase models include only water and aqueous CO_2, and are useful for studying convective density mixing within the aquifer brine. Such models are employed, for instance, by [11, 66, 53]. More extensive one-phase models include chemical reactions between the aqueous CO_2 and the mineral phase, like the model used by [12]. Two-phase com-

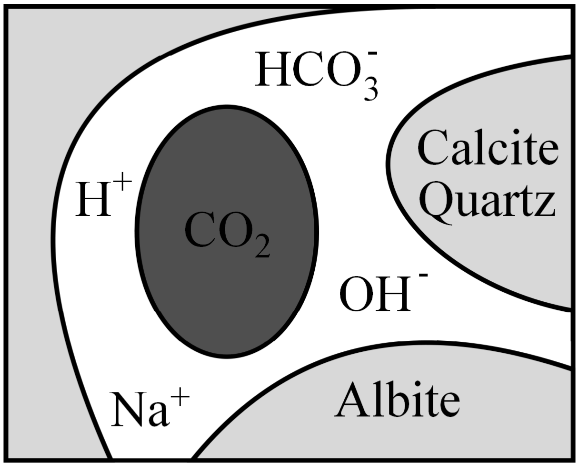


Figure 1.1.1: A component-based model of the pore fluid

ponent models also include a gaseous CO_2 component, and are used to study CO_2 plumes that migrate from an injection site towards the top of the aquifer. Complex simulations, including both gaseous and dissolved CO_2, as well as mineral reactions, have been performed by [31, 64, 52, 41, 51], for instance.

In compositional models involving reactions, chemical species that exists in multiple phases are commonly treated as separate components. We will follow the same approach in the model we present. For instance, the amounts of gaseous and aqueous CO_2 will be two different variables in our formulation, and the relation between them will be modelled as an instantaneous chemical equillibrium reaction. Thus, by definition, every component will exist in a single phase only. This is somewhat different from what is common in nonreactive models. The black oil model, for instance, defines pure and dissolved gas as a single component, even though it is spread out between two phases. With the introduction of chemical reactions, however, the individual treatment of pure and dissolved species appears to be a more natural approach. For instance, this formulation allows a simple and clean representation of reaction rates.

The gaseous phase will in general contain water vapor that dissolve into the gaseous phase. For most applications, the amount of water vapor will be very small, but for high-temperature, low-pressure conditions, its importance may be significant. To keep a clear presentation, we will in this thesis assume that CO_2 is the only gaseous component. However, models with significant amount of water vapor can be easily treated within the same framework, by simply adding gaseous H_2O as an additional component. We also note that a third liquid CO_2 phase may appear under certain low-pressure, low-temperature conditions. A non-aqueous liquid phase may also be required for simulations where CO_2 is used for enhanced oil recovery. In our presentation, we will assume that there are only two phases in the model, which is the most commonly occuring situation, but the extension to three-phase flow is straightforward.

1.1.1 Concentrations and Phase Volumes

We will use two different notions to describe the concentration of the components. The first is the *bulk concentration*, given by

$$c_i = \lim_{\Omega \to \mathbf{x}} \frac{N_i(\Omega)}{V(\Omega)},$$

where $N_i(\Omega)$ denotes the molar amount of the component i inside a region Ω of the reservoir, and $V(\Omega)$ is the volume of Ω. The concentration is defined in the limit where Ω approaches a single point $\mathbf{x}$. In contrast, we define the *interphasial concentration* as

$$\hat{c}_i = \lim_{\Omega \to \mathbf{x}} \frac{N_i(\Omega)}{V_\ell(\Omega)},$$

where ℓ is the phase containing component i, and $V_\ell(\Omega)$ is the volume of that phase inside Ω. The relation between c_i and $\hat{c}_i$ is given by $c_i = \psi_\ell \hat{c}_i$, where

$$\psi_\ell = \lim_{\Omega \to \mathbf{x}} \frac{V_\ell(\Omega)}{V(\Omega)}.$$

That is, ψ_ℓ is the relative volume of the phase ℓ at a certain location. We will refer to this quantity as the *phase volume fraction*. Alternatively, one can describe the amount of phases present by using the *porosity* ϕ and the *saturations* S_l, S_g. These variables are related by

$$\begin{aligned} \phi &= \psi_g + \psi_l \\ S_g &= \psi_g / \phi \\ S_l &= \psi_l / \phi, \end{aligned}$$

where the subscripts g and l denote the gas and liquid phase, respectively. We will use both sets of variables, depending on which choice is the most convenient.

We will choose the bulk concentrations $\mathbf{c} = (c_1, \ldots, c_m)^\top$, together with the pressure p and the temperature T, as primary variables in our formulation. The interphasial concentrations and the phase volume fractions will be secondary variables, that are calculated from the bulk concentrations whenever necessary. Of course, other choices of primary variables are possible as well. Most notably, the saturations and the interphasial concentrations are often used as primary variables in immiscible multiphase models. However, we want a model that allows mass exchange between the phases, i.e., heterogenous chemical reactions. In this case, the governing equations are simplified by expressing them in terms of the bulk concentrations.

1.1.2 Compressible Reservoirs

In the definitions above, it is assumed that the bulk volume of the reservoir is constant during the simulation. We briefly note that the volume of a reservoir region may also be modelled as pressure-dependent. In this case, the bulk volume of a reservoir region is often calculated by the relation

$$V(\Omega,\, p) = \left[1 + \alpha\, (p - p_0)\right] V\left(\Omega,\, p_0\right),$$

where α is the rock compressibility factor, p the current pressure, and p_0 the initial pressure. For compressible reservoirs, we define the concentrations and volume fractions by the relations

$$c_i = \lim_{\Omega \to \mathbf{x}} \frac{N_i(\Omega)}{V\left(\Omega,\, p_0\right)}$$

and

$$\psi_\ell = \lim_{\Omega \to \mathbf{x}} \frac{V_\ell(\Omega)}{V\left(\Omega,\, p_0\right)}.$$

The other definitions remain unaltered.

1.1.3 Calculating Phase Volumes

Since phase volume fractions are secondary variables in our formulation, we will need to calculate their values from the primary variables. For a specific phase ℓ, the volume fraction is calculated by the simple relation

$$\psi_\ell = \sum c_i/\rho_i, \qquad (1.1.1)$$

where ρ_i is the molar density of component i, and the sum ranges over all components in the phase. The aqueous phase often contains components that has little influence on the phase volume, since their concentrations are vanishingly small compared to the water component. These can safely be disregarded from (1.1.1). The molar densities of the remaining components can be regarded as constant in this context, except for the density of CO_2, which should usually be determined from an equation of state. For this purpose, the Peng-Robinson or Soave-Redlich-Kwong equation is often used. In some applications, however, it may be safe to assume that the density of CO_2 is constant as well. This is especially true for deep saline aquifers, where the pressure is high, and CO_2 acts more like a liquid than a gas.

Naturally, we want the calculated phase volume fractions to sum up to 1, or $1 + \alpha\,(p - p_0)$ in the case of compressible rocks. This constraint is known as the *volume balance equation*,

$$\sum_{i=1}^{m} \frac{c_i}{\rho_i} = 1 + \alpha\,(p - p_0)\,. \tag{1.1.2}$$

There are two principal ways of enforcing volume balance during numerical simulations. One way is to use the volume constraint to eliminate the conservation equation for either water or CO_2, which is useful when the system is solved by a fully coupled numerical method. The other is to keep the volume balance as an additional governing equation, which is the usual choice for methods based on operator splitting. We will follow the latter approach, as some form of operator splitting is normally employed when including chemical reactions in the formulation.

1.2 The Pressure Equation

When keeping the volume balance as a primary equation, the governing equations becomes a *differential-algebraic system* (DAE), that is, a combination of differential and algebraic equations,

$$\begin{aligned}
0 &= \mathcal{V}(p,\, T,\, c_1,\, \ldots,\, c_m,\, t) && \text{(Volume balance)} \\
\frac{\partial c_i}{\partial t} &= \mathcal{M}_i(p,\, T,\, c_1,\, \ldots,\, c_m,\, t) && \text{(Mass balances)} \\
\frac{\partial T}{\partial t} &= \mathcal{T}(p,\, T,\, c_1,\, \ldots,\, c_m,\, t) && \text{(Energy balance)}
\end{aligned} \tag{1.2.1}$$

The specific formulations for mass and energy balances are given later. We see that the system contains no explicit mention of the pressure derivative with respect to time. Therefore, the system must (at least partially) be solved implicitly. In addition, the volume balance equation is only weakly dependent on pressure, especially in regions where no gas is present. Thus, the system has a near-singular Jacobian, and is essentially an *index-2* DAE. Such systems are difficult to solve, as the usual implicit integration methods require a nonsingular Jacobian to converge.

To alleviate this problem, it is common to construct a *pressure equation*, in which the derivative of the pressure appears explicitly. This can be acheived, for instance, by replacing the volume balance equation by its derivative. In general, such a procedure is called *index reduction* of the differential-algebraic system. Let R be the departure from volume balance,

$$R\left(p,\, T,\, c_1,\, \ldots,\, c_m\right) = \sum_{i=1}^{m} \frac{c_i}{\rho_i} - 1 - \alpha\left(p - p_0\right), \tag{1.2.2}$$

where we have included the rock compressibility for completeness. The pressure equation is obtained by differentiating with respect to time, and using the chain rule[1],

$$\frac{\partial R}{\partial t} = \frac{\partial R}{\partial p}\frac{\partial p}{\partial t} + \frac{\partial R}{\partial T}\frac{\partial T}{\partial t} + \sum_{i=1}^{m} \frac{\partial R}{\partial c_i}\frac{\partial c_i}{\partial t}. \tag{1.2.3}$$

Here, the volume derivatives $\frac{\partial R}{\partial p}$, $\frac{\partial R}{\partial T}$ and $\frac{\partial R}{\partial c_i}$ can be calculated directly from (1.2.2), while the derivatives $\frac{\partial T}{\partial t}$ and $\frac{\partial c_i}{\partial t}$ are replaced by the conservation equations (1.4.3) and (1.3.1).

Physically, the pressure equation describes how the pressure must change in order to maintain volume balance. The two last terms describe the rate of volume change due to physical processes, and the left-hand side describes how the departure from volume balance changes with time. It is tempting to set $\frac{\partial R}{\partial t} = 0$, since $R = 0$ at the beginning of the simulation, and we want R to stay at this value. However, at the beginning of a time step, we may have $R \neq 0$ due to numerical errors. Thus, setting $\frac{\partial R}{\partial t} = 0$ will preserve the errors, leading to an unstable computation. Instead, we want the discrepancy corrected, so that volume balance is acheived at the end of

[1] Equation (1.2.3) is sometimes derived differently in the oil and gas literature. In the influential paper by Watts[61], for instance, the equation is derived by starting with the mass conservation equations, multiplying with the volume factors $\frac{\partial R}{\partial c_i}$, then summing the expressions, and finally interpreting the result as the derivative of the volume balance. In essence, the two derivations are the same, only with steps reversed.

each step. For this purpose, it is common to set

$$\frac{\partial R}{\partial t} = \frac{R_{final} - R_{initial}}{\Delta t},$$

where $R_{final} = 0$ is the desired value at the end of the step, $R_{initial}$ is the actual value of R at the beginning of the step, and Δt is the time step length. This term acts like a control variable, ensuring that the solution stays close to thermodynamic equillibrium. It is interesting to note that this correction procedure bears close resemblance to the *Baumgarte stabilisation* [5], a common technique for stabilising index-reduced DAE systems.

Most chemical reactions will have neglible effect on the fluid volume, and can be omitted from (1.2.3). The main exception is the transition from gaseous to aqueous CO_2, which is modelled as an equillibrium reaction. Since aqueous CO_2 occupies less space, dissolution of CO_2 into water leads to a reduction in the total fluid volume. Also, the reaction changes the saturations and the relative permeabilities of the phases. The easiest way of including this effect into the pressure equation, is to apply a formal change of variables. Let c_g be the concentration of gaseous CO_2, c_d the concentration of dissolved CO_2 and $c_t = c_g + c_d$ the total CO_2 concentration. Using the phase equillibrium assumption, we can define c_g and c_d as functions of c_t. Consequently, we can make the following replacement in the pressure equation,

$$\begin{aligned} \frac{\partial R}{\partial c_g}\frac{\partial c_g}{\partial t} + \frac{\partial R}{\partial c_d}\frac{\partial c_d}{\partial t} &= \frac{\partial R}{\partial c_g}\frac{\partial c_g}{\partial c_t}\frac{\partial c_t}{\partial t} + \frac{\partial R}{\partial c_d}\frac{\partial c_d}{\partial c_t}\frac{\partial c_t}{\partial t} \\ &= \left(\frac{\partial R}{\partial c_g}\frac{\partial c_g}{\partial c_t} + \frac{\partial R}{\partial c_d}\frac{\partial c_d}{\partial c_t}\right)\left(\frac{\partial c_g}{\partial t} + \frac{\partial c_d}{\partial t}\right). \end{aligned}$$

Since the total concentration $c_g + c_d$ is unaffected by the phase transition reaction, we have thus obtained a pressure equation without chemistry terms. In models where water vapor is included, the same approach can be applied to the gaseous and liquid water component.

The procedure outlined above is equivalent to treating gaseous and dissolved CO_2 as one single component, which is precisely what we discouraged in Section 1.1. But in this particular case, the total concentration is a natural variable of choice. Nevertheless, we still want to keep two separate variables for the gaseous and dissolved CO_2, as this is more convenient for the rest of the equations we are considering.

For simple models where all the densities are constant, and the rock compressibility is zero, equation (1.2.3) simplifies to

$$\frac{\partial R}{\partial t} = \sum_{i=1}^{m} \frac{1}{\rho_i}\frac{\partial c_i}{\partial t}, \qquad (1.2.4)$$

which is the *incompressible pressure equation.* When inserting the expression (1.3.1) for $\frac{\partial c_i}{\partial t}$, this becomes an elliptic equation for pressure, since the time derivative of the pressure is not involved. Also, if there are no phase transitions, the volume constraint (1.1.2) can be enforced explicitly during the solution procedure, so that no additional stabilisation is required. Thus, one can use $\frac{\partial R}{\partial t} = 0$ without concern. Incompressible models are especially relevant when studying CO_2 storage scenarios where the gas has been completely dissolved in the brine. In this case, the only mobile phase is the incompressible aqueous phase. As (1.2.4) is easier to solve than (1.2.3), some authors use incompressible models for multiphase simulations as well. One example is the semi-analytical migration model of [42], another is the streamline-based approach used by [43]. CO_2 is a dense, supercritical fluid at reservoir conditions, so the incompressibility approximation may be justified. The relative importance of CO_2 compressibility is, however, problem-dependent. In particular, dissolution of CO_2 into brine may lead to a significant overall volume change. Obi and Blunt deals with this issue by assuming that dissolved CO_2 has the same volume as gaseous CO_2. While this is clearly incorrect, the fraction of dissolved CO_2 in brine is typically small. Their approximation may therefore be sound, especially if other kinds of errors dominate.

1.3 Mass Balance

The mass conservation equations describe how the concentrations the chemical components will change with time, due to physical processes within the fluid. In summary,

$$\text{Rate of change} \quad = \quad \text{Advection} \quad + \quad \text{Dispersion} \quad + \quad \text{Reactions} \tag{1.3.1}$$

There will be one such equation for every component present. In the following, we will give a mathematical description of each term in (1.3.1). The relative importance of the terms depend on the component in question. For instance, the concentrations of the aqueous ions ${CO_3}^{2-}$ and OH^- are typically small, and their concentrations are mainly determined by the chemical reactions. On the other hand, the concentration of water is hardly affected by reactions at all, since the amount of water is very large compared to the reaction scale.

Advection refers to the large-scale fluid movement caused by pressure differences and boyancy forces. Let ℓ be the phase that contains component i, and let $\mathbf{u}_\ell$ be the volumetric flow rate of that phase, often called the *Darcy velocity.* Mass change due to advection can then be conveniently expressed as

$$\text{Advection } = -\nabla \bullet (\hat{c}_i \mathbf{u}_\ell) , \tag{1.3.2}$$

where $\hat{c}_i = c_i/\psi_\ell$. Of course, $\mathbf{u}_s = 0$, since the solid phase is not moving. For the fluid and gas phase, the Darcy velocity is given by *Darcy's law*,

$$\mathbf{u}_\ell = K \frac{k_\ell^{rel}}{\mu_\ell} (\nabla p_\ell + \rho_\ell \mathbf{g}) . \tag{1.3.3}$$

Here, K is the permeability of the rock, k_ℓ^{rel} the relative permeability of the phase, μ_ℓ the viscosity, p_ℓ the pressure, ρ_ℓ the density, and $\mathbf{g}$ the acceleration of gravity. While it is common to set $p_g = p_l = p$, the pressures of the gas and liquid phases are, in general, not equal. In cases where the discrepancy is important, we set $p_l = p$ and $p_g = p + P_c$. Here, P_c is the *capillary pressure*, which can be calculated from the fluid composition. We refer to [19] for further details.

Usually, the permeability K is taken to be constant. However, precipitation and dissolution of minerals may change the value of this parameter. To relate rock permeability to change of porosity, the Kozeny-Carman equation is often used,

$$K = K_0 \left(\frac{\phi}{\phi_0} \right)^3 \left(\frac{1 - \phi_0}{1 - \phi} \right)^2 , \tag{1.3.4}$$

where K_0 and ϕ_0 are the initial permeability and porosity, respectively. Laboratory tests show that the permeability changes due to CO_2 injection may be of significant importance for certain mineral assemblages[62, 40].

Dispersion is caused by small, turbulent currents occuring at the pore scale during advection. The large-scale effects of dispersion is similar to that of molecular diffusion, it causes mass to move from areas of high concentration, to areas of low concentration. While diffusion is usually neglible in reservoir simulation models, dispersion may be of importance. A simple model of dispersion is provided by Fick's law,

$$\text{Dispersion } = \nabla \bullet (k_D \left\| \mathbf{u}_\ell \right\| \nabla c_i) ,$$

where k_D is the rock- and fluid-dependent dispersion coefficient. It should be noted, however, that many numerical solution methods create artificial diffusion, which is usually of greater magnitude than the physical dispersion described by the term above. When using a diffusive numerical scheme, one may thus neglect the dispersion term altogether.

Chemical reactions is the final process that may change the concentrations of the components. While advection and dispersion only applies to the components in the mobile phases, reactions may influence solid minerals as well. The specific mathematical formulation of this term is model-dependent, and will be discussed later. At present, we simply write

$$\text{Reactions } = R_i\,(p,\,T,\,c_1,\,\ldots,\,c_m)\,,$$

where R_i is the molar production rate of component i, per volume. The rate is often strongly and nonlinearly dependent of other components at the same location, which complicates the treatment of the mass conservation equation. On the other hand, unlike advection and dispersion, the reaction term is *spatially decoupled.* That is, the reaction rates are independent of the component concentrations at neighbouring locations. This is an important feature that can be exploited for more efficient numerical integration.

In addition to the source terms described above, one can also include external mass sources in the model, for instance, production and injection wells. If we denote this additional term by Q_M, the total equation for mass conservation is given by

$$\frac{\partial}{\partial t}c_i = -\nabla \bullet \left(\frac{c_i}{\psi_\ell}\mathbf{u}_\ell\right) + \nabla \bullet \left(k_D \left\|\mathbf{u}_\ell\right\| \nabla c_i\right) + R_i\,(p,\,T,\,c_1,\,\ldots,\,c_m) + Q_M. \quad (1.3.5)$$

1.4 Energy Balance

The energy conservation equation is the last of the governing set of equations. For most simulations of CO_2 storage scenarios, the reservoir temperature may be regarded as constant, or at least time-independent. However, non-isothermal effects can be significant during the early injection period[65], or when considering leakage of CO_2 through abandoned wells[50, 10]. There are also instances where the heat generated from chemical reactions may play an important role. For instance, it has been theorised that exothermic geologic reactions might help sustain an optimal temperature for in-situ carbonation[29]. To assess such questions, we need an equation that describes the evolution of the reservoir temperature. As a starting point, we use the principle of energy conservation,

$$\begin{aligned}\text{Rate of energy change } &= \text{Convection} + \text{Conduction}\\ &\qquad + \text{Compression/Dissipation.}\end{aligned} \quad (1.4.1)$$

Note that the effects of chemical reactions is absent from this equation. While reactions do influence the temperature of the reservoir fluid, they only cause a transition from chemical to thermal energy, and do not affect the total internal energy per se. To capture the thermal effects of reactions, we therefore need a relation between the temperature and the energy. For each component i, the internal energy per mole is given by

$$e_i = \Delta_f E_i^{\ominus} + \int_{T^{\ominus}}^{T} C_i \, dT.$$

Here, $\Delta_f E^{\ominus}$ is the standard energy of formation per mole, $T^{\ominus}$ the temperature in the reference state, and C the molar heat capacity at constant volume. Strictly speaking, the heat capacity is dependent on pressure as well, but this dependence is usually small. Assuming no temperature differences between the phases, we get

$$\begin{aligned} \text{Rate of energy change } &= \frac{\partial}{\partial t} \sum_{i=1}^{m} e_i c_i \\ &= \frac{\partial T}{\partial t} \sum_{i=1}^{m} c_i C_i + \sum_{i=1}^{m} e_i \frac{\partial c_i}{\partial t}, \end{aligned}$$

where we have summed over all the components. Using equation (1.3.5), we can rewrite the last term as

$$e_i \frac{\partial c_i}{\partial t} = -e_i \nabla \bullet (\hat{c}_i \mathbf{u}_i) + e_i R_i.$$

As before, $\hat{c}_i$ is the interphasial concentration of component i, and $\mathbf{u}_i$ is the Darcy velocity for the phase containing the component. We have ignored the dispersion term, since its thermal effect is typically small for the applications we are considering. Also, we have delayed the inclusion of the source term. With these simplifications, we finally arrive at

$$\text{Rate of energy change } = \frac{\partial T}{\partial t} \sum_{i=1}^{m} c_i C_i - \sum_{i=1}^{m} e_i \nabla \bullet (\hat{c}_i \mathbf{u}_\ell) + \sum_{i=1}^{m} e_i R_i.$$

Having established the link between temperature and energy change, we proceed to define the other terms of equation (1.4.1). First of all, the energy transport due to *convection* is given by

$$\text{Convection } = -\sum_{i=1}^{m} \nabla \bullet (e_i \hat{c}_i \mathbf{u}_i)\,. \tag{1.4.2}$$

Again, we can rewrite this term to emphasise its temperature dependence, giving the equivalent expression

$$\text{Convection} \;= -\sum_{i=1}^{m} e_i \nabla \bullet (\hat{c}_i \mathbf{u}_i) - \sum_{i=1}^{m} \hat{c}_i C_i \mathbf{u}_i \bullet \nabla T.$$

The second term in equation (1.4.1) is *conduction*, which refers to heat transfer caused by molecular vibrations. Its large-scale effect is to even out temperature differences within the fluid. Conduction is usually modelled by Fourier's law, which is similar to Fick's law of diffusion,

$$\text{Conduction} \;= \nabla \bullet (k_F \nabla T)\,,$$

where k_F is called the thermal conductivity. If conduction is the only important thermal effect, (1.4.1) is completely decoupled from the remaining model equations, allowing the assumption of a constant temperature gradient. Thus, the energy equation is only useful when the other thermal effects are of comparable magnitude.

The last term in (1.4.1) is given by

$$\text{Compression / Dissipation} \;= \sum_{\ell \in \{g,l\}} \left(-p_\ell \nabla \bullet \mathbf{u}_\ell + \boldsymbol{\tau}_\ell : \nabla \mathbf{u}_\ell\right),$$

where $\boldsymbol{\tau}$ is the dissipation tensor, and the sum ranges over the mobile phases. This expression describes the conversion between mechanical and internal energy, that is, the heat generated from *compression work* and *viscous dissipation*. For typical applications, both of these terms are much smaller than the other energy terms, and can be disregarded[15].

As for the mass conservation equation, we can also include *external energy sources* in our model, for instance due to injection and production wells. Denoting these sources by Q_E, the final equation for energy conservation is given by

$$\frac{\partial}{\partial t} \sum_{i=1}^{m} e_i c_i = -\sum_{i=1}^{m} \nabla \bullet (e_i \hat{c}_i \mathbf{u}_i) + \nabla \bullet (k_F \nabla T) + Q_E,$$

or, using the temperature as primary variable,

$$\frac{\partial T}{\partial t} \sum_{i=1}^{m} c_i C_i = -\sum_{i=1}^{m} \hat{c}_i C_i \mathbf{u}_i \bullet \nabla T - \sum_{i=1}^{m} e_i R_i + \nabla \bullet (k_F \nabla T) + Q_E. \tag{1.4.3}$$

1.5 Numerical Solution of the System

Having established the governing set of equations, we briefly comment on how the system can be solved. Our final system of equations is abstractly described as

$$\frac{\partial p}{\partial t} = \mathcal{P}(p,\, T,\, c_1,\, \ldots,\, c_m,\, t) \qquad \text{(Pressure equation)} \qquad (1.5.1a)$$

$$\frac{\partial c_i}{\partial t} = \mathcal{M}_i(p,\, T,\, c_1,\, \ldots,\, c_m,\, t) \qquad \text{(Mass balances)} \qquad (1.5.1b)$$

$$\frac{\partial T}{\partial t} = \mathcal{T}(p,\, T,\, c_1,\, \ldots,\, c_m,\, t). \qquad \text{(Energy balance)} \qquad (1.5.1c)$$

Usually, the system is first discretised in space, before integrating in time (the *method of lines*). The computational domain is then divided into discrete grid blocks, and the spatial differential operators are discretised accordingly. There is, of course, more than one way of doing this. A multitude of different discretisation schemes are found in the literature, each with their own advantages. It is even possible to use *adaptive* schemes, where the spatial grid is changed for every time step.

After discretisation, (1.5.1) becomes a set of ordinary differential equations. For simple models containing few chemical components (for instance, the black-oil model), it is often feasible to solve this system directly, using an implicit integration method. This is called a *fully coupled*, or *fully implicit* approach. It is a stable and robust solution method, but very memory- and time-consuming if the number of components is high. For two- and three-dimensional simulations, the difficulties are even more severe.

An alternative approach is to solve the equations *sequentially*. For instance, one can compute the pressure evolution within a time step by assuming that the flow determining parameters (e.g. the viscosities and relative permeabilities) are constant. This way, the pressure equation is decoupled from the other equations, leaving a smaller system to solve. Having determined the new pressure gradient, one can solve the mass and energy conservation equations subsequently. Such a procedure is illustrated in Figure 1.5.1.

Sequential solution methods introduce additional *splitting errors* because we assume constant parameters when computing the pressure gradient. On the other hand, the approach allows smaller step sizes to be used, since each step is faster to compute. For systems with many components, this benefit makes sequential methods more attractive than fully coupled methods. We also note that many variants of sequential

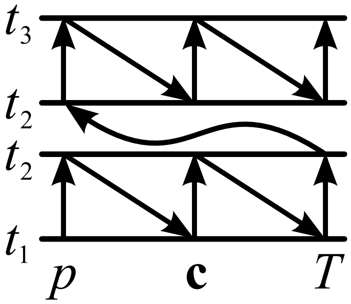


Figure 1.5.1: A simple splitting scheme

schemes exists. For instance, the pressure equation can be solved together with the mass conservation equation for CO_2 and water, to reduce the splitting errors. It is also possible to use iterative schemes, in which the pressure equation is re-solved after the mass and energy change is computed, to correct the splitting errors.

In this thesis, our main focus will be the mass conservation equation, since this is the part of (1.5.1) that is mostly affected by introducing chemically reactive species. Although the other two equations are affected as well, these can usually be solved by the numerical techniques commonly used for nonreactive models. The mass conservation equation, on the other hand, must be solved in an entirely different way when chemistry is introduced.

Chapter 2

Structure of the Reactive Term

We have now formulated the differential equations used to describe reactive flow through a porous medum. As we have seen, an accumulation term corresponding to chemical reactions shows up in both the mass and energy conservation equations. In this chapter, we will take a closer look at the specific form of this term, and present some geochemical models that are commonly used for simulations of reactive transport in porous media. As these often give rise to strongly coupled nonlinear equations, they can only be solved analytically for simple cases.

2.1 The Kinetic Mass Action Law

To begin our discussion, let us consider a single chemical reaction of the form

$$r_1 R_1 + \ldots + r_n R_n \to p_1 P_1 + \ldots + p_m P_m, \tag{2.1.1}$$

where $R_1, \ldots, R_n$ are the reactants, $P_1, \ldots, P_m$ the products, while $r_1, \ldots, r_n$ and $p_1, \ldots, p_m$ are their corresponding stoichiometric coefficients. In addition, we define a variable ξ, called the *reaction progress variable*, or the *extent of reaction.* This may be thought of as a hypothetical reaction product, of which one mole is produced every time the reaction event occurs. A common model for the reaction rate is the *kinetic mass action law*,

$$\frac{d\xi}{dt} = k\left(\prod_{i=1}^{n} a\left(R_i\right)^{r_i} - \frac{1}{K}\prod_{j=1}^{m} a\left(P_j\right)^{p_j}\right). \tag{2.1.2}$$

In this equation, k is the rate constant, K is the equillibrium constant and a denote the activity of a component.

We will now give a brief description of each parameter in (2.1.2), how they can be calculated, and how they depend on the fluid state. One must bear in mind, however, that neither of the models presented give an entirely accurate description of the physical processes involved. Geochemical reaction modelling is a complex field, and there exists numerous sources of modelling errors. As always when building a mathematical model, there will be a tradeoff between model accuracy and computational speed. The very simplest models give unreliable results, while it often unnecessary to use the most comprehensive and time-consuming approaches.

2.1.1 Activity of Gases

The activity of a gaseous component can be computed from its partial pressure p_{part} as follows,

$$a = \frac{\varphi p_{part}}{p^{\ominus}},$$

where $p^{\ominus}$ is a reference pressure (usually 1 bar), and φ is called the *fugacity coefficient*. This parameter can be calculated from the expression

$$\ln \varphi = \int_{p_0}^{p} \frac{Z-1}{p}\, dp,$$

where Z is the compressibility factor of the gas, and p_0 is a pressure level where the gas behaves ideally. For gaseous CO_2 at typical reservoir conditions, the fugacity coefficient varies between 0.4 and 0.8.

2.1.2 Activities of Dissolved Species

For a component dissolved in the aqueous phase, the activity is calculated from its interphasial concentration $\hat{c}$,

$$a = \gamma \frac{\hat{c}}{\hat{c}^{\ominus}},$$

where $\hat{c}^{\ominus}$ is a reference concentration (usually 1 mol/dm^3), and γ is the *activity coefficient*. The parameter γ is strongly dependent on the *ionic strength* I of the solution, which is defined as

$$I = \frac{1}{2} \sum \hat{c}_i z_i^2.$$

Here, z_i is the charge number of component i, and the sum ranges over all the dissolved species. There are different models for calculating activity coefficients from the ionic strength. [20] suggests the Truesdell-Jones activity model, which is given by

$$\ln(\gamma) = \frac{-Az^2\sqrt{I}}{1 + \alpha B\sqrt{I}} + \beta I,$$

where A and B are temperature- and solvent-dependent parameters, while α and β are ion specific parameters. This model provides acceptable accuracy for the typical ionic strengths of aquifer brines. Other alternatives include the similar B-dot model, the simpler Davies equation, and the more accurate Pitzer equations[7].

At the ionic strengths encountered in typical aquifer brines, the activity coefficients of charged species are significantly lower than 1. Monovalent species have activity coefficients of magnitude 0.6 - 0.9, divalent ions in the range 0.1 - 0.4. Multivalent ions like Al^{3+} show even lower values, while the activities of nonpolar species (like $SiO_2(aq)$) are in the range 1.1 - 1.5.

Inaccurate calculations of activities may affect the apparent solubilities of the minerals, and is often a major error source in geochemical calculations. Thus, one should always choose an activity model that reflects the accuracy requirements of the other parts of the simulation. To ensure precise activity predictions, it may be helpful to couple the reactive transport code with an external program specialised for this purpose. For instance, the widely used speciation program PHREEQC[46], which is freely available, is designed to be easily integrated with other software.

2.1.3 Activities of Solids and Liquids

For ideally behaving systems, the activity of solids and liquid solvents (like water) are set to 1. While the activity of water is reduced in solutions of high ionic strength, the activity coefficient is still close to 1 for typical aquifer conditions. For instance, in sea water, H_2O has an activity of 0.95. Consequently, the error of assuming ideal activity is not as big for water as for aqueous ions. Details on how to predict the activity of water more accurately may be found, for instance, in [7]. Such models are also implemented in software like PHREEQC.

2.1.4 The Equillibrium Constant

We first define the *ion activity product* Q, which is given by the relation

$$Q = \frac{\prod_{i=1}^{n} a\left(R_i\right)^{r_i}}{\prod_{j=1}^{m} a\left(P_j\right)^{p_j}}.$$

The equillibrium constant K is the value of the ion activity product when the reaction rate is zero. This value may be calculated from thermodynamic data, using the relation

$$RT \ln K = -\Delta H^{\ominus} + T\Delta S^{\ominus}. \tag{2.1.3}$$

Here, $\Delta H^{\ominus}$ is the standard enthalpy change for the reaction, $\Delta S^{\ominus}$ is the standard entropy change, and R is the gas constant. The equillibrium constant is only negligibly affected by pressure changes. This may seem counterintuitive, as the solubility of CO_2, for instance, increases significantly with the pressure. However, this is mainly due to the increased activity of the gaseous CO_2 component. On the other hand, K is clearly dependent on the temperature. If a tabulated equillibrium constant K_0 at a certain reference temperature T_0 is available, one may use (2.1.3) to calculate K at a different temperature, by assuming that $\Delta H^{\ominus}$ and $\Delta S^{\ominus}$ change little with temperature. This is called the *van 't Hoff equation*,

$$\ln\left(K/K_0\right) = -\frac{\Delta H^{\ominus}}{R}\left(\frac{1}{T} - \frac{1}{T_0}\right).$$

For typical mineral dissolution reactions (see Table 2.1.2), $\Delta H^{\ominus}$ varies between -600 kJ/mol (highly exothermic) and 30 kJ/mol (slightly endothermic). Temperature changes can have a large impact on the equillibrium constant if the reaction enthalpy is large. For instance, if $\Delta H^{\ominus} = -600\,\mathrm{kJ/mol}$, a temperature increase from 80 °C to 100 °C may reduce the equillibrium constant by a factor of 10^{-5}. As a consequence, minerals that are stable in one region of the reservoir, may be unstable in another region, even if the composition of the pore fluid is equal at both locations.

The dissolution of CO_2 into water is also a temperature-dependent process, with $\Delta H^{\ominus} = -20\,\mathrm{kJ/mol}$. Thus, the solubility of the gas decreases with increasing temperatures. As the CO_2 dissolution process is often important to model accurately, CO_2 solubility has been tested experimentally for large ranges of pressure, temperature and salinity conditions. Such data have been used to construct accurate expressions for the solubility of CO_2, which may be applied independently of the chosen activity and fugacity models (see, for instance, [9]).

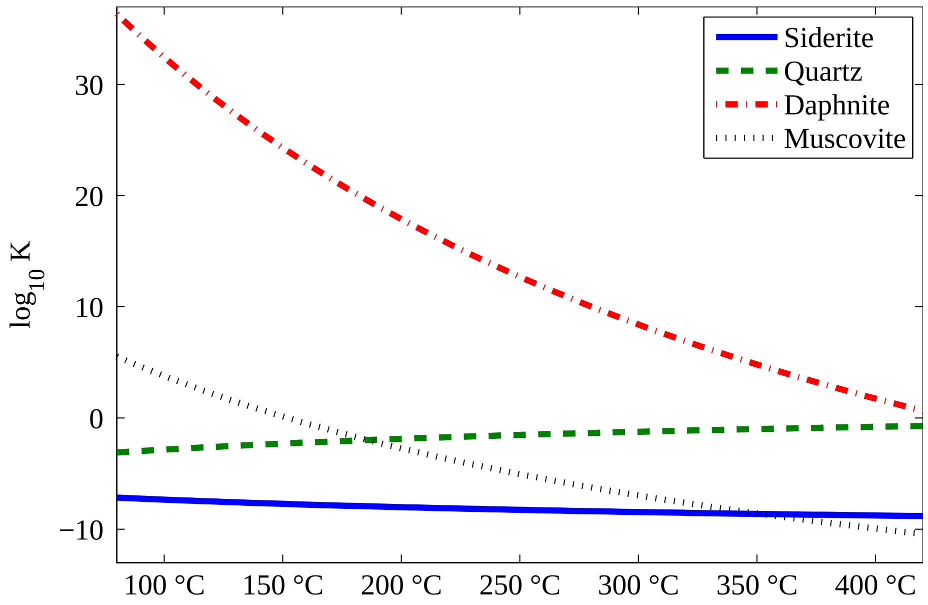


Figure 2.1.1: Temperature dependence for some equillibrium constants

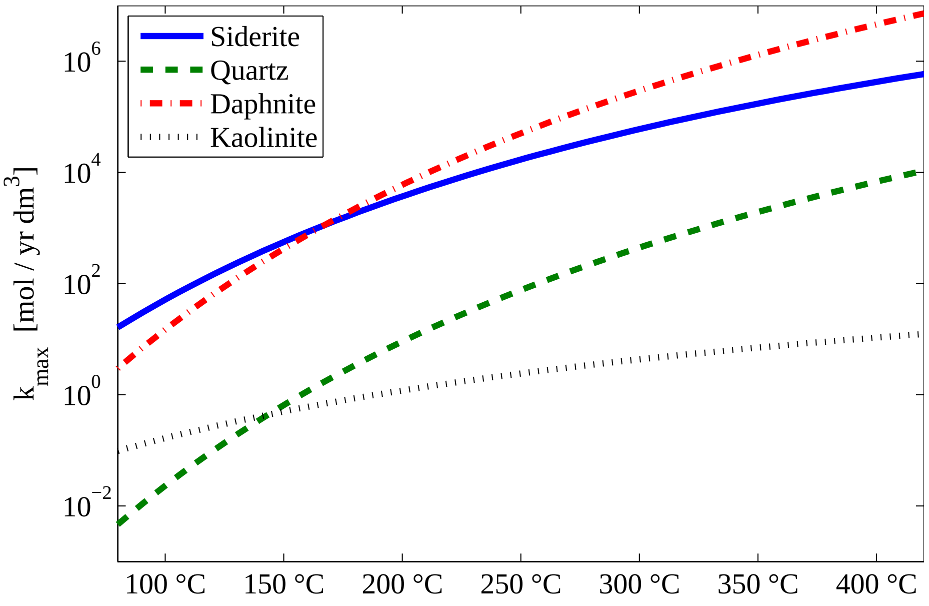


Figure 2.1.2: Temperature dependence for some rate constants

2.1.5 Equillibrium State Expression

From (2.1.2), it is clear that the reaction rate is zero whenever $Q = K$. However, since concentrations never can be negative, the reaction rate may also be zero if any of the reactants are absent from the system. For instance, if the brine is subsaturated with respect to a certain mineral, the mineral will dissolve until either saturation is acheived (which corresponds to $Q = K$), or until the amount m of mineral present have reached zero. Thus, the equillibrium condition can be expressed as

$$Q = K,\, m > 0 \quad \wedge \quad m = 0,\, Q < K.$$

Here, it is assumed that the mineral appears on the left-hand side of (2.1.1). Kräutle[33] showed that this condition can be expressed compactly as one single equation,

$$\min{(m,\, K - Q)} = 0. \tag{2.1.4}$$

Thus, whenever a mineral reaction satisfies (2.1.4), the dissolution/precipitation rate is zero, and chemical equillibrium has been attained. When finding equillibrium concentrations numerically, we will use a slightly different form of this expression,

$$\min{(m,\, \ln K - \ln Q)} = 0, \tag{2.1.5}$$

since the two arguments m and $\ln{(K/Q)}$ are then of the same magnitude.

2.1.6 The Rate Constant

A common model for the rate constant k is the Arrhenius equation,

$$k = Ae^{-E_a/RT},$$

where E_a is the *apparent activation energy* of the reaction, and A is the *pre-exponential factor*, both of which are determined experimentally. For a dissolution or precipitation reaction, A is proportional to the mineral surface area, which is usually taken to be a simple function of its volume fraction. For instance, one may set

$$A = A_0 + A_1 c, \tag{2.1.6}$$

where c is the mineral's bulk concentration, A_1 is the reactive surface area per mole, and A_0 is the "minimal" reactive area per volume. This is a very crude model, since surface areas change in complex ways during dissolution and precipitation. In particular, it is difficult to estimate the “reactive surface” for a mineral which is not initially present. Nevertheless, as few alternatives exists, the simple relation (2.1.6) is commonly used.

Often, rate constants for mineral reactions are tabulated at a certain reference temperature. Rearranging Arrhenius’ equation, these values can be used to calculate rate constants at different temperatures as well,

$$\ln\left(k/k_0\right) = -\frac{E_a}{R}\left(\frac{1}{T}-\frac{1}{T_0}\right).$$

Here, k_0 is the reference rate, and T_0 the reference temperature. For typical dissolution reactions, E_a varies between 20 kJ/mol and 90 kJ/mol (see Table 2.1.2). To illustrate the temperature effects on reaction rates, consider the dissolution rate of quartz relative to that of kaolinite. At 70 °C, the rate constant for kaolinite is 40 times larger than the quartz rate. If the temperature is increased to 130 °C, however, the kaolinite reaction becomes 5 times faster, while the rate of the quartz reaction increases by a factor of 100. Thus, at 130 °C, both rate constants are nearly equal.

2.1.7 Rates of Composite Reactions

The kinetic mass action law (2.1.2) is designed to work well for chemical reactions that occur in a single reaction step. However, dissolution and precipitation reactions are often composed of several substeps. For instance, the dissolution of calcite

$$CaCO_3 \rightarrow Ca^{2+} + {CO_3}^{2-}, \tag{2.1.7}$$

may for instance occur by the following sequence of intermediate substeps,

$$CaCO_3 + H^+ \rightarrow Ca^{2+} + {HCO_3}^- \tag{2.1.8}$$

$$ {HCO_3}^- \rightarrow H^+ + {CO_3}^{2-} \tag{2.1.9}$$

According to [47], this is the dominant dissolution mechanism in acid solutions with low CO_2 content. Here, H^+ acts like a catalyst to the composite reaction (2.1.7). Consequently, the concentration of H^+ will influence the reaction rate. Instead of including intermediate substeps in the mathematical model, one may choose to retain only the overall reaction formula, and add some extra terms to the reaction rate to account for catalysing species. The expression most commonly used for this purpose, is the one suggested by [58] (slightly simplified),

$$\frac{d\xi}{dt} = k\left(\prod_{i=1}^{q} a\left(T_i\right)^{t_i}\right)\left(1-\frac{Q}{K}\right). \tag{2.1.10}$$

Again, k is the rate constant, K is the equillibrium constant, and Q is the ion activity product. The species $T_1, \ldots, T_q$ are the ones that catalyse the reaction, and $t_1, \ldots, t_q$ are empirically determined powers.

To illustrate the use of formula (2.1.10), suppose that (2.1.8) is the rate determining step of the calcite dissolution reaction (2.1.7). A reasonable reaction rate model is then

$$\frac{d\xi}{dt} = k \cdot a\left(\mathrm{CaCO_3}\right) \cdot a\left(\mathrm{H^+}\right)\left(1 - \frac{a\left(\mathrm{CO_3}^{2-}\right) \cdot a\left(\mathrm{Ca^{2+}}\right)}{K \cdot a\left(\mathrm{CaCO_3}\right)}\right). \tag{2.1.11}$$

Here, the species participating in the overall reaction (2.1.7) are used to calculate the ion activity product Q, while the species on the left hand side of (2.1.8) are used to determine the catalytic factor. If we assume that the second step (2.1.9) is an equillibrium reaction, equation (2.1.11) is actually equivalent to the kinetic mass action law (2.1.2), applied to the rate determining step (2.1.8).

It is important to note that changes in the fluid composition and temperature can favor other intermediate reactions, causing a different set of species to become catalytic. For instance, if the aqueous CO_2 content is high, the reaction occurs by the mechanism

$$\begin{aligned} \mathrm{CaCO_3 + H_2O + CO_2(aq)} &\rightarrow \mathrm{Ca^{2+} + 2HCO_3}^- \\ \mathrm{HCO_3}^- &\rightarrow \mathrm{H^+ + CO_3}^{2-} \\ \mathrm{HCO_3^- + H^+} &\rightarrow \mathrm{H_2O + CO_2(aq)}. \end{aligned}$$

Here, CO_2(aq) is the major catalytic species. Often, rate laws for different mechanisms can be added to give an expression valid for broader ranges of temperature and fluid composition.

2.1.8 Compilations of Kinetic Data

To simulate a realistic scenario of geochemical processes, one needs thermodynamical and kinetic data for a range of different minerals. Compilations of such data are sometimes found in the literature, see Table 2.1.1 and Table 2.1.2 . Curiously, the kinetic data in these compilations may sometimes differ significantly, even when they are citing the same primary sources. For instance, the activation energy of the kaolinite dissolution reaction is reported to be 29.8 kJ/mol by [27], and 62.8 kJ/mol by [63]. While the source of the discrepancies remain unclear, they reflect the level of uncertainty underlying dissolution and precipitation models. When geochemistry

Reference	Name of simulator
Hellevang [20]	ATHENA
Xu et al. [63]	TOUGHREACT
Johnson et al. [27]	NUFT
Palandri [44]	(none)

Table 2.1.1: Compilations of kinetic data

is coupled to a transport solver, it is also common to use very simple expressions for the reaction rates (e.g., expression (2.1.10) without catalytic terms). This adds another layer of uncertainty. Thus, the reaction paths predicted from such models may not be entirely trustworthy. However, the end results are usually determined by equillibrium parameters, which are independent of the reaction model chosen. Equillibrium data can be calculated directly from the thermodynamical properties of the species involved, and are therefore more reliable that the kinetic parameters used in the rate laws. Parameters for equillibrium calcuations are also more readily available than kinetic data. For instance, [3] contains a comprehensive list of thermodynamic parameters for a range of different minerals.

2.2 Chemical Reaction Networks

We have now seen how the rate of an individual reaction may be calculated. However, the molar production rate of a component may be affected by a number of different chemical reactions at once. To illustrate, we consider a small reaction network,

$$A \to B \tag{2.2.1a}$$

$$2B \to C. \tag{2.2.1b}$$

Here, the species A, B and C are connected by two simple chemical reactions. Let r_1 and r_2 be the rates of the individual reactions, and R_A, R_B, R_C the production rates of the components. The reaction rates are the derivatives of the reaction progress variables, $\mathbf{r} = \frac{d}{dt}\boldsymbol{\xi}$, and the component production rates are the derivatives of the concentrations, $\mathbf{R} = \frac{d}{dt}\mathbf{c}$. The relation between them is linear, given by

$$\begin{aligned} R_A &= -r_1 \\ R_B &= r_1 - 2r_2 \\ R_C &= r_2. \end{aligned}$$

	$\rho/\frac{\mathrm{mol}}{\mathrm{dm}^3}$	$k_{max}/\frac{\mathrm{mol}}{\mathrm{yr\,dm}^3}$	$E_a/\frac{\mathrm{kJ}}{\mathrm{mol}}$	$\Delta H^{\ominus}/\frac{\mathrm{kJ}}{\mathrm{mol}}$	$\log_{10} K$
$\mathrm{Calcite + CO_2(aq) + H_2O \rightleftarrows Ca^{2+} + 2HCO_3^-}$	27.08	equillibrium	-	-16.0	-4.50
$\mathrm{Magnesite + CO_2(aq) + H_2O \rightleftarrows Mg^{2+} + 2HCO_3^-}$	35.58	3.16×10^{-1}	62.8	-34.8	-4.05
$\mathrm{Siderite + CO_2(aq) + H_2O \rightleftarrows Fe^{2+} + 2HCO_3^-}$	34.18	3.16×10^{-1}	62.8	-22.8	-6.54
$\mathrm{Dawsonite + 3H^+ \rightleftarrows Na^+ + Al^{3+} + HCO_3^- + 2H_2O}$	16.81	2.36×10^{0}	62.8	-68.6	3.66
$\mathrm{Albite + 4H^+ \rightleftarrows Na^+ + Al^{3+} + 3SiO_2(aq) + 2H_2O}$	9.99	2.76×10^{-3}	80.3	-44.1	2.08
$\mathrm{K - Feldspar + 4H^+ \rightleftarrows K^+ + Al^{3+} + 3SiO_2(aq) + 2H_2O}$	9.20	6.24×10^{-4}	51.7	-16.2	-0.96
$\mathrm{Quartz \rightleftarrows SiO_2(aq)}$	43.77	1.88×10^{-5}	87.7	32.9	-4.00
$\mathrm{Chalcedony \rightleftarrows SiO_2(aq)}$	43.27	1.36×10^{-2}	87.7	31.4	-3.73
$\mathrm{Kaolinite + 6H^+ \rightleftarrows 2Al^{3+} + 2SiO_2(aq) + 5H_2O}$	10.07	1.58×10^{-2}	29.0	-136.3	5.44
$\mathrm{Clinochlore + 16H^+ \rightleftarrows 5Mg^{2+} + 2Al^{3+} + 3SiO_2(aq) + 12H_2O}$	4.77	1.18×10^{-2}	88.0	-596.9	65.87
$\mathrm{Daphnite + 16H^+ \rightleftarrows 5Fe^{2+} + 2Al^{3+} + 3SiO_2(aq) + 12H_2O}$	4.49	1.18×10^{-2}	88.0	-492.1	49.78
$\mathrm{Muscovite + 10H^+ \rightleftarrows K^+ + 3Al^{3+} + 3SiO_2(aq) + 6H_2O}$	7.10	3.94×10^{-3}	22.0	-220.0	11.53
$\mathrm{Phlogopite + 10H^+ \rightleftarrows K^+ + Al^{3+} + 3Mg^{2+} + 3SiO_2(aq) + 6H_2O}$	6.71	1.58×10^{-2}	29.0	-302.8	36.76
$\mathrm{Annite + 10H^+ \rightleftarrows K^+ + Al^{3+} + 3Fe^{2+} + 3SiO_2(aq) + 6H_2O}$	7.91	1.58×10^{-2}	29.0	-252.2	28.79
$\mathrm{Labradorite} + \frac{32}{5}\mathrm{H^+} \rightleftarrows \frac{2}{5}\mathrm{Na^+} + \frac{3}{5}\mathrm{Ca^{2+}} + \frac{8}{5}\mathrm{Al^{3+}} + \frac{12}{5}\mathrm{SiO_2(aq)} + \frac{16}{5}\mathrm{H_2O}$	9.93	5.19×10^{-1}	42.1	-219.9	21.14
$\mathrm{Gibbsite + 3H^+ \rightleftarrows Al^{3+} + 3H_2O}$	30.13	8.64×10^{-2}	53.0	-95.1	6.97

The rate constant k is calculated as $k = k_{max}V$, where V is the mineral's volume fraction. It is assumed that none of the aqueous species act catalytically. All parameters are reported at a reference temperature of $T^{\ominus} = 298.15\,\mathrm{K}$.

Table 2.1.2: Kinetic parameters used in the ATHENA simulator

Using matrix notation, we can express this as

$$\begin{bmatrix} R_A \\ R_B \\ R_C \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & -2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix},$$

or simply

$$\mathbf{R} = \mathbb{S}^\top \mathbf{r}. \tag{2.2.2}$$

The matrix $\mathbb{S}$ is called the *stoichiometry matrix.* Each row in this matrix represents a reaction, each column a species. Note how the stoichiometric coefficients of (2.2.1) reappear in the matrix. For every row, negative entries correspond to reactants, positive entries to products, and the zeros are components that do not participate in the reaction.

2.2.1 Reaction Paths for Closed Systems

For a closed chemical system (i.e., no fluid transport), the evolution of the components participating in a reaction network is given by the differential equation

$$\frac{d\mathbf{c}}{dt} = \mathbb{S}^\top \mathbf{r}(\mathbf{c}). \tag{2.2.3}$$

Here, the number of equations equals the number of components. The solution of this equation is called a *reaction path.* For very simple rate expressions only, reaction paths can be found analytically. As an example, we consider the system

$$A \to C \tag{2.2.4a}$$

$$B \to C \tag{2.2.4b}$$

with the reaction rates

$$\begin{aligned} r_1 &= k_1 \left(c_A - c_C/K_1 \right) \\ r_2 &= k_2 \left(c_B - c_C/K_2 \right), \end{aligned}$$

where K_1, K_2 are equillibrium constants, and k_1, k_2 are kinetic constants. This gives a linear system of equations,

$$\frac{d}{dt} \begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix} = \begin{bmatrix} -k_1 & 0 & k_1/K_1 \\ 0 & -k_2 & k_2/K_2 \\ k_1 & k_2 & -\frac{k_1}{K_1} - \frac{k_2}{K_2} \end{bmatrix} \begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix},$$

which can be solved by standard methods of eigenvalue decomposition. For instance, if $K_1 = \frac{1}{2}$, $K_2 = \frac{1}{3}$, and $k_1 = k_2 = 1$, the solution to the equation is

$$\begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix} = \frac{1}{6}\left(c_{A0} + c_{B0} + c_{C0}\right)\cdot \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} + \frac{e^{-6t}}{30}\left(-c_{A0} - c_{B0} + 5c_{C0}\right)\cdot \begin{bmatrix} -2 \\ -3 \\ 5 \end{bmatrix} + \frac{e^{-t}}{5}\left(-3c_{A0} + 2c_{B0}\right)\cdot \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix},$$

where c_{A0}, c_{B0} and c_{C0} are the initial concentrations of species A, B and C, respectively. We see that the first fundamental solution is the equillibrium solution, which is approached as $t \to \infty$. The other fundamental solutions correspond to two composite reactions,

$$2A + 3B \xrightarrow{2r_1+3r_2} 5C \tag{2.2.5a}$$

$$A \xrightarrow{r_1-r_2} B, \tag{2.2.5b}$$

that decouple the system. Both of these are linear combinations of the original two reactions, as indicated, and neither of them influences the reaction rate of the other. Reaction (2.2.5a) is 6 times faster than (2.2.5b), as evident from the eigenvalues. A dimensionless plot of the solution for different initial values is shown in Figure 2.2.1. For both sets of initial conditions, we see that the solution is dominated by the fast reaction at first. Eventually, the slower reaction begins to influence the system, which then gradually approaches equillibrium.

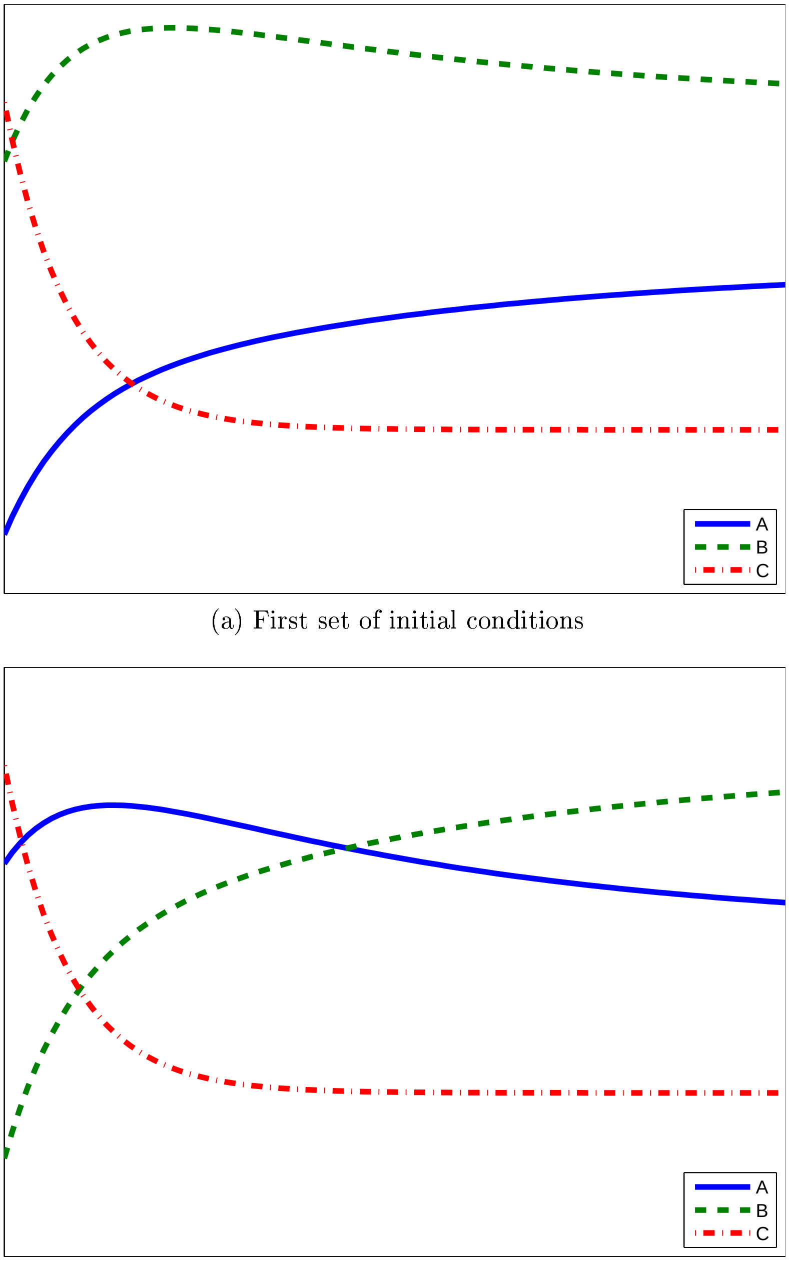
2.2.2 The Thermal Reactive Term

Recall that the temperature conservation equation (1.4.3) contains a reactive term of the form

$$\sum_{i=1}^{m} e_i R_i,$$

where e_i is the specific internal energy for the component i, and R_i is the production rate of i due to chemical reactions. Using vector notation and equation (2.2.2), this term can be rewritten as

$$\begin{aligned}\sum_{i=1}^{m} e_i R_i &= \mathbf{e} \bullet \mathbf{R} \\ &= \left(\mathbb{S}\mathbf{e}\right)^{\top} \mathbf{r}.\end{aligned}$$



(a) First set of initial conditions

(b) Second set of initial conditions

Figure 2.2.1: Sample reaction paths for (2.2.4)

Here, the term $\mathbb{S}\mathbf{e}$ is actually equivalent to the vector of reaction energies. Thus, for a specific reaction j, we may set

$$(\mathbb{S}\mathbf{e})_j = \Delta E_j \approx \Delta E_j^{\ominus} \approx \Delta H_j^{\ominus}.$$

While the reaction energy ΔE is not precisely equal to the standard reaction enthalpy $\Delta H^{\ominus}$, the difference is small. At most, they are of the same magnitude as the dissipation and compression terms which we have already excluded from (1.4.3). Therefore, it is common to simply use $\Delta H^{\ominus}$, which is readily available in thermodynamical tables.

2.2.3 Reaction Paths for Open Systems

We now turn to the question of how to couple chemical reactions with fluid transport. Recall from Section 1.3 that the mass conservation of a species i is given by

$$\frac{\partial c_i}{\partial t} = \mathcal{L}\left(p,\, T,\, c_i\right) + R_i\left(p,\, T,\, c_1,\, \ldots,\, c_m\right),$$

where we have omitted the external source term for simplicity, and expressed dispersion and advection abstractly by the transport operator $\mathcal{L}$. If we assume that the mass conservation equation is decoupled from the temperature and pressure equations by operator splitting, as depicted in Section 1.5, it suffices to indicate the concentration dependence only,

$$\frac{\partial c_i}{\partial t} = \mathcal{L}\left(c_i\right) + R_i\left(c_1,\, \ldots,\, c_m\right).$$

Using vector notation, this is equivalent to

$$\frac{\partial \mathbf{c}}{\partial t} = \boldsymbol{\mathcal{L}}\left(\mathbf{c}\right) + \mathbb{S}^{\top}\mathbf{r}\left(\mathbf{c}\right). \tag{2.2.6}$$

Alternatively, one can solve the temperature and mass conservation equations simultaneously, which will give a combined equation of roughly the same structure as (2.2.6). In that case, $\boldsymbol{\mathcal{L}}$ will be expanded to include the temperature conduction and convection terms, and $\mathbb{S}$ will include an extra column corresponding to the reaction enthalpies.

To illustrate how fluid flow may interact with chemical reactions, let us once again consider a simple example that is analytically solvable. Specifically, let us calculate

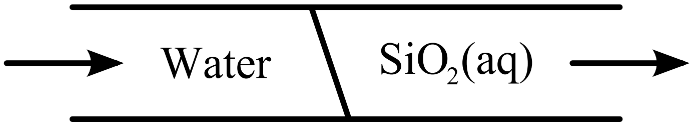


Figure 2.2.2: Flushing of a quartz sample

the concentration of SiO$_2$(aq) within a one-dimensional, quartz dominated core sample, as the sample is flushed with pure water at a constant rate (Figure 2.2.2). The mineral dissolution reaction is simply

$$\text{Quartz} \to \text{SiO}_2(\text{aq}),$$

with the reaction rate

$$r = k\left(1 - a/K\right),$$

where a is the activity of SiO$_2$(aq), K is the equillibrium constant, and k is the kinetic rate constant. Recall that the activity is given by

$$a = \phi\gamma c,$$

where ϕ the porosity, γ is the activity coefficient, and c the bulk concentration. The equillibrium concentration of SiO$_2$(aq) is thus readily seen to be

$$c_{eq} = \frac{K}{\phi\gamma}.$$

Let us assume that ϕ and γ are constants, as well as the darcy velocity u. The evolution of c is then given by

$$\frac{\partial c}{\partial t} = -\frac{u}{\phi}\frac{\partial c}{\partial x} + \frac{k}{c_{eq}}\left(c_{eq} - c\right).$$

If we assume that SiO$_2$(aq) is in equillibrium with quartz initially, the analytical solution is given by the method of characteristics,

$$\frac{c(t,\, x)}{c_{eq}} = \begin{cases} 1 - \exp\left(-\frac{k\phi}{c_{eq}u}x\right) & \text{if } x\phi < ut \\ 1 & \text{if } x\phi > ut. \end{cases}$$

The dimensionless parameter $D = \frac{k\phi L}{c_{eq}u} = \frac{k\phi^2\gamma L}{Ku}$, where L is the length of the sample, determines the shape of the solution. This parameter describes the time scale of the advection relative to reaction, and is often called the *Damköhler number.*

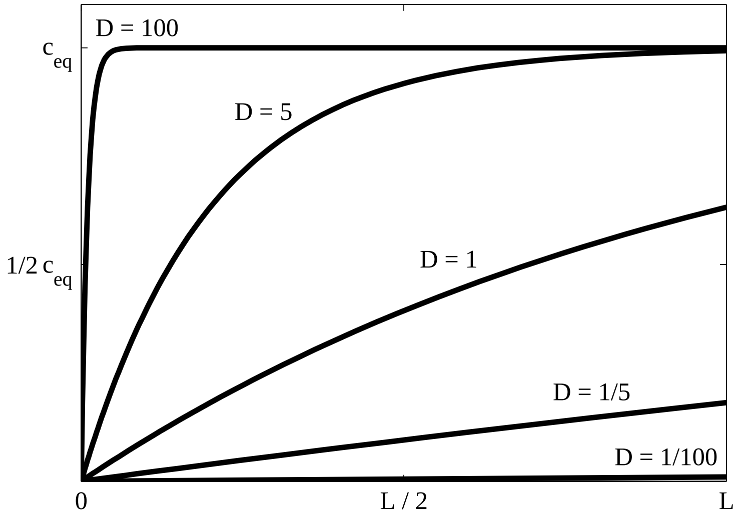


Figure 2.2.3: Flushing of a quartz sample

For the quartz dissolution reaction, D is strongly dependent on temperature. To illustrate, let us assume that $\phi = 0.25$, $\gamma = 1$, $L = 1\,\mathrm{m}$ and $u = 1\,\mathrm{cm/day}$. Then, we can use the thermodynamic parameters in Table 2.1.2 to find D in the range 50 °C to 300 °C:

D	T
0.013	50 °C
0.21	100 °C
1.7	150 °C
9.0	200 °C
103	300 °C

Table 2.2.1: Damköhler numbers for the flushing of quartz

A plot of the final-state concentration distribution for this range of Damköhler numbers is shown in Figure 2.2.3. For high temperatures, D is large, and the process is dominated by chemistry. Consequently, most of the fluid is in chemical equillibrium with the quartz. With decreasing temperatures, D decreases as well, and the process becomes dominated by advection instead. In this case, the dissolution rate is not sufficiently fast to replenish the $\mathrm{SiO_2(aq)}$ that is flushed out by water injection, and the sample becomes filled with pure water instead.

Chapter 3

Stiffness of Reactive Systems

In Section 2.2.1, we calculated the reaction path for a simple reaction network. During the solution procedure, we identified two different time scales that characterised the system, one six times larger than the other. For realistic chemical systems, the slowest and fastest solution modes often differ by many orders of magnitude, a property called *stiffness*. It is well-known that stiff systems are challenging to solve numerically, and that implicit integration methods must be used to gain sufficient numerical stability. In this chapter, we will try to give the reader an intuitive feeling for the kind of difficulties that may arise. Furthermore, we will discuss a class of Runge-Kutta methods that are well suited to cope with stiffness.

3.1 Sources of Stiffness

For a differential equation system of the form

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}),$$

the time scales of the solution can be identified by considering the eigenvalues of the system's *Jacobian matrix*,

$$\mathbb{J} = \frac{\partial \mathbf{f}}{\partial \mathbf{y}}.$$

Specifically, the ratio of the largest to the smallest time scale is the ratio of the largest to smallest nonzero eigenvalue[1] of $-\mathbb{J}$. This quantity is called the *stiffness*

[1] If any of the eigenvalues are negative or complex, absolute values are used.

ratio of the system. In this section, we will be concerned with the stiffness ratio of the mass conservation equation. For nonreactive models, the equation is nonstiff as long as no diffusion is included in the model. Even with diffusion, the equation is only mildly stiff, and can be solved by stabilised explicit approaches like the Runge-Kutta-Chebyshev[60] methods. When chemistry is introduced, the stiffness quickly becomes so severe that implicit methods are the only viable solution alternative. In general, there are four main chemistry-related factors that may contribute to a large stiffness ratio. We will now describe each of these separately, and illustrate their effect by using analytically solvable examples.

3.1.1 Rate Constants

If a chemical system includes reactions with very different rate constants, this will obviously influence the dominating time scales of the system. To illustrate this, consider a closed system (i.e., no mass transport) of three components A, B and C, connected through the chemical reactions

$$\begin{aligned} A &\rightarrow B \\ B &\rightarrow C. \end{aligned}$$

Let the reaction rates be given by

$$\begin{aligned} r_1 &= k_1\left(c_A - c_B\right) \\ r_2 &= k_2\left(c_B - c_C\right). \end{aligned}$$

Here, c_A, c_B and c_C are the concentrations of each component. The equillibrium constants of both reactions are 1, while the reaction rates are given by the parameters k_1 and k_2. Since no mass transport is involved, the concentrations change with time according to

$$\frac{d}{dt}\begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix} = \begin{bmatrix} -k_1 & k_1 & 0 \\ k_1 & -k_1 - k_2 & k_2 \\ 0 & k_2 & -k_2 \end{bmatrix} \begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix}.$$

The Jacobian of this system is singular, so one of the eigenvalues is 0. The other two are given by

$$\begin{aligned} -\lambda_1 &= k_1 + k_2 - \sqrt{{k_1}^2 - k_1 k_2 + {k_2}^2} \\ -\lambda_2 &= k_1 + k_2 + \sqrt{{k_1}^2 - k_1 k_2 + {k_2}^2} \end{aligned}$$

If $k_1 = k_2$, we have $\lambda_{\max}/\lambda_{\min} = 3$, and the system is nonstiff. However, if $k_1 \gg k_2$, the stiffness ratio is approximately $2\frac{k_1}{k_2} + 1$, which may be arbitrarily large. This shows that differing rate constants can indeed be a source of stiffness in a reactive system. As evident from Table 2.1.2, this is a commonly occuring situation when simulating geochemical reactions.

3.1.2 Equillibrium Concentrations

The equillibrium constants determine the ratio between species concentrations when chemical equillibrium has been attained. Stiffness can be introduced to the system if the magnitude of these concentrations are very different. We can show this by considering a closed three-component system once again. This time, we define the reactions

$$\begin{aligned} A &\to C \\ B &\to C, \end{aligned}$$

with the reaction rates

$$\begin{aligned} r_1 &= c_A - c_C/K_1 \\ r_2 &= c_B - c_C/K_2. \end{aligned}$$

Here, the rate constants are both equal to 1, while the equillibrium constants are given by K_1 and K_2. The evolution of the concentrations is given by

$$\frac{d}{dt}\begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1/K_1 \\ 0 & -1 & 1/K_2 \\ 1 & 1 & -\frac{1}{K_1} - \frac{1}{K_2} \end{bmatrix} \begin{bmatrix} c_A \\ c_B \\ c_C \end{bmatrix},$$

and the nonzero eigenvalues of the system are simply

$$\begin{aligned} -\lambda_1 &= 1 \\ -\lambda_2 &= \frac{1}{K_1} + \frac{1}{K_2} + 1. \end{aligned}$$

Thus, if the equillibrium concentration of C is small compared to either A or B, the eigenvalue ratio is large, and the system may be stiff. Large differences in equillibrium concentrations is a commonly occuring situation. For instance, in an aqueous solution saturated with CO_2, the equillibrium concentration of CO_2(aq) is about 10 times as high as the concentration of CO_3^{2-}. Despite this large concentration difference, is is important to retain both species in the model, as the reaction rates of the system are greatly influenced by changes in their concentrations.

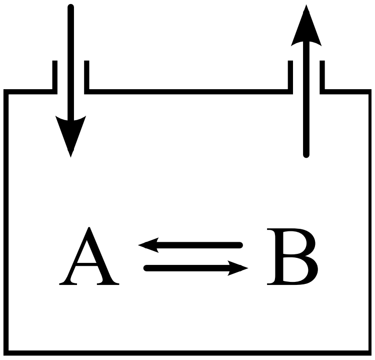


Figure 3.1.1: A simple reaction-advection system

3.1.3 Advection

If a reaction system is open, that is, if the system can exchange mass with the surroundings, yet another time scale is introduced. We illustrate this by considering a simple reaction between two species,

$$A \to B,$$

occuring in a tank that is flushed with pure water (see Figure 3.1.1). The reaction rate is given by

$$r = k\left(c_A - c_B\right).$$

Despite this being a zero-dimensional system, it gives rise to a system of ordinary differential equations that resemble a spatial discretisation of the mass conservation equation. If u is the volumetric outflow rate, scaled by the tank volume, the concentrations in the tank are described by the equation

$$\frac{d}{dt}\begin{bmatrix} c_A \\ c_B \end{bmatrix} = \begin{bmatrix} -u-k & k \\ k & -u-k \end{bmatrix}\begin{bmatrix} c_A \\ c_B \end{bmatrix}.$$

This time, the eigenvalues are given by

$$\begin{aligned} -\lambda_1 &= u \\ -\lambda_2 &= u + 2k. \end{aligned}$$

The eigenvalue ratio is $2\frac{k}{u} + 1$, which is large if the reaction rate is much faster than the flow rate. This is a commonly occuring situation as well. For instance, gaseous CO_2 that has gathered beneath an impermeable cap rock, may dissolve into the aquifer brine below, and set up slowly migrating convective currents. In this context, the rate of many mineral reactions will be much faster than the flow rate.

3.1.4 Nonlinearity

The last source of stiffness we will demonstrate, is the effect of nonlinear reaction rates. For this purpose, we once again consider the system depicted in Figure 3.1.1. This time, we let the reaction between them be slightly different,

$$A \to 2B,$$

with the reaction rate

$$r = k\left(c_A - {c_B}^2/K\right).$$

Also, we let the tank be flushed by a solution where the concentrations of A and B are in equillibrium. Then the concentrations in the tank will eventually approach the concentrations of the incoming fluid. We already know that stiffness may be introduced to the system if equillibrium concentrations are widely different. To eliminate this effect, we set $c_A = c_B = K$ at the inflow boundary.

With these assumptions, concentrations evolve according to

$$\frac{d}{dt}c_A = -u\left(c_A - K\right) - k\left(c_A - {c_B}^2/K\right) \tag{3.1.1a}$$

$$\frac{d}{dt}c_B = -u\left(c_B - K\right) + 2k\left(c_A - {c_B}^2/K\right). \tag{3.1.1b}$$

Since the reaction rate is nonlinear, the Jacobian of the system is no longer a constant matrix. Thus, the characteristic time scales of the system are dependent on fluid composition, and may also change with time. Indeed, the Jacobian is given by

$$\mathbb{J} = \begin{bmatrix} -u-k & 2kc_B/K \\ 2k & -u-4kc_B/K \end{bmatrix},$$

and its eigenvalues are

$$\begin{aligned} -\lambda_1 &= u \\ -\lambda_2 &= k + u + 4kc_B/K. \end{aligned}$$

Assuming that the time scales of reaction and advection are equal (i.e., $u = k$), the stiffness ratio is $2 + 4c_B/K$. As the solution is approaching equillibrium ($c_B \to K$), the ratio approaches 6, and the equation is nonstiff. However, if the original concentration of B in the tank is much larger than K, the equation may be stiff during the first part of the solution period. In a simulation of CO_2 migration, this situation frequently occurs, since the equillibrium concentrations of many aqueous species change by orders of magnitude if a reservoir region is flushed with CO_2.

3.2 Stiff Instability

To demonstrate why explicit integration methods have trouble integrating stiff equations, we try to solve an advection-reaction equation numerically. Consider once again the system shown in Figure 3.1.1, with the reaction rate given in Section 3.1.4. Furthermore, assume that the inflow concentrations of both species are zero, and that the initial tank concentrations are both equal to K. Thus, the main source of stiffness in our system is the relation between the reactive and advective time scale.

Scaling the concentrations by their initial values, we get the following system of differential equations,

$$\frac{d}{dt}c_A = -uc_A - k\left(c_A - {c_B}^2\right) \tag{3.2.1a}$$

$$\frac{d}{dt}c_B = -uc_B + 2k\left(c_A - {c_B}^2\right). \tag{3.2.1b}$$

Furthermore, we choose an integration interval of $[0, \frac{1}{u}]$, which matches the time scale of the advection. Although we can not solve the system analytically, it can be partially solved by observing that

$$\frac{d}{dt}\left(2c_A + c_B\right) = -u\left(2c_A + c_B\right),$$

which has the exact solution

$$2c_A + c_B = 3e^{-ut}.$$

This relation can be substituted into (3.2.1), and we are left to solve either of the one-dimensional equations

$$\frac{d}{dt}c_A = -uc_A - k\left(c_A - \left(3e^{-ut} - 2c_A\right)^2\right) \tag{3.2.2a}$$

$$\frac{d}{dt}c_B = -uc_B + k\left(3e^{-ut} - c_B - 2{c_B}^2\right). \tag{3.2.2b}$$

3.2.1 Explicit Euler Method

As there are no simple ways of solving (3.2.2) analytically, we must proceed by numerical means. First, we try to solve the equation using the explicit Euler method.

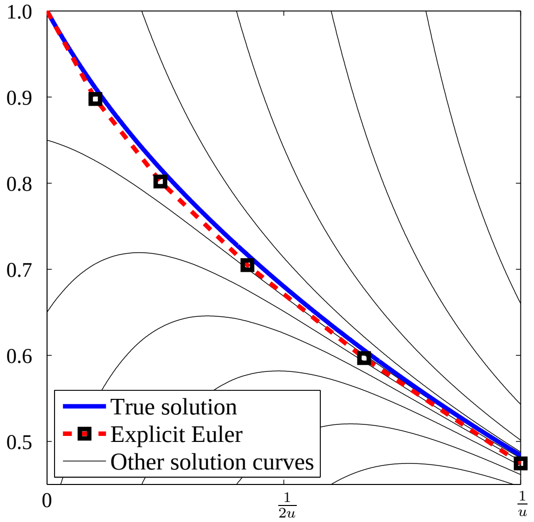


Figure 3.2.1: Explicit solution with $u = k$

For a differential equation $\frac{d}{dt}\mathbf{y} = \mathbf{f}(\mathbf{y},\, t)$, one explicit Euler step of length Δt is given by

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\, \mathbf{f}\left(\mathbf{y}_n,\, t_n\right).$$

For each time step, the algorithm finds the next function value by a linear extrapolation, based on the current slope. The resulting step value will not lie on the correct solution curve, since the true solution is curving. Nevertheless, if the equation is nonstiff, the slopes of neighbouring solution curves are almost the same as for the true solution. Thus, the error stays bounded throughout the integration interval.

In figure 3.2.1, Equation (3.2.2b) is solved with $k = u$, using 5 explicit Euler steps. The true solution and neighbouring solution curves are also shown. In this case, the reaction rates are of the same magnitude as the advection velocity, and the equation is nonstiff. We clearly see that the slopes of neighbouring curves are similar to that of the true solution, and the method therefore experiences no problems. If we increase the reaction rate, however, the slopes are changing much more rapidly, and we may be forced to reduce the step length to avoid instabilities. This is clearly seen in Figure 3.2.2, where we have solved the equation for $k = 10\,u$, using increasingly larger steps. As long as the step length is chosen to match the smallest time scale of the system (the reaction scale), the explicit method integrates the equation accurately. But if the step length is increased slightly above this level, the computed solution will

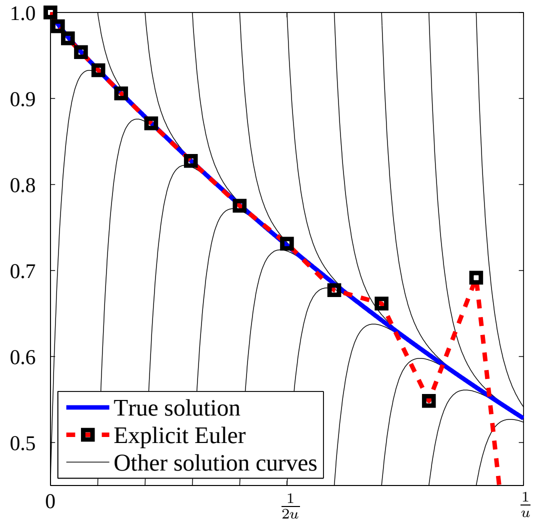


Figure 3.2.2: Explicit solution with $k = 10u$

quickly diverge in an oscillating fashion.

The argument presented above applies to higher-order explicit methods as well, as they are all based on extrapolation of some kind. We note that problems with strict accuracy requirements have to be integrated with small time steps regardless of the problem's stiffness ratio. Therefore, large eigenvalue differences is only a problem when error tolerances are crude, and when the integration interval is determined by the magnitude of the largest eigenvalue.

3.2.2 Implicit Euler Method

Now, let us test the performance of the implicit Euler method on (3.2.2b). Recall that one implicit Euler step is given by

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \Delta t\, \mathbf{f}\left(\mathbf{y}_{n+1},\, t_{n+1}\right),$$

when applied to an equation of the form $\frac{d}{dt}\mathbf{y} = \mathbf{f}(\mathbf{y},\, t)$. Since the step value $\mathbf{y}_{n+1}$ appears on both sides of this equation, it must be found using an iterative algorithm designed for nonlinear equations. Roughly speaking, the implicit Euler method tries to find a step value $\mathbf{y}_{n+1}$ such that a backward linear extrapolation from $\mathbf{y}_{n+1}$ is equal

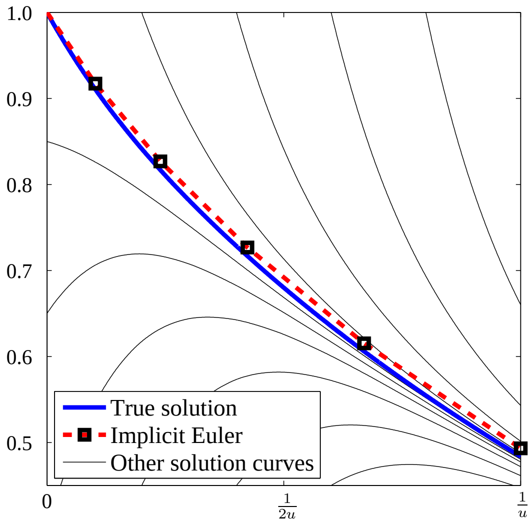


Figure 3.2.3: Implicit solution with $k = u$

to the value $\mathbf{y}_n$ at the beginning of the time step. For nonstiff equations, the algorithm requires just as many time steps as its explicit counterpart, as shown in 3.2.3 . Since an implicit time step is computationally costly, the explicit method clearly has the best performance in this case. However, this figure changes dramatically with increasing stiffness. While stiffness restricts the time steps of explicit methods, it permits implicit methods to take even larger steps. For instance, if $k = 10\,u$, the equation is solved with acceptable accuracy by a single implicit step, as shown in Figure 3.2.4.

3.2.3 Linear Stability Analysis

To explain the difference in stability properties for the implicit and explicit Euler method, we will make use of the differential equation

$$\frac{dy}{dt} = \lambda y, \quad y_0 = 1, \tag{3.2.3}$$

known as *Dahlquist's test equation.* This equation is often used to measure the stability performance of a numerical method, since it is simple to analyse. Furthermore, the performance on (3.2.3) carries on to more complicated problems as well. All

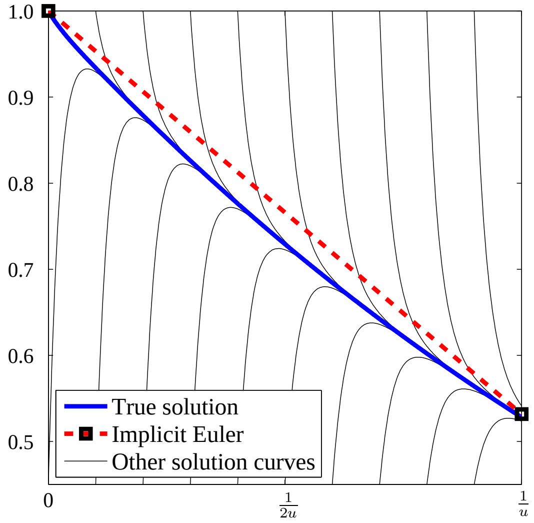


Figure 3.2.4: Implicit solution with $k = 10u$

multidimensional linear equations can, for instance, be decoupled into several one-dimensional test equations. For each of these, the parameter λ will correspond to an eigenvalue of the coefficient matrix. Similarly, nonlinear equations may be locally decoupled by the eigenvalues of the system's Jacobian matrix. Many of the stability results on (3.2.3) are therefore valid for the nonlinear case as well. One must, however, be aware of certain phenomena that complicate the analysis of nonlinear problems. Finally, we note that non-autonomous differential equations can be cast into autonomous form by introducing an extra variable $\frac{\partial u}{dt} = 1$ to the system. This allows them to be analysed within the same framework as autonomous equations.

Now, consider an arbitrary one-step method, applied to the test equation. Let y_1 be the computed value after one time step of length Δt. It turns out that y_1 is dependent on the dimensionless, complex-valued quantity $z = \lambda \Delta t$ only. Thus we can write $y_1 = R(z)$, which is called the *stability function*. For the exact analytical solution operator, given by $R(z) = e^z$, we have $|R(z)| < 1$ whenever the real part of λ is negative. This is not always the case for a numerical method.

Let us calculate the stability function of Euler's method, which performed badly on the stiff equation. One time step, applied to the test equation, is given by

$$y_1 = 1 + \lambda \Delta t,$$

so we have

$$R(z) = 1 + z.$$

Here, we see that a λ with a large negative real part may actually cause the numerical solution to grow, even if the true solution is decaying. To ensure stability, we must require $|1 + z| < 1$. This is exactly the phenomenon we observed in Section 3.2.1. There, we tried to solve an equation where the magnitude of one eigenvalue was large compared to the integration interval. This forced the time step to be reduced below the accuracy requirement level, to avoid exponential error growth.

In Section 3.2.2, we observed that the implicit Euler method integrated the stiff equation without problems. If we apply the method to the test equation, we get

$$y_1 = 1 + \lambda y_1 \Delta t,$$

and the stability function is readily seen to be

$$R(z) = \frac{1}{1 - z}.$$

Here,$|R(z)| < 1$ for all z with negative real part, just like the exact solution operator. That is, if the true solution is decaying, the numerical solution will decay as well. The step length can therefore be determined by the required accuracy, instead of stability considerations. This is the reason of why the implicit method performed so well.

The region region of the complex plane for which $|R(z)| < 1$, is called the *region of absolute stability* for the numerical method. The analysis above shows that the explicit Euler method has a bounded stability region, which is why it performs badly on stiff equations. On the other hand, the stability region of the implicit Euler method covers the entire left half of the complex plane. This property, called *A-stability*, allows the implicit method to use large time steps, even though some of the system's eigenvalues are large compared to the integration interval.

Often, A-stability is not sufficient to ensure proper behaviour when applied to stiff equations. For instance, consider the stability function of the trapezoidal method

$$R(z) = \frac{1 + \frac{1}{2}z}{1 - \frac{1}{2}z},$$

which is readily seen to satisfy the stability criterion for all z values with a negative real part. However, the trapezoidal method has the undesirable property that

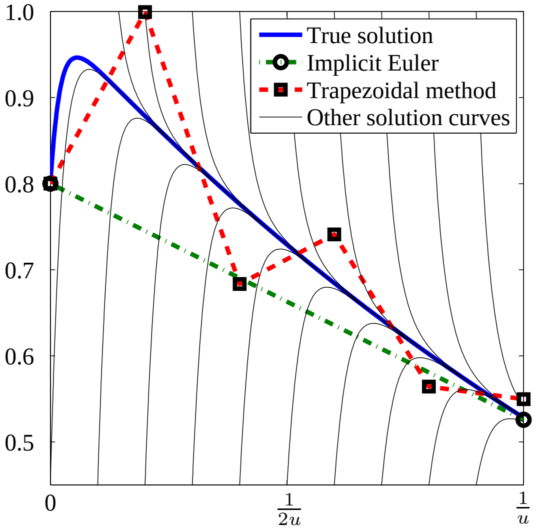


Figure 3.2.5: Trapezoidal and implicit Euler method with $k = 10u$

$|R(z)| \to 1$ as $z \to \infty$. Thus, solution components with a large z value are damped out very slowly. Figure 3.2.5 shows the trapezoidal method applied to (3.2.2b), with a step length of $\frac{1}{5u}$, and with an initial value slightly below the equillibrium value. We see that the numerical errors decay very slowly, since $R(z)$ is close to 1 for the fastest solution component. On the other hand, the stability function of the implicit Euler method satisfies $\lim_{z\to\infty} |R(z)| = 0$, just like the exact solution operator. Methods with this property, that are also A-stable, are called *L-stable*. As evident from Figure 3.2.5, the L-stable Euler method has no problems integrating past the initial transient solution.

3.3 Implicit Runge-Kutta Methods

While the implicit Euler method is very robust, it is only first-order accurate with respect to the step length. Fortunately, it is possible to derive higher-order integration methods that are equally well suited to handle stiff equations. If the integration interval is long, the solution smooth, and the error tolerances strict, methods of the multistep type give the best performance[18]. For this reason, multistep methods are popular for simulating batch reactor systems, which are reactive systems that do not involve fluid flow. For instance, the widely used batch reactor simulators KINSIM[4], COPASI[24] and CHEMKIN[28] are all based on multistep integration methods.

However, these methods may not be the best choice for complex reaction problems involving flow. As such problems are usually solved by operator splitting, the integration intervals are small, and the splitting errors destroy the high accuracy of multistep methods. A better alternative is to use one-step methods, which we will consider in this section. Specifically, we will focus on implicit Runge-Kutta methods, which have been successfully applied to other reaction-convection problems[56, 35, 30]. Other alternatives include the Rosenbrock methods, which have been recommended for certain atmospheric chemistry problems[55], or the extrapolation methods[18].

3.3.1 Definitions

In general, a Runge-Kutta method is an integration rule that can be written in the form

$$\begin{aligned}
\mathbf{g}_1 =& \mathbf{y}_n + \Delta t \sum a_{1j} \mathbf{f}\left(\mathbf{g}_j\right) \\
\mathbf{g}_2 =& \mathbf{y}_n + \Delta t \sum a_{2j} \mathbf{f}\left(\mathbf{g}_j\right) \\
&\vdots \\
\mathbf{g}_N =& \mathbf{y}_n + \Delta t \sum a_{Nj} \mathbf{f}\left(\mathbf{g}_j\right) \\
\mathbf{y}_{n+1} =& \mathbf{y}_n + \Delta t \sum b_j \mathbf{f}\left(\mathbf{g}_j\right),
\end{aligned}$$

when applied to an ordinary differential equation

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}).$$

In (3.3.1), $\mathbf{g}_1, \ldots, \mathbf{g}_N$ are called *internal stages.* The coefficients used in (3.3.1) are specific to each Runge-Kutta method. Often, the coefficients are displayed in a so-called *Butcher tableau,*

$$\begin{array}{c|ccc}
c_1 & a_{11} & \cdots & a_{1N} \\
\vdots & \vdots & & \vdots \\
c_N & a_{N1} & \cdots & a_{NN} \\
\hline
 & b_1 & \cdots & b_N
\end{array} \quad ,$$

where the leftmost column is defined by $c_i = a_{i1} + \ldots + a_{iN}$. The Butcher tableaus of several different Runge-Kutta methods are shown in Table 3.3.1.

3.3.2 Newton's Method

For some of the schemes in Table 3.3.1, the upper triangular part of the tableau is zero, which means that all the internal stages in (3.3.1) can be calculated explicitly. The other methods are implicit, and must be coupled with an iterative nonlinear equation solver. Hairer and Wanner[18] points out that the fix-point iteration algorithm, which is commonly used to solve nonlinear equations, is not suited to solve (3.3.1) when the Runge-Kutta method is applied to stiff systems. The reason is that this destroys the stability properties of the method, which is the reason of why we want to use an implicit method in the first place. Instead, they recommend the use of Newton's method to find the internal stage values.

Recall that one Newton iteration, applied to a nonlinear equation $\mathbf{F}(\mathbf{x}) = \mathbf{0}$, improves an approximate solution $\mathbf{x}_n$ by solving the equation

$$\left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\mathbf{x}_n} (\mathbf{x}_{n+1} - \mathbf{x}_n) = -\mathbf{F}(\mathbf{x}_n)\,. \tag{3.3.2}$$

Here, $\mathbf{x}_{n+1}$ is the improved solution estimate, and $\partial\mathbf{F}/\partial\mathbf{x}$ is the Jacobian matrix. If the initial solution estimate $\mathbf{x}_0$ is close to the true solution, the convergence of Newton's method is of second order. When applied to the system (3.3.1), it is reasonable to choose $\mathbf{y}_n$ as an initial guess for the stage values, as these are known to be close when the time step is small. If this initial guess is to far from the true solution, the step length Δt can be reduced until convergence is acheived.

If the number of variables in (3.3.2) is very large, one Newton iteration requires the solution of a large system of linear equations, which is computationally costly. In this case, it is better to employ the *simplified* Newton's method, which is defined by the iteration formula

$$\left.\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} (\mathbf{x}_{n+1} - \mathbf{x}_n) = -\mathbf{F}(\mathbf{x}_n)\,. \tag{3.3.3}$$

The only difference from (3.3.2), is that the Jacobian is evaluated at the initial value only. The method still converges, provided that $\mathbf{x}_0$ is close to the true solution, but the convergence is only linear. On the other hand, the computationally demanding

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}$$

(a) Explicit Euler

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}$$

(b) Implicit Euler

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & \frac{1}{2} & \frac{1}{2} \\ \hline & \frac{1}{2} & \frac{1}{2} \end{array}$$

(c) Trapezoidal rule

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} \end{array}$$

(d) Classical RK4

$$\begin{array}{c|ccc} \beta & \beta & 0 & 0 \\ \frac{\beta+1}{2} & \frac{-\beta+1}{2} & \beta & 0 \\ 1 & \frac{-6\beta^2+16\beta-1}{4} & \frac{6\beta^2-20\beta+5}{4} & \beta \\ \hline & \frac{-6\beta^2+16\beta-1}{4} & \frac{6\beta^2-20\beta+5}{4} & \beta \end{array}$$

$$\beta = 0.435866521508459.$$

(e) SDIRK3[1]

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 2-\sqrt{2} & \frac{2-\sqrt{2}}{2} & \frac{2-\sqrt{2}}{2} & 0 \\ 1 & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{2-\sqrt{2}}{2} \\ \hline & \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{2-\sqrt{2}}{2} \end{array}$$

(f) TR-BDF2[25]

Table 3.3.1: Some Runge-Kutta methods

step of (3.3.2) is the factorisation of the Jacobian matrix. This only needs be done once if formula (3.3.3) is used. The simplified approach can therefore be faster, even though more iterations are required. However, if the Jacobian is sparse and unstructured, (3.3.2) is most efficiently solved using iterative algorithms that do not require matrix factorisations. In this case, the pure Newton's method is the best choice.

3.3.3 Consistency, Order and Stability

To be consistent, the coefficients of a Runge-Kutta method must satisfy the condition

$$\sum_i b_i = 1.$$

If this requirement is fulfilled, the accuracy of the method is at least first order. Higher order methods can be constructed by adjusting the coefficients to fulfill certain *order conditions*, which can be derived by comparing (3.3.1) with the Taylor series of $\mathbf{f}$. To acheive second order convergence, for instance, the coefficients must satisfy

$$\sum_i b_i c_i = \frac{1}{2},$$

and to acheive third order convergence, the coefficients must in addition satisfy

$$\sum_i b_i c_i^2 = \frac{1}{3}$$
$$\sum_{i,j} b_i a_{ij} c_j = \frac{1}{6}.$$

The number of conditions is rapidly increasing with order, so Runge-Kutta methods of order higher than 5 are seldom used.

The stability function of a Runge-Kutta method is also determined by its coefficients. A direct application of (3.3.1) to the test equation (3.2.3), reveals that the stability function is given by

$$R(z) = 1 + z\,\mathbf{b}^\top \left(\mathbb{I} - z\mathbb{A}\right)^{-1} \mathbb{1},$$

where $\mathbf{b}^\top = \begin{bmatrix} b_1, & b_2, & \cdots & b_N \end{bmatrix}$, $\mathbb{1} = \begin{bmatrix} 1, & 1, & \cdots, & 1 \end{bmatrix}^\top$ and

$$\mathbb{A} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix}.$$

We can also write this expression in terms of determinants as

$$R(z) = \frac{\det\left(\mathbb{I} - z\mathbb{A} + z\mathbb{1}\mathbf{b}^\top\right)}{\det\left(\mathbb{I} - z\mathbb{A}\right)}.$$

In Figure 3.3.1, we have shown a complex contour plot of the stability functions for the methods in Table 3.3.1. The shaded areas correspond to the regions of absolute stability. We clearly see that the stability regions of the explicit methods are bounded. For explicit schemes, the upper triangular part of $\mathbb{A}$ is zero, and the stability function thus reduces to a polynominal expression in z. The stability function is therefore unbounded for z values with a large negative real part, which are typical for stiff equations. On the other hand, the stability function of implicit methods are rational functions of z. This is the reason of why they can be constructed such that $|R(z)| < 1$ even though the magnitude of z is large.

3.3.4 DAE Systems and Singular Perturbation Problems

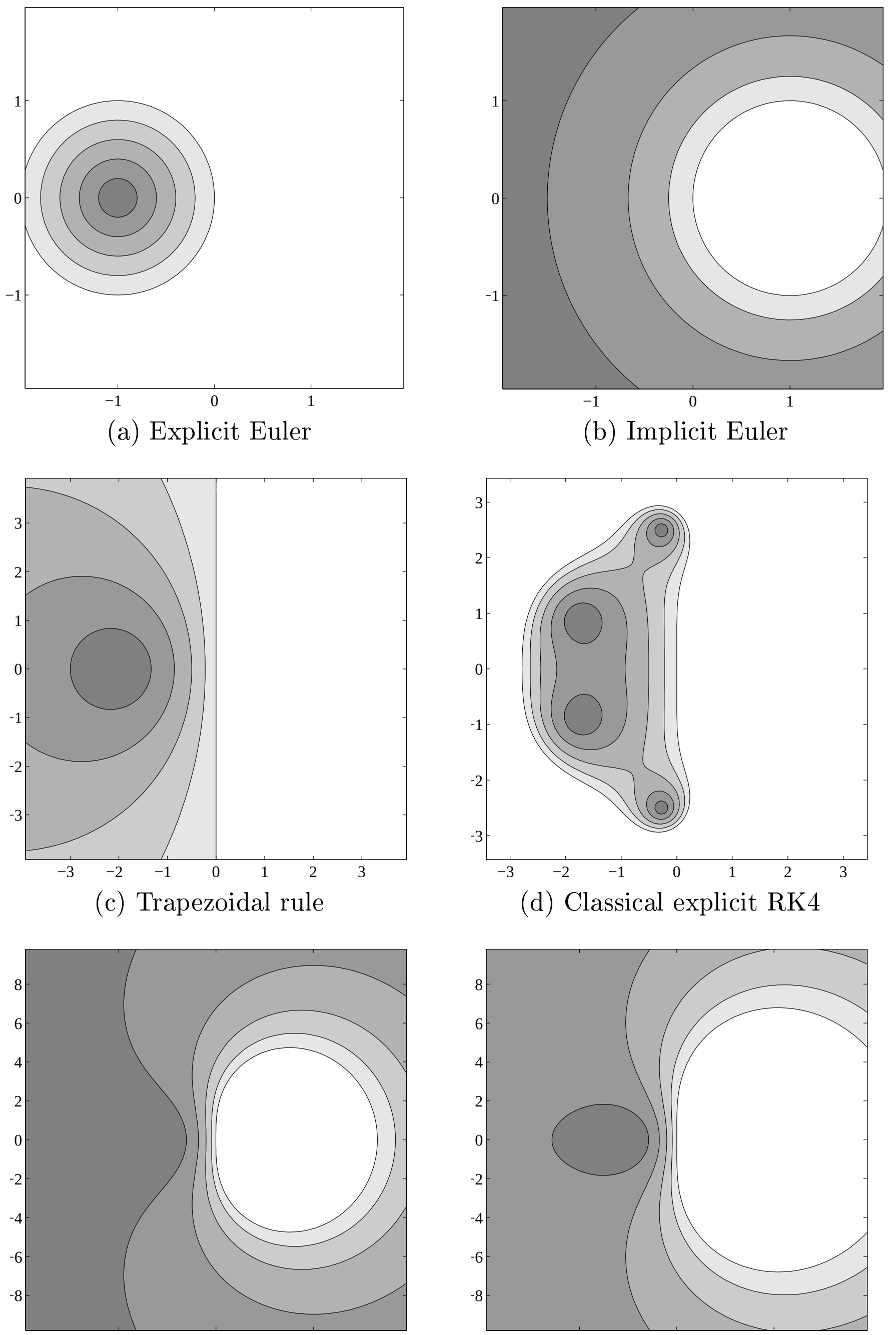
Recall that a differential-algebraic system of equations (DAE) is a system of the form

$$\mathbb{M}\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}), \tag{3.3.4}$$

where the *mass matrix* $\mathbb{M}$ is singular. We will mainly be interested in DAE systems of index 1, which means that the Jacobian matrix $\partial\mathbf{f}/\partial\mathbf{y}$ is invertible in the null space of $\mathbb{M}$. If this condition does not hold, the system is a higher-index DAE, and must be solved using other kinds of numerical methods.

Differential-algebraic systems are of interest to us, since the mass conservation equation (1.3.5) becomes a DAE if any of the chemical reactions are modelled as equilibrium reactions. Also, if the stiffness of an ODE becomes very large, it behaves almost like a DAE, and must be treated the same way. Let us take a moment to illustrate how this can happen. Consider the system described in Section 3.1.2, and suppose $K_1 = K_2 = K$ for simplicity. Furthermore, let us scale the concentrations by their equillibrium values, using $x = c_A$, $y = c_B$ and $z = c_C/K$. The system is then described by

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & \frac{1}{K} \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ 0 & -1 & 1 \\ 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix},$$



(a) Explicit Euler

(b) Implicit Euler

(c) Trapezoidal rule

(d) Classical explicit RK4

(e) SDIRK3[1]

(f) TR-BDF2[25]

Figure 3.3.1: Stability regions

or, equivalently,

$$\frac{d}{dt}\begin{bmatrix}1 & & \\ & 1 & \\ & & K\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix} = \begin{bmatrix}-1 & 0 & 1\\ 0 & -1 & 1\\ 1 & 1 & -2\end{bmatrix}\begin{bmatrix}x\\ y\\ z\end{bmatrix}. \tag{3.3.5}$$

Now, if K is very small, the mass matrix becomes near-singular, and the system behaves as a DAE. Often, such "near-DAE" systems are called *singular perturbation problems* (SPP). We can solve (3.3.5) approximately by setting $K = 0$, $z = (x + y)/2$, which leads to the nonstiff system

$$\frac{d}{dt}\begin{bmatrix}x\\ y\end{bmatrix} = \begin{bmatrix}-\frac{1}{2} & \frac{1}{2}\\ \frac{1}{2} & -\frac{1}{2}\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}.$$

While this approach is a stable and robust one, it may not be as straightforward to carry out for practical applications. First of all, since the reaction rates are usually nonlinear, it may be difficult to express the near-algebraic variables explicitly in terms of the differental variables. Secondly, a variable that is near-algebraic in one part of the simulation, can act like a normal differential variable elsewhere. For instance, if a reservoir region is flooded with CO_2, the chemistry of the region changes completely. This may cause equillibrium concentrations to change by orders of magnitude, slow reactions may become fast reactions, and so on. To use the approach outlined above, one must therefore use different sets of equations during the course of the simulation.

An alternative is to apply an integration scheme that can handle singular mass matrices. For instance, A-stable Runge-Kutta methods where the stability function satisfies

$$\left|\lim_{z\to\infty} R(z)\right| < 1,$$

can be used on equations like (3.3.4) without reformulating the system. Methods that are in addition *stiffly accurate*, are of special interest. Stiff accuracy simply means that the computed step value $\mathbf{y}_{n+1}$ is equal to the last internal stage $\mathbf{g}_N$. For instance, all the implicit methods of Table 3.3.1 are stiffly accurate, since the last two rows of their Butcher tableaus are equal. Methods with this property can compute the algebraic variables of DAE systems with the same accuracy as the differential variables. Otherwise, one may experience an order reduction in the algebraic components, as discovered by Prothero and Robinson[49]. To use a stiffly accurate method on problems of the form (3.3.4), the internal stages are computed according to

$$\begin{aligned}\mathbb{M}\mathbf{g}_1 =& \mathbb{M}\mathbf{y}_n + \Delta t \sum a_{1j}\mathbf{f}\left(\mathbf{g}_j\right)\\ \mathbb{M}\mathbf{g}_2 =& \mathbb{M}\mathbf{y}_n + \Delta t \sum a_{2j}\mathbf{f}\left(\mathbf{g}_j\right)\\ &\vdots\\ \mathbb{M}\mathbf{g}_N =& \mathbb{M}\mathbf{y}_n + \Delta t \sum a_{Nj}\mathbf{f}\left(\mathbf{g}_j\right).\end{aligned}$$

Finally, the solution is advanced by setting $\mathbf{y}_{n+1} = \mathbf{g}_N$.

3.3.5 Error Estimation

Runge-Kutta methods are often equipped with an *error estimator* of the form

$$\hat{\mathbf{y}}_{n+1} = \mathbf{y}_n + \Delta t \sum \hat{b}_j \mathbf{f}\left(\mathbf{g}_j\right),$$

which is a lower-order approximation to the solution at the end of the time step. We can use this value to obtain a crude estimate for the error of the integration rule. Let p and q be the global accuracy orders for $\mathbf{y}_{n+1}$ and $\hat{\mathbf{y}}_{n+1}$, respectively. For small values of Δt, we have

$$\left|\mathbf{y}\left(t_n + \Delta t\right) - \mathbf{y}_{n+1}\right| < \left|\mathbf{y}\left(t_n + \Delta t\right) - \hat{\mathbf{y}}_{n+1}\right| \approx \left|\mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1}\right|.$$

Thus, we can use $\varepsilon = \left|\mathbf{y}_{n+1} - \hat{\mathbf{y}}_{n+1}\right|$ as a crude error estimate for $\mathbf{y}_{n+1}$. If the order of accuracy for $\hat{\mathbf{y}}_{n+1}$ is p, we expect the error estimate to satisfy

$$\lim_{\Delta t \to 0} \frac{\varepsilon}{\Delta t^{p+1}} = C \tag{3.3.6}$$

for some constant C. This relation can be used to adjust the step size to match a prescribed error tolerance Tol. Specifically, if one time step of length Δt_n has been successfully computed, and the error has been estimated to ε_n, we expect that a step length of

$$\Delta t_{n+1} = \Delta t_n \left(\frac{Tol}{\varepsilon_n}\right)^{1/p+1} \tag{3.3.7}$$

will bring the error of the next step closer to the tolerance level.

Expression (3.3.7) is commonly used for calculating step lengths of nonstiff equations. For stiff equations, however, we would like to use steps that are much larger than the shortest time scale of the system. Thus, the asymptotic formula (3.3.6) may not be reliable. For reasons that we will not cover here, a step size controller of the form

$$\Delta t_{n+2} = \frac{(\Delta t_{n+1})^2}{\Delta t_n} \left(\frac{Tol \cdot \varepsilon_n}{(\varepsilon_{n+1})^2} \right)^{1/p+1} \tag{3.3.8}$$

is more appropriate in this case[17]. This expression uses the step sizes and error estimates from two previous steps, and is derived using techniques from control theory. If only a single previous time step is available, one can set $\Delta t_{n+1} = \Delta t_n$ and $\varepsilon_{n+1} = \varepsilon_n$. Although (3.3.8) is more reliable than (3.3.7), it does sometimes suggest time steps that results in errors significantly above the tolerance level. Whenever this happens, the usual approach is to halve the time step and restart integration.

3.3.6 Diagonally Implicit Runge-Kutta Methods

Some Runge-Kutta methods have coefficient matrices where all the entries are nonzero. These are sometimes called *fully implicit* Runge-Kutta methods (FIRK). If a FIRK scheme is used to solve an ODE of size m, each time step involves the solution of a nonlinear system in $m \cdot N$ variables, where N is the number of internal stages. Although the covergence orders of these methods are large, the computational cost of a time step is large as well, since all the internal stages are coupled. An alternative is provided by the *diagonally implicit* Runge-Kutta methods (DIRK)[1], where the coefficient matrix $\mathbb{A}$ is a lower triangular matrix. For these methods, the internal stages can be computed in sequence. Therefore, a DIRK time step is both easier to implement and faster to compute. Another advantage of the DIRK methods, is that every internal stage $\mathbf{g}_i$ is a first-order approximation to the solution at $t = t_n + c_i \Delta t$. Therefore, after an internal stage has been computed, one can use extrapolation techniques to find a good estimate for the next stage value. Such *stage value predictors* can then be used as a starting value for the Netwon iteration instead of $\mathbf{y}_n$, reducing the number of iterations required.

The coefficient matrix $\mathbb{A}$ of many DIRK methods are designed such that the entries on the diagonal are equal. These methods are called *singly diagonally implicit* (SDIRK). Three examples of such methods are shown in Table 3.3.2, where simple low-order error estimators have been added as additional stages. To see why equal diagonal elements is desirable, let us consider the structure of the nonlinear systems that must be solved for a DIRK method. Observe that each internal stage is computed from an expression of the form

$$\mathbf{g}_i - \Delta t\, a_{ii} \mathbf{f}(\mathbf{g}_i) = \mathbf{y}_n + \Delta t \left(a_{i,1} \mathbf{f}(\mathbf{g}_1) + \ldots + a_{i,i-1} \mathbf{f}(\mathbf{g}_{i-1}) \right). \tag{3.3.9}$$

In Section 3.3.2, we stated that it is often advantageous to solve (3.3.9) using simplified Newton iterations. In this case, the Jacobian is given by

$$\mathbb{I} - \Delta t \, a_{ii} \left. \frac{\partial \mathbf{f}}{\partial \mathbf{g}} \right|_{\mathbf{y}_n} .$$

If the diagonal elements a_{ii} are equal, the Jacobian is the same for all stage equations. Thus, the matrix need only be factorised once every time step, which may shorten the computational time significantly for large systems.

Yet another class of DIRK methods are the ESDIRK methods[36, 2], which are singly diagonally implicit schemes with an explicit first stage. Similar to the SDIRK methods, the diagonal entries of the coefficient matrix $\mathbb{A}$ are equal, except for the first entry, which is equal to zero. In other words, the first internal stage $\mathbf{g}_1$ is equal to $\mathbf{y}_n$, the function value at the beginning of the step. In Table 3.3.1, one example of this design (TR-BDF2) is shown. ESDIRK methods require the same amount of implicit stages as the SDIRK methods, to acheive the same order of accuracy. The additional advantage is that the Butcher tableau of ESDIRK methods have a larger amount of parameter values, compared to SDIRK methods of the same order. Thus, we have more degrees of freedom when constructing ESDIRK methods, and we are able to tune the parameters to suit special interests. For instance, Kværnø[36] was able to construct a family of L-stable ESDIRK methods where the second last internal stage is used as an L-stable free error estimator, see Table 3.3.3. This is not possible for SDIRK methods, at least not without adding implicit stages (as in Table 3.3.2), or sacrificing the stability of the error estimator. In addition, every internal stage $\mathbf{g}_i$ of the methods in Table 3.3.3 is a *second-order* approximation to $\mathbf{y}\left(t_n + c_i \Delta t\right)$, that is, the methods have a *stage order* of 2. This fact can be exploited to construct efficient stage predictors for the Newton iterations. For SDIRK methods, the maximum possible stage order is 1.

0.292893218813452	0.292893218813452	0	0
1[1]	0.707106781186548	0.292893218813452	0
1[2]	0	0.707106781186548	0.292893218813452

[1] Advancing step, order 2

[2] Error estimator, order 1

(a) SDIRK21[1]

0.435866521508459	0.435866521508459	0	0	0
0.717933260754229	0.282066739245771	0.435866521508459	0	0
1[1]	1.208496649176010	-0.644363170684469	0.435866521508459	0
1[2]	0.886315063833775	0	-0.322181585342234	0.435866521508459

[1] Advancing step, order 3

[2] Error estimator, order 2

(b) SDIRK32[1]

$\frac{1}{4}$	$\frac{1}{4}$	0	0	0	0	0
$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{4}$	0	0	0	0
$\frac{11}{20}$	$\frac{17}{50}$	$-\frac{1}{25}$	$\frac{1}{4}$	0	0	0
$\frac{1}{2}$	$\frac{371}{1360}$	$-\frac{137}{2720}$	$\frac{15}{544}$	$\frac{1}{4}$	0	0
1[1]	$\frac{25}{24}$	$-\frac{49}{48}$	$\frac{125}{16}$	$-\frac{85}{12}$	$\frac{1}{4}$	0
1[2]	$\frac{43}{24}$	$\frac{113}{48}$	$\frac{75}{16}$	$-\frac{85}{12}$	-1	$\frac{1}{4}$

[1] Advancing step, order 4

[2] Error estimator, order 3

(c) SDIRK43[18]

Table 3.3.2: SDIRK methods with error estimators

0	0	0	0	0
0.871733043	0.4358665215	0.4358665215	0	0
1^2	0.4905633884	0.07357009008	0.4358665215	0
1^1	0.3088099700	1.4905633883	-1.2352398797	0.4358665215

[1] Advancing step, order 3
[2] Error estimator, order 2

(a) ESDIRK32

0	0	0	0	0	0
1.145632125	0.5728160625	0.5728160625	0	0	0
0.597104988	0.167235462	-0.142946537	0.572816062500000	0	0
1^2	0.262603290	-0.311904327	0.476484975	0.5728160625	0
1^1	0.197216548	0.176843784	0.815442181	-0.762318576	0.5728160625

[1] Advancing step, order 4
[2] Error estimator, order 3

(b) ESDIRK43

0	0	0	0	0	0	0	0
0.52	0.26	0.26	0	0	0	0	0
1.230333210	0.13000000	0.840333210	0.26	0	0	0	0
0.895765984	0.223719615	0.476755323	-0.064708954	0.26	0	0	0
0.436393610	0.166485643	0.104500188	0.036314823	-0.130907045	0.26	0	0
1^2	0.138556402	0.000000000	-0.04245337	0.024466579	0.619430391	0.26	0
1^1	0.136597512	0.000000000	-0.054969088	-0.041186267	0.629933049	0.069624794	0.26

[1] Advancing step, order 5
[2] Error estimator, order 4

(c) ESDIRK54

Table 3.3.3: A family of ESDIRK methods[36]

Chapter 4

Numerical solution strategies

In the previous chapter, we presented a family of Runge-Kutta methods that were suitable for solving the reactive mass conservation equation. We now turn to the question of how to express the equation such that numerical integration methods are more easily applied. First, we investigate the possibilities of splitting the convective and reactive part of the equation, solving each of them in a sequential fashion. Secondly, we discuss how different changes of variables can increase the stability and speed of the solution process.

4.1 Operator Splitting

We have already stated (in Section 1.5) that the pressure, temperature and mass conservation equations are usually solved in a sequential manner when chemical reactions are involved. This procedure, called operator splitting, divides every time step into several substeps, each describing separate physical processes. Sometimes, it is feasible to employ operator splitting on the substep level as well. In particular, consider again the mass conservation substep, which may be abstractly described as

$$\frac{\partial \mathbf{c}}{\partial t} = \boldsymbol{\mathcal{L}}\left(\mathbf{c}\right) + \boldsymbol{\mathcal{R}}\left(\mathbf{c}\right).$$

Here, $\boldsymbol{\mathcal{L}}$ is the convective term, and $\boldsymbol{\mathcal{R}}$ the reactive term[1]. One splitted time step, with a step length of h, can then be written as

$$\frac{\partial \mathbf{v}}{\partial t} = \boldsymbol{\mathcal{L}}(\mathbf{v}) \qquad \mathbf{v}(0) = \mathbf{c}_n \tag{4.1.1a}$$

$$\frac{\partial \mathbf{w}}{\partial t} = \boldsymbol{\mathcal{R}}(\mathbf{w}) \qquad \mathbf{w}(0) = \mathbf{v}(h) \tag{4.1.1b}$$

$$\mathbf{c}_{n+1} = \mathbf{w}(h), \tag{4.1.1c}$$

where $\mathbf{v}$ and $\mathbf{w}$ are auxillary internal variables. The scheme approximates the solution by taking one internal step involving only convection, and then a second internal step involving only reaction.

The main advantage of this approach, is that separate, specialised numerical solvers can be applied for each substep. For instance, the transport step can be solved explicitly, using a method that minimizes numerical diffusion. Also, the reaction step can be solved using parallel computation, since the chemical reactions in a single grid block do not affect the concentrations at other locations. As we have seen, the reaction step must be solved implicitly for stability reasons, which is usually a computationally intensive procedure. Fortunately, since the the reactions are spatially decoupled, the size of the implicit system to solve is small. If the mass conservation equation is solved without splitting, one must use the expensive implicit approach on the system as a whole, which may be particularly time-consuming for two- and three-dimensional simulations. In addition, one loses the opportunity of solving the transport step using a diffusion-minimizing method.

There are, however, problems with the sequential approach as well. These issues are related both to the accuracy of the solution, and to the numerical stability. In this section, we will explore the problems that may arise, and consider some improved schemes that may alleviate them.

4.1.1 Splitting Errors

Consider at first the simple splitting scheme (4.1.1), and suppose a one-step method is used to solve each of the subequations. The overall scheme will then converge to the correct solution as $h \to 0$. In general, however, the approximation will only be first order accurate, even if exact solvers are used in each substep. This is because an

[1] As remarked in Section 2.2.3, the temperature equation has the same overall structure. Thus, the methods described in this section applies to the temperature equation as well.

additional *splitting error* is introduced to the solution. To highlight this phenomenon, we investigate the linear equation

$$\frac{d\mathbf{u}}{dt} = \mathbb{A}\mathbf{u} + \mathbb{B}\mathbf{u}, \quad t \in [0, T], \quad \mathbf{u}(0) = \mathbf{u}_0.$$

which has the exact solution

$$\mathbf{u}(T) = e^{(\mathbb{A}+\mathbb{B})T}\mathbf{u}_0.$$

Using Taylor series expansion, we may also express this as

$$\mathbf{u}(T) = \left(\mathbb{I} + T(\mathbb{A}+\mathbb{B}) + \frac{1}{2}T^2\left(\mathbb{A}^2 + \mathbb{A}\mathbb{B} + \mathbb{B}\mathbb{A} + \mathbb{B}^2\right) + \dots\right)\mathbf{u}_0. \tag{4.1.2}$$

Now, let us solve the differential equation by using the splitting scheme (4.1.1), with a splitting interval of h. Using exact subsolvers, a single time step will have the form

$$\mathbf{u}_{n+1} = e^{\mathbb{B}h}e^{\mathbb{A}h}\mathbf{u}_n,$$

or, using Taylor series expansion,

$$\mathbf{u}_{n+1} = \left(\mathbb{I} + h\mathbb{B} + \frac{1}{2}h^2\mathbb{B}^2 + \dots\right)\left(\mathbb{I} + h\mathbb{A} + \frac{1}{2}h^2\mathbb{A}^2 + \dots\right)\mathbf{u}_n.$$

Adding the time step increments up to $T = hN$ and simplifying, we arrive at

$$\mathbf{u}_N = \left(\mathbb{I} + hN(\mathbb{A}+\mathbb{B}) + \frac{1}{2}h^2N^2\left(\mathbb{A}^2 + \frac{N-1}{N}\mathbb{A}\mathbb{B} + \frac{N+1}{N}\mathbb{B}\mathbb{A} + \mathbb{B}^2\right) + \dots\right)\mathbf{u}_0.$$

Let us compare this answer to the known exact solution (4.1.2). We see that the first Taylor terms agree, but there is a slight discrepancy in the second order term:

$$\mathbf{u}(T) - \mathbf{u}_N = \frac{1}{2}hT(\mathbb{A}\mathbb{B} - \mathbb{B}\mathbb{A})\mathbf{u}_0 + O\left(h^2\right).$$

If the matrices $\mathbb{A}$ and $\mathbb{B}$ commute, i.e., $\mathbb{A}\mathbb{B} = \mathbb{B}\mathbb{A}$, all the Taylor terms agree, and the splitting error is zero. For noncommuting matrices, however, the overall method is only first order accurate, even though the subsolvers are exact. A similar analysis may also be carried out for nonlinear equations using the Lie operator formalism, e.g., along the lines of [38, 26].

4.1.2 Strang Splitting

From analyses of the splitting error, different splitting schemes can be devised, with improved convergence behaviour. One of the simplest was first proposed by Strang[59], and is called the *Strang splitting*. The idea is to introduce an additional internal stage into the splitting scheme,

$$\frac{\partial \mathbf{v}}{\partial t} = \boldsymbol{\mathcal{L}}(v) \qquad \mathbf{v}(0) = \mathbf{c}_n \tag{4.1.3a}$$

$$\frac{\partial \mathbf{w}}{\partial t} = \boldsymbol{\mathcal{R}}(w) \qquad \mathbf{w}(0) = \mathbf{v}(h/2) \tag{4.1.3b}$$

$$\frac{\partial \mathbf{z}}{\partial t} = \boldsymbol{\mathcal{L}}(w) \qquad \mathbf{z}(0) = \mathbf{w}(h) \tag{4.1.3c}$$

$$\mathbf{c}_{n+1} = \mathbf{z}(h/2). \tag{4.1.3d}$$

This new arrangement is equivalent to two half steps of (4.1.1), with reversed sequences. The symmetry causes first order splitting errors to cancel out for small values of h. Since the Strang splitting is simple to implement, it is quite commonly used. Similar schemes can also be devised for equations with several operators involved. Even higher-order splittings can be constructed, based on the same principles. These are plagued by stability problems, however, and are therefore rarely used. We refer to [26] for more details.

4.1.3 Iterative Splitting

Another way of reducing splitting errors is *iterative splitting*. This method is commonly used by the hydrology community, where it is referred to as SIA, Sequential Iterative Approach. Using the same notation as before, the iterative splitting scheme can be expressed as

$$\begin{aligned}
\frac{\partial \mathbf{v}^{(i)}}{\partial t} &= \boldsymbol{\mathcal{L}}\left(\mathbf{v}^{(i)}\right) + \boldsymbol{\mathcal{R}}\left(\mathbf{w}^{(i-1)}\right) & \mathbf{v}^{(i)}(0) &= \mathbf{c}_n \\
\frac{\partial \mathbf{w}^{(i)}}{\partial t} &= \boldsymbol{\mathcal{L}}\left(\mathbf{v}^{(i)}\right) + \boldsymbol{\mathcal{R}}\left(\mathbf{w}^{(i)}\right) & \mathbf{w}^{(i)}(0) &= \mathbf{c}_n \\
& & \mathbf{c}_{n+1} &= \mathbf{w}^{(final)}(h).
\end{aligned} \tag{4.1.4}$$

Here, i indicates the iteration index. The operators $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{R}}$ are still decoupled, even though they appear simultaneously in both equations. In the first substep, the function $\mathbf{w}$ is considered known, so the step is a differential equation for $\mathbf{v}$ only. The step is equivalent to a pure transport step, only with $\boldsymbol{\mathcal{R}}$ acting as a known mass source at each location. Likewise, the second step corresponds to a pure reaction step with a known convective mass source. To reduce splitting errors, we can now iterate upon this scheme, using the newly calculated function $\mathbf{w}$ in the first substep. Indeed, the overall structure of the algorithm resembles the Gauss-Seidel method for solving systems of linear equations. To start the iteration, we will need a first guess for $\mathbf{w}$. For the mass conservation equation, it is natural to choose $\mathbf{w}^{(0)}$ such that $\boldsymbol{\mathcal{R}}\left(\mathbf{w}^{(0)}\right) = 0$, i.e., chemical equillibrium. When using iterative splitting, the order of the splitting error is equal to the number of iterations, a very attractive feature[13]. In practice, the iterations are usually stopped when a specified tolerance limit is met. The number of required iterations can also be used to guide the choice of step length. According to [13], an iteration count above three usually suggests a step length reduction.

4.2 Linear Recombination Methods

In Section 2.2.3, we saw that the mass conservation equation can be described by

$$\frac{\partial \mathbf{c}}{\partial t} = \boldsymbol{\mathcal{L}}\left(\mathbf{c}\right) + \mathbb{S}^{\top}\mathbf{r}\left(\mathbf{c}\right), \tag{4.2.1}$$

where $\mathbb{S}$ is the stoichiometry matrix, and $\mathbf{r}$ are the reaction rates. While this is a perfectly valid formulation of the equation, it might not be suited for numerical computations. Since the equation is stiff, it must be solved by implicit methods, and we must therefore require that the Jacobian of the system is well-conditioned. This is not the case if any of the reactions are fast compared to the time scale of interest, or the concentrations differ widely in magnitude, both of which are commonly occuring situations. To alleviate this, the equation can be multiplied by a carefully designed matrix $\mathbb{M}$,

$$\mathbb{M}\frac{\partial \mathbf{c}}{\partial t} = \mathbb{M}\boldsymbol{\mathcal{L}}\left(\mathbf{c}\right) + \mathbb{M}\mathbb{S}^{\top}\mathbf{r}\left(\mathbf{c}\right), \tag{4.2.2}$$

such that the fast and slow modes of the equation are partially decoupled prior to integration. This corresponds to a linear recombination of the original mass conservation equations. If the matrix is designed such that the condition number of the Jacobian is low, $\mathbb{M}$ itself may be near-singular. This is no problem, as the transformed equation can be easily solved using a Runge-Kutta method that handles singular mass matrices. Sometimes, it will also be advantageous to reformulate the equation in terms of a different set of variables. For instance, if $\mathbb{N}$ is a well-conditioned matrix, we can define $\mathbf{c} = \mathbb{N}\mathbf{v}$ and express (4.2.2) in terms of $\mathbf{v}$,

$$\mathbb{M}\mathbb{N}\frac{\partial \mathbf{v}}{\partial t} = \mathbb{M}\boldsymbol{\mathcal{L}}\left(\mathbb{N}\mathbf{v}\right) + \mathbb{M}\mathbb{S}^{\top}\mathbf{r}\left(\mathbb{N}\mathbf{v}\right). \tag{4.2.3}$$

This may reduce the condition number of the Jacobian even more. In addition, this approach allows us to construct variables that are unaffected by reactions, which are then decoupled from the reactive variables.

4.2.1 Scaling the Equation

To allow meaningful discussions of stability and conditioning, we must scale the mass conservation equation according to typical concentrations and reaction rates. However, it is well-known that scaling of reaction-advection systems is a difficult task, since the concentrations and rates may change by orders of magnitude within short periods of time. When lacking better alternatives, we will usually choose to scale the concentrations by their initial values, and the reactions by their rate constants. Frequent rescaling may, however, be necessary during the course of the simulation.

Let $\mathbf{c}_0$ and $\mathbf{r}_0$ be the chosen concentration and reaction scales, respectively. Furthermore, let $\mathbb{C}$ and $\mathbb{R}$ be diagonal matrices with the entries of $\mathbf{c}_0$ and $\mathbf{r}_0$ on their respective main diagonals. A scaled version of (4.2.1) is then

$$\frac{\partial \mathbf{y}}{\partial t} = \boldsymbol{\mathcal{L}}\left(\mathbf{y}\right) + \mathbb{T}\mathbf{q}\left(\mathbf{y}\right), \tag{4.2.4}$$

where $\mathbb{C}\mathbf{y} = \mathbf{c}$, $\mathbf{q}(\mathbf{y}) = \mathbb{R}^{-1}\mathbf{r}\left(\mathbb{C}\mathbf{y}\right)$ and $\mathbb{T} = \mathbb{C}^{-1}\mathbb{S}^{\top}\mathbb{R}$. This equation will be the basis for our study of the different recombination methods.

4.2.2 Condition Number of the Jacobian

As described in Section 3.3, the application of an implicit integration method to (4.2.4) results in a number of nonlinear equations that must be solved with Newton's method. Recall that the iteration matrix used for this purpose is given by

$$\mathbb{J} = \mathbb{I} - \gamma\Delta t\frac{\partial \mathbf{f}}{\partial \mathbf{y}},$$

where Δt is the step length, γ is a method-specific parameter, and $\mathbf{f}$ is the ODE function,

$$\mathbf{f}(\mathbf{y}) = \boldsymbol{\mathcal{L}}\left(\mathbf{y}\right) + \mathbb{T}\mathbf{q}\left(\mathbf{y}\right).$$

Let us illustrate how the stiffness of the system can cause $\mathbb{J}$ to become ill-conditioned. For this purpose, consider the advection-reaction system analysed in Section 3.1.4. For simplicity, let the inflow concentrations be zero. When scaling the concentrations by their initial values, this system is given by

$$\mathbf{f}\left(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}\right) = -u\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} -k \\ 2k\beta \end{bmatrix}\left(y_1 - {y_2}^2\right),$$

and the Jacobian is

$$\frac{\partial \mathbf{f}}{\partial \mathbf{y}} = \begin{bmatrix} -u-k & 2ky_2 \\ 2k\beta & -u-4k\beta y_2 \end{bmatrix},$$

where k is the rate constant, u is the advection velocity, y_1 and y_2 are the scaled component concentrations, and β is the initial A concentration divided by the initial B concentration. Thus, $\mathbb{J}$ for this system is given by

$$\mathbb{J} = \begin{bmatrix} 1+\tilde{u}+\tilde{k} & -2\tilde{k}y_2 \\ -2\tilde{k}\beta & 1+\tilde{u}+4\tilde{k}\beta y_2 \end{bmatrix},$$

where $\tilde{u} = u\gamma\Delta t$ and $\tilde{k} = k\gamma\Delta t$. There are a number of ways this matrix can become ill-conditioned:

1. Concentration differences: If $\beta \gg 1$, the last row is much larger than the first one, leading to a condition number of order $\mathcal{O}\left(\beta\right)$.
2. Nonlinearity: If $y_2 \gg 1$, the last column is much larger than the first one, leading to a condition number of order $\mathcal{O}\left(y_2\right)$
3. Fast reactions: If $\tilde{k} \gg 1$, the two rows are almost parallel, leading to a condition number of order $\mathcal{O}\left(\tilde{k}\right)$

All of these situations can be alleviated by reducing the time step, as $\lim_{\Delta t\to 0}\mathbb{J} = \mathbb{I}$. A better way, however, is to multiply the system (4.2.4) with a mass matrix $\mathbb{M}$ prior to integration, and possibly use another set of variables $\mathbf{v} = \mathbb{N}^{-1}\mathbf{y}$. The system is then given by

$$\mathbb{M}\mathbb{N}\frac{\partial \mathbf{v}}{\partial t} = \mathbb{M}\boldsymbol{\mathcal{L}}\left(\mathbb{N}\mathbf{v}\right) + \mathbb{M}\mathbb{T}\mathbf{q}\left(\mathbb{N}\mathbf{v}\right),$$

and the Newton iteration matrix is

$$\mathbb{J} = \mathbb{M}\mathbb{N} - \gamma \Delta t \mathbb{M} \begin{bmatrix} -u-k & 2ky_2 \\ 2k\beta & -u-4k\beta y_2 \end{bmatrix} \mathbb{N}.$$

For instance, let

$$\mathbb{M} = \begin{bmatrix} 0 & \kappa \\ 1 & \frac{1}{2\beta} \end{bmatrix},$$
$$\kappa = \frac{1}{1+\tilde{u}+\tilde{k}+4\beta\tilde{k}y_2}$$

and

$$\mathbb{N} = \begin{bmatrix} -\frac{1}{2\beta} & 1 \\ 1 & 0 \end{bmatrix}.$$

This results in

$$\mathbb{J} = \begin{bmatrix} 1 & -2\kappa\beta\tilde{k} \\ 0 & 1+\tilde{u} \end{bmatrix}.$$

Unless $\beta\tilde{k}y_2 \ll 1$ and $\beta\tilde{k} \gg 1$ at the same time, the condition number of this matrix is bounded for all parameter values. Thus, the transformed system can be stably integrated using long steps regardless of the system's stiffness.

4.2.3 Reaction Invariants

Before we address the question of how to construct $\mathbb{M}$ and $\mathbb{N}$, we highlight another feature that can be obtained by reformulating the equations: It is often possible to construct $\mathbb{M}$ and $\mathbb{N}$ such that some of the variables in $\mathbf{v} = \mathbb{N}^{-1}\mathbf{y}$ are invariant of reactions. The requirement is that the stoichiometry matrix $\mathbb{S}$ has a nontrivial null space. This is true if the number of components is larger than the number of reactions, which is nearly always the case. For instance, consider the system given in the previous section. After reformulation, the new system is given by

$$\begin{bmatrix} \kappa & 0 \\ 0 & 1 \end{bmatrix} \frac{d\mathbf{v}}{dt} = -u \begin{bmatrix} \kappa & 0 \\ 0 & 1 \end{bmatrix} \mathbf{v} + \begin{bmatrix} 2k\beta\kappa \\ 0 \end{bmatrix} \left(v_2 - \frac{v_1}{2\beta} - {v_1}^2 \right).$$

Evidently, the variable v_2 is not affected by reactions, and is therefore called a *reaction invariant.*

To see how such reaction invariants can be identified, let us partition $\mathbb{M}$ into

$$\mathbb{M} = \begin{bmatrix} \mathbb{M}_1 \\ \mathbb{M}_2 \end{bmatrix}.$$

Furthermore let us construct $\mathbb{M}$ and $\mathbb{N}$ such that $\mathbb{M}_2\mathbb{T} = \mathbb{0}$ and $\mathbb{M}_2\mathbb{N} = \begin{bmatrix} \mathbb{0} & \mathbb{I} \end{bmatrix}$, and define $\mathbf{v}_2 = \mathbb{M}_2\mathbb{N}\mathbf{v}$. The reformulated system is then of the form

$$\mathbb{M}_1\mathbb{N}\frac{\partial \mathbf{v}}{\partial t} = \mathbb{M}_1\boldsymbol{\mathcal{L}}\left(\mathbb{N}\mathbf{v}\right) + \mathbb{M}_1\mathbb{T}\mathbf{q}\left(\mathbb{N}\mathbf{v}\right) \qquad (4.2.5a)$$

$$\frac{\partial \mathbf{v}_2}{\partial t} = \mathbb{M}_2\boldsymbol{\mathcal{L}}\left(\mathbb{N}\mathbf{v}\right). \qquad (4.2.5b)$$

It is now possible to use an IMEX scheme on (4.2.5), solving for the first equation implicitly, and the second explicitly. This way, the number of equations that must be solved implicitly is reduced. In addition, Kräutle[33] showed that $\mathbb{M}_2$ can be constructed such that $\mathbb{M}_2$ and $\boldsymbol{\mathcal{L}}$ commute, obtaining further decouplings.

4.2.4 Recombination by LU Factorisations

We now turn to the question of how to create the recombination matrices. A simple, but popular way is by using LU factorisation on $\mathbb{T}$. Recall that this procedure factorises the matrix into

$$\mathbb{P}\mathbb{T} = \begin{bmatrix} \mathbb{L}_1 \\ \mathbb{L}_2 \end{bmatrix} \mathbb{U},$$

where $\mathbb{P}$ is a permutation matrix, $\mathbb{L}_1$ is invertible and lower triangular, and $\mathbb{U}$ is upper triangular. The decoupling is then obtained by the matrices

$$\begin{aligned} \mathbb{M}_1 &= \begin{bmatrix} \mathbb{I} & \mathbb{0} \end{bmatrix} \mathbb{P} \\ \mathbb{M}_2 &= \begin{bmatrix} -\mathbb{L}_2\mathbb{L}_1^{-1} & \mathbb{I} \end{bmatrix} \mathbb{P} \\ \mathbb{N} &= \mathbb{M}^{-1}. \end{aligned}$$

Variants of this approach is used by [23, 58, 46, 7, 54], to name a few. We see at once that $\mathbb{M}_2\mathbb{T} = \mathbb{0}$ and $\mathbb{M}_2\mathbb{N} = \begin{bmatrix} \mathbb{0} & \mathbb{I} \end{bmatrix}$, so the transformed system will be of the form (4.2.5), with $\mathbf{v}_2 = \mathbb{M}_2\mathbf{y}$ as reaction invariants. It is interesting to note that with this choice of recombination matrices, $\mathbf{v}_1 = \mathbb{M}_1\mathbf{y}$ will actually be a subset of $\mathbf{y}$. Many authors use the term *secondary species* for these components. This terminology makes most sense when all the chemical reactions are equillibrium reactions, as the secondary species can then be eliminated from the system of equations. Also, some authors use the term *total concentrations* for the reaction invariants $\mathbf{v}_2$. Historically, such variables were often created using physical intuition, letting for instance $\mathbf{v}_2$ correspond to the total concentration of chemical elements (say, C, O, Na, H etc.), or the total charge. As these variables are not affected by reactions, they are natural choices for reaction invariants, and are still used by some authors (see [41], for instance). However, the less intuitive reaction invariants found by LU factorisation are equally valid, and may also have improved stability properties.

LU factorisation is unique up to the choice of $\mathbb{P}$, which is often called the row pivoting matrix. $\mathbb{P}$ can not be chosen entirely freely, since we must require that the upper part of $\mathbb{P}\mathbb{T}$ is invertible. The best numerical stability is gained when $\mathbb{P}$ is chosen such that the condition number of $\mathbb{N}$ is low. Typically, this corresponds to letting $\mathbf{v}_1$ contain the species with the smallest concentrations, as is also observed by Saaltink et al.[54]. Modern algorithms for LU factorisation can choose $\mathbb{P}$ automatically, which is usually the more reliable approach. However, to my knowledge, the literature only considers manual row pivoting for choosing primary and secondary species.

The matrix design outlined above generates the maximum number of reaction invariants, which is determined by the null space of $\mathbb{T}^\top$. It also reduces high condition numbers if they are caused by large concentration differences, but not if they are caused by fast reactions. An improved scheme is given by the following choice,

$$\begin{aligned}
\mathbb{M}_1 &= \mathbb{D}\begin{bmatrix}\mathbb{L}_1^{-1} & \mathbb{0}\end{bmatrix}\mathbb{P}\\
\mathbb{M}_2 &= \begin{bmatrix}-\mathbb{L}_2\mathbb{L}_1^{-1} & \mathbb{I}\end{bmatrix}\mathbb{P}\\
\mathbb{N}^{-1} &= \begin{bmatrix}\mathbb{I} & \mathbb{0}\\ -\mathbb{L}_2\mathbb{L}_1^{-1} & \mathbb{I}\end{bmatrix}\mathbb{P}.
\end{aligned}$$

Here, $\mathbb{D}$ is a diagonal matrix constructed as follows: Let an element of $\mathbb{U}$ be denoted by u_{ij}. The diagonal elements of $\mathbb{D}$ are then given by

$$d_{ii} = \frac{1}{\max\left(1,\, \max_j |u_{ij}|\right)}.$$

This scheme allows both the concentrations and reaction rates of the model to be widely different, without having adverse effects on the system's condition number. An even better result is obtained if the reactions are sorted in descending order according to their typical rates.

4.2.5 Other Factorisation Methods

Another way of forming the recombination matrices, is by QR factorisation. This is similar to the method used by Friedly and Rubin[14], which is based on Gram-Schmidth orthogonalisation. In short, this factorisation decomposes the stoichiometry matrix according to

$$\mathbb{T} = \begin{bmatrix} \mathbb{Q}_1 & \mathbb{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbb{R} \\ \mathbb{0} \end{bmatrix},$$

where the matrix $\mathbb{Q} = \begin{bmatrix} \mathbb{Q}_1 & \mathbb{Q}_2 \end{bmatrix}$ is orthogonal, and $\mathbb{R}$ is upper triangular. From this decomposition, we can construct the recombination matrices

$$\begin{aligned} \mathbb{N} &= \mathbb{Q} \\ \mathbb{M}_1 &= \mathbb{D}\mathbb{Q}_1^\top \\ \mathbb{M}_2 &= \mathbb{Q}_2^\top \end{aligned}$$

The matrix $\mathbb{D}$ is constructed the same way as in the previous section, only with $\mathbb{R}$ used in place of $\mathbb{U}$. With this choice, the conversion between $\mathbf{v}$ and the scaled concentrations $\mathbf{y}$ is done via the orthogonal matrix $\mathbb{Q}$, which has the optimal condition number of 1. On the other hand, the computational cost of computing a QR factorisation is larger than for the LU factorisation. This method also performs best if the reactions are sorted according to their rates.

A third way of making recombination matrices, is by a singular value decomposition (SVD). This is numerically equivalent to the method used by Kräutle[33], who describes it in terms of the generalized inverse of the stoichiometry matrix. After applying the SVD algorithm, the scaled stoichiometry matrix is decomposed as

$$\mathbb{T} = \begin{bmatrix} \mathbb{U}_1 & \mathbb{U}_2 \end{bmatrix} \begin{bmatrix} \mathbb{E} \\ \mathbb{0} \end{bmatrix} \mathbb{V}^\top.$$

Here, the matrices $\mathbb{U} = \begin{bmatrix} \mathbb{U}_1 & \mathbb{U}_2 \end{bmatrix}$ and $\mathbb{V}$ are orthogonal, and $\mathbb{E}$ is a positive semidefinite diagonal matrix. Using this factorisation, we can choose the recombination matrices as

$$\begin{aligned} \mathbb{N} &= \mathbb{U} \\ \mathbb{M}_1 &= \mathbb{D}\mathbb{U}_1^\top \\ \mathbb{M}_2 &= \mathbb{U}_2^\top, \end{aligned}$$

similar to what we did for the QR factorisation. The $\mathbb{D}$ matrix is this time computed using $\mathbb{E}$ instead of $\mathbb{R}$. If this method is used, it is not necessary to sort the reactions, since an automatic reordering is provided by the algorithm. However, the cost of computing an SVD decomposition is even a bit larger than for the QR factorisation.

4.2.6 Eliminating Equillibrium Reactions

For a typical reaction network, the rate of some reactions are so fast that they are best described as equillibrium reactions. In this case, the actual rate of the reactions are considered unknown, thus becoming additional variables in the mass conservation equation. Specifically, let the reaction rate vector be given by

$$\mathbf{q}\left(\mathbf{y},\, \mathbf{q}_{eq}\right) = \begin{bmatrix} \mathbf{q}_{eq} \\ \mathbf{q}_{kin}(\mathbf{y}) \end{bmatrix},$$

where $\mathbf{q}_{eq}$ are the unknown equillibrium rates, and $\mathbf{q}_{kin}(\mathbf{y})$ are known expressions describing the scaled kinetic rates. Also, we partition the stoichiometry matrix and the equillibrium constant vector accordingly,

$$\begin{aligned} \mathbb{S} &= \begin{bmatrix} \mathbb{S}_{eq} \\ \mathbb{S}_{kin} \end{bmatrix} \\ \mathbb{T} &= \begin{bmatrix} \mathbb{T}_{eq} & \mathbb{T}_{kin} \end{bmatrix} \\ \mathbf{K} &= \begin{bmatrix} \mathbf{K}_{eq} \\ \mathbf{K}_{kin} \end{bmatrix}. \end{aligned}$$

To close the system, we add an expression describing chemical equillibrium,

$$\mathbf{Q}(\mathbf{y}) = \ln \mathbf{K}_{eq} - \mathbb{S}_{eq} \ln \mathbf{a}\left(\mathbf{y}\right) = \mathbf{0}. \tag{4.2.6}$$

The function $\mathbf{a}(\mathbf{y})$ denotes the activities of the species. If any of the equillibrium reactions are inhomogenous, the expression must be modified as explained in Section 2.1.5.

In total, this gives the following system for describing mass conservation,

$$\mathbf{0} = \mathbf{Q}(\mathbf{y}) \tag{4.2.7a}$$

$$\frac{\partial \mathbf{y}}{\partial t} = \boldsymbol{\mathcal{L}}\left(\mathbf{y}\right) + \mathbb{T}_{eq}\mathbf{q}_{eq} + \mathbb{T}_{kin}\mathbf{q}_{kin}\left(\mathbf{y}\right). \tag{4.2.7b}$$

Unfortunately, (4.2.7) is a differential-algebraic system of index 2, which is difficult to solve numerically. In addition, the system is larger than the ones we have previously encountered, because of the extra variables included. However, we can use techniques similar to the ones used in Sections 4.2.4 and 4.2.5, to eliminate the unknown equillibrium rates and reduce the size and index of the system. For instance, consider an LU factorisation of $\mathbb{T}_{eq}$,

$$\mathbb{PT}_{eq} = \begin{bmatrix} \mathbb{L}_1 \\ \mathbb{L}_2 \end{bmatrix} \mathbb{U}.$$

We can use this decomposition to construct an elimination matrix of the form

$$\begin{aligned} \mathbb{M}_1 &= \begin{bmatrix} \mathbb{I} & \mathbb{0} \end{bmatrix} \mathbb{P} \\ \mathbb{M}_2 &= \begin{bmatrix} -\mathbb{L}_2\mathbb{L}_1^{-1} & \mathbb{I} \end{bmatrix} \mathbb{P} \\ \mathbb{N} &= \mathbb{M}^{-1}. \end{aligned}$$

This results in the system

$$\mathbf{0} = \mathbf{Q}(\mathbb{N}\mathbf{v}) \tag{4.2.8a}$$

$$\mathbb{M}_1\mathbb{N}\frac{\partial \mathbf{v}}{\partial t} = \mathbb{M}_1\boldsymbol{\mathcal{L}}\left(\mathbb{N}\mathbf{v}\right) + \mathbb{L}_1\mathbb{U}\mathbf{q}_{eq} + \mathbb{M}_1\mathbb{T}_{kin}\mathbf{q}_{kin}\left(\mathbb{N}\mathbf{v}\right). \tag{4.2.8b}$$

$$\frac{\partial \mathbf{v}_2}{\partial t} = \mathbb{M}_2\boldsymbol{\mathcal{L}}\left(\mathbb{N}\mathbf{v}\right) + \mathbb{M}_2\mathbb{T}_{kin}\mathbf{q}_{kin}\left(\mathbb{N}\mathbf{v}\right), \tag{4.2.8c}$$

where $\mathbf{v}_2 = \mathbb{M}_2\mathbb{N}\mathbf{v}$ is the lower part of $\mathbf{v}$. Now, we can remove (4.2.8b) from the system, obtaining an index-1 DAE where the number of variables equals the number of components. Thereafter, we can apply the methods of Sections 4.2.4 and 4.2.5 to (4.2.8c), to identify reaction invariants and reduce the condition number. Finally, if the activity coefficients can be regarded as constant during the time step, (4.2.8a) can be solved explicitly for $\mathbf{v}_1$, and substituted into (4.2.8c). This reduces the number of equations even further.

4.3 Ensuring Positivity During Integration

The chemical rate functions are often ill-behaved or undefined when the concentrations of aqueous components are nonpositive. For instance, in the rate law (2.1.10), the concentrations of aqueous reactants appear in the denominator. Also, in the equilibrium expression (4.2.6), the logarithms of the concentrations are involved. Thus, it is vital to ensure that the concentrations of aqueous components always stay positive, not only in the final solution, but also during intermediate calculations. As we have seen in Chapter (3), integration of the mass conservation equation by implicit methods leads to nonlinear equations that must be solved by Newton's method. Because of the nonlinear nature of the equations, Newton's method frequently produces intermediate iterates with negative concentrations, which may cause the iteration to fail, diverge, or even converge to a nonphysical solution. This behaviour becomes even more difficult to control when a change of variables is used, as we have done in the previous sections. Since the entries of the transformed variable **v** can be both positive and negative, it is not immediately clear how to prevent intermediate **v** iterates that correspond to negative concentrations. In this section, we present techniques that can be used to deal with this problem.

To illustrate how negative concentrations can appear during Newton iterations, consider the simple reaction $A \to B$, with the scaled reaction rate $q = 10 - \frac{10}{y}$, where y is the concentration of A. The evolution of y is then described by

$$\frac{dy}{dt} = -10 + \frac{10}{y}.$$

Let us integrate this equation using, say, the implicit Euler method. Using a step length of 1, we have

$$y_{end} = y_{init} - 10 + \frac{10}{y_{end}}, \tag{4.3.1}$$

where y_{init} and y_{end} are the values at the beginning and end of the time step, respectively. This is a quadratic equation for y_{end}, whose positive solution is in the range [1, 2] if y_{init} is in the range [1, 7]. Let us now apply Newton's method to find y_{end}. With $x_0 = y_{init}$ as a starting value, a single iteration gives the result

$$x_1 = x_0 + \frac{-10x_0^2 + 10x_0}{x_0^2 + 10}.$$

If y_{init} happens to be in the range [3, 7], however, this step would produce a negative value. If the iteration is allowed to continue further from this point, the iterative algorithm would converge to a nonphysical negative solution. Although a positive solution of (4.3.1) do exist, the Newton algorithm is not able to detect it.

The simplest way of avoiding the situation described above, is to reduce the length of the integration step every time a negative concentration occurs. This will bring the value of y_{end} closer to y_{init}, and the algorithm will then converge to the physically correct solution. For the problems we are considering, however, error tolerances are crude, and long time steps are preferred. There are two alternative ways of forcing the intermediate Newton steps to become positive. The first one, recommended by

Bethke[7], is to use partial Newton steps whenever a full step leads to negative concentrations. The method requires that the nonlinear equation is formulated in terms of the concentrations $\mathbf{c}$ or the scaled concentrations $\mathbf{y}$. For instance, let the equation be defined by the equation $\mathbf{f}(\mathbf{y}) = \mathbf{0}$. Bethke's scheme, adapted for simplified Newton iterations, is then given by

$$\mathbb{J} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{y}}\right|_{\mathbf{y}_0} \tag{4.3.2a}$$

$$\Delta \mathbf{y}_n = \mathbb{J}^{-1} \mathbf{f}\left(\mathbf{y}_n\right) \tag{4.3.2b}$$

$$1/\delta = \max\left(1, \frac{1}{1-\gamma} \cdot \frac{\Delta \mathbf{y}_n}{\mathbf{y}_n}\right) \tag{4.3.2c}$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n - \delta \Delta \mathbf{y}_n, \tag{4.3.2d}$$

where the fraction of vectors $\Delta \mathbf{y}_n / \mathbf{y}_n$ means element-wise division. With this setup, the entries of $\mathbf{y}$ are at most reduced by a factor of γ each step, so the algorithm will never produce any negative concentrations. Bethke suggests a parameter value of $\gamma = 0.5$, but a smaller value (e.g. $\gamma = 0.1$) often allows faster convergence without sacrificing stability. The parameter may even be set to a different value for each species. For instance, it makes sense to set $\gamma = 0$ for mineral species, since kinetic reaction rates are well-defined also when the amount of minerals present is zero.

Bethke's algorithm can be modified to be used with the transformation techniques described in the previous sections. If any of these are applied, the mass conservation equation is formulated in terms of a linear combination of concentrations, $\mathbf{v} = \mathbb{N}^{-1}\mathbf{y}$. Thus, the nonlinear equations arising from an implicit integration step, will be of the form $\mathbf{f}(\mathbf{v}) = \mathbf{0}$. In this case, we can apply (4.3.2) to the composite equation $\mathbf{f}(\mathbb{N}^{-1}\mathbf{y})$. After some simplifications, this results in the scheme

$$\begin{aligned} \mathbb{J} &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{v}}\right|_{\mathbf{v}_0} \\ \Delta \mathbf{v}_n &= \mathbb{J}^{-1} \mathbf{f}\left(\mathbf{v}_n\right) \\ 1/\delta &= \max\left(1, \frac{1}{1-\gamma} \cdot \frac{\mathbb{N}\Delta \mathbf{v}_n}{\mathbb{N}\mathbf{v}_n}\right) \\ \mathbf{v}_{n+1} &= \mathbf{v}_n - \delta \Delta \mathbf{v}_n. \end{aligned}$$

A second strategy is to reformulate the nonlinear equation using logarithms. The idea is to substitute $\mathbf{v}$ with $\boldsymbol{\ell} = \ln\left(\mathbb{N}\mathbf{v}\right)$, and solve the equation $\mathbf{f}\left(\mathbb{N}^{-1}e^{\boldsymbol{\ell}}\right) = \mathbf{0}$ with respect to $\boldsymbol{\ell}$. This results in the scheme

$$\begin{aligned}
\mathbb{J} &= \left.\frac{\partial \mathbf{f}}{\partial \boldsymbol{\ell}}\right|_{\boldsymbol{\ell}_0} \\
\boldsymbol{\ell}_{n+1} &= \boldsymbol{\ell}_n - \mathbb{J}^{-1}\mathbf{f}\left(\mathbb{N}^{-1}e^{\boldsymbol{\ell}_n}\right),
\end{aligned}$$

or, equivalently,

$$\begin{aligned}
\mathbb{J} &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{v}}\right|_{\mathbf{v}_0} && \text{(4.3.3a)} \\
\Delta\mathbf{v}_n &= \mathbb{J}^{-1}\mathbf{f}\left(\mathbf{v}_n\right) && \text{(4.3.3b)} \\
\mathbf{y}_n &= \mathbb{N}\mathbf{v}_n && \text{(4.3.3c)} \\
\Delta\mathbf{y}_n &= \mathbb{N}\Delta\mathbf{v}_n && \text{(4.3.3d)} \\
\mathbf{y}_{n+1} &= \mathbf{y}_n e^{-\Delta\mathbf{y}_n/\mathbf{y}_0} && \text{(4.3.3e)} \\
\mathbf{v}_{n+1} &= \mathbb{N}^{-1}\mathbf{y}_{n+1}. && \text{(4.3.3f)}
\end{aligned}$$

If the value of $\Delta\mathbf{y}_n$ is small compared to $\mathbf{y}_n$, the iteration schemes (4.3.2) and (4.3.3) give approximately equal results. If $\Delta\mathbf{y}_n$ is large and positive, the logarithm formulation may lead to unstable computations and failed steps. On the other hand, if $\Delta\mathbf{y}_n$ is large and negative, (4.3.3) often provides faster convergence than (4.3.2). Overall, we have found the logarithm formulation to be the most efficient way of ensuring positivity during integration. For gaseous CO_2, the use of logarithms is not as natural, since the gas may become completely dissolved in the water phase during integration. The same line of reasoning applies to the solid minerals, which may also dissolve completely. For these components, algorithm (4.3.2) with $\gamma = 0$ should be used.

Chapter 5

Results

In the preceding chapters, we have presented several different ways of solving the mass conservation equation in the presence of chemically reactive components. In this chapter, we will test the behaviour of the methods most commonly used in practice, with respect to accuracy, efficiency and stability. In particular, we will investigate the interplay between different sources of numerical errors, related to splitting, discretisation and temporal integration. Some of the tests will be performed on a one-dimensional case, as one-dimensional examples are easier to analyse when several methods are to be compared. Also, we tested the most promising methods on a two-dimensional example, with a more complicated mineralogy.

5.1 The Stability of Dawsonite

The first test case we consider, is based on the data in [21]. Here, the stability of the mineral Dawsonite is investigated, when the amount of CO_2 within a reservoir is declining. For high CO_2 concentrations, Dawsonite is stable, and the precipitation of Dawsonite provides an increased potensial for CO_2 storage. However, these elevated amounts of CO_2 will only be sustained during the injection period. Thus, it is important to know how fast the mineral dissolves when injection is terminated, and the concentration of aqueous CO_2 near the injection point starts to fall. If the dissolution process is fast, carbon that is captured within the mineral will be released into the brine once again. In this case, the mineral capture mechanism is only a temporary one.

The test problem simulates a one-dimensional core sample, filled with brine, that has a high CO_2 concentration initially. During the course of the simulation, water with low CO_2 content is injected into the sample at a constant rate. This resembles the situation that will occur in a storage reservoir after CO_2 injection is terminated, in which gaseous CO_2 rises from the injection point due to boyancy forces, and is replaced by brine with a lower CO_2 concentration.

We have chosen a one-dimensional test example for several reasons. First of all, one-dimensional simulations are important in their own right. Since they are easier to solve, one can often include a more accurate geochemical model in a one-dimensional model without sacrificing stability and computational speed. For instance, the widely used geochemical simulator PHREEQC includes support for one-dimensional transport only. Also, even multidimensional problems can be solved as a a collection of one-dimensional problems if streamline-based methods is used to model the flow (see, for instance, [43]). Finally, the results emerging from a one-dimensional simulation is easier to interpret, and is therefore suitable for comparing different numerical methods against one another.

The initial and inflow concentrations of the components used in the test case are shown in Table 5.1.1, whereas the reactions and their rates are shown in Table 5.1.2. The relevant equillibrium constants are calculated by assuming that the initial brine composition is in equillibrium with the minerals present.

Compared to [21], we have simplified the mineralogy slightly. First of all, we have assumed that the concentration of Na^+ is constant. Since the amount of Na^+ initially present is very large, it is not significantly affected by chemical reactions. Secondly, we have also assumed that the concentration of $SiO_2(aq)$ is constant. This is justified by the fact that $SiO_2(aq)$ stays in approximate equillibrium with quartz, which is also assumed to be present in the sample. Finally, we have used the common assumption that the activity coefficient of a single aqueous component is determined by the ionic strength. This allows us to assume that the activity coefficients are constant, since the ionic strength is mainly determined by the concentrations of Na^+ and Cl^-, which are approximately independent of the reactions.

5.1.1 Mathematical Formulation

We choose to order our model components as follows,

$$\left(\mathrm{H}^+,\ \mathrm{HCO}_3^-,\ \mathrm{CO}_2(\mathrm{aq}),\ \mathrm{Ca}^{2+},\ \mathrm{Al}^{3+},\ \text{Calcite},\ \text{Albite},\ \text{Dawsonite}\right).$$

Components	Initial concentrations	Inflow concentrations
H^+	1.76×10^{-5}	8.79×10^{-7}
HCO_3^-	5.37×10^{-2}	3.46×10^{-3}
$CO_2(aq)$	5.26×10^{-1}	7.82×10^{-4}
Ca^{2+}	3.03×10^{-2}	2.32×10^{-2}
Al^{3+}	1.77×10^{-8}	7.00×10^{-8}
Na^+	9.46×10^{-1}	9.46×10^{-1}
Cl^-	9.68×10^{-1}	9.68×10^{-1}
$SiO_2(aq)$	5.71×10^{-4}	5.71×10^{-4}

(a) Aqueous components

Minerals	Initial volume fraction
Calcite	0.06
Dawsonite	0.1
Albite	0.003
Other minerals	0.587

(b) Mineral components

Table 5.1.1: Initial and inflow concentrations

Reactions	$k_{max}/[\text{mol}/\text{dm}^3\,\text{yr}]$
$CO_2(aq) + H_2O \rightleftarrows HCO_3^- + H^+$	equillibrium
$\text{Calcite} + H^+ \rightleftarrows Ca^{2+} + HCO_3^-$	6.7×10^{-1}
$\text{Dawsonite} + 3H^+ \rightleftarrows Na^+ + Al^{3+} + HCO_3^- + 2H_2O$	1.0×10^{-2}
$\text{Albite} + 4H^+ \rightleftarrows Na^+ + Al^{3+} + 3SiO_2(aq) + 2H_2O$	3.6×10^{-3}

Table 5.1.2: Reactions and reaction rates

Our stoichiometry matrix is given by

$$\mathbb{S} = \begin{array}{c} \begin{array}{cccccccc} H^+ & HCO_3^- & CO_2(aq) & Ca^{2+} & Al^{3+} & \text{Calcite} & \text{Albite} & \text{Dawsonite} \end{array} \\ \begin{bmatrix} 1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & 0 & -1 & 0 & 0 \\ -3 & 1 & 0 & 0 & 1 & 0 & 0 & -1 \\ -4 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \end{bmatrix} \end{array},$$

where the order of the reactions are the same as in Table 5.1.2. We can divide the matrix into four parts as follows,

$$\mathbb{S} = \begin{bmatrix} \mathbb{S}_{eq} \\ \mathbb{S}_{kin} \end{bmatrix} = \begin{bmatrix} \mathbb{S}^{aq} & \mathbb{S}^{min} \end{bmatrix} = \begin{bmatrix} \mathbb{S}^{aq}_{eq} & \mathbb{S}^{min}_{eq} \\ \mathbb{S}^{aq}_{kin} & \mathbb{S}^{min}_{kin} \end{bmatrix}.$$

The upper part consists of one row, namely, the equillibrium reaction, whereas the other rows describe the mineral reactions. Likewise, the 5 leftmost columns correspond to the aqueous species, while the 3 rightmost columns correspond to the mineral species.

Since we have assumed chemical equillibrium initially, it is reasonable to scale the concentrations by their initial values. We also choose to scale the reaction rates by the the rate constants k_{max} from Table 5.1.2, times the initial volume fractions from Table 5.1.1. Using the notation from Section 4.2.1, let $\mathbb{T}$ be the scaled and transposed stoichiometry matrix, $\mathbf{y}$ the scaled concentrations, and $\mathbf{q}$ the scaled reaction rates. As with the stoichiometry matrix, we partition $\mathbb{T}$, $\mathbf{y}$ and $\mathbf{q}$ according to aqueous/mineral components and kinetic/equillibrium reactions. With these definitions, the evolution of the fluid composition is given by the mass conservation equation and the equillibrium condition,

$$\mathbf{0} = \mathbb{S}^{aq}_{eq} \ln \mathbf{y}_{aq} \tag{5.1.1a}$$

$$\frac{\partial \mathbf{y}}{\partial t} = -\frac{\mathbf{u}}{\phi} \cdot \frac{\partial \mathbf{y}}{\partial x} + \mathbb{T}_{eq}\mathbf{q}_{eq} + \mathbb{T}_{kin}\mathbf{q}_{kin}(\mathbf{y}). \tag{5.1.1b}$$

The dot operator $(\cdot)$ denote element-wise multiplication, and $\mathbf{u}$ is a vector of the form

$$\mathbf{u} = u \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^\top,$$

where u is the darcy velocity of the aqueous phase. Observe that the entries corresponding to the immobile mineral species are set to zero. The kinetic reaction rates $\mathbf{q}_{kin}$ are modelled using (2.1.10), assuming no catalytic species, as is done in [21]. In terms of $\mathbf{y}$, the rates are given by

$$\mathbf{q}_{kin}(\mathbf{y}) = \mathbf{y}_{min} \cdot \left(1 - \exp\left(\mathbb{S}^{aq}_{kin} \ln \mathbf{y}_{aq}\right)\right), \tag{5.1.2}$$

with the Jacobian matrix

$$\frac{\partial \mathbf{q}_{kin}}{\partial \mathbf{y}} = \left[-\mathbb{D}\left(\mathbf{y}_{min}\right) \cdot \exp\left(\mathbb{S}^{aq}_{kin}\mathbf{y}_{aq}\right) \mathbb{S}^{aq}_{kin}, \quad \mathbb{D}\left(1 - \exp\left(\mathbb{S}^{aq}_{kin} \ln \mathbf{y}_{aq}\right)\right) \right],$$

where $\mathbb{D}(\mathbf{v})$ symbolises a diagonal matrix with the entries of $\mathbf{v}$ on its main diagonal. As long as the concentrations stay fairly close to their initial values, the entries of this Jacobian are of the same magnitude, and no further rescaling will be required.

5.1.2 Elimination of Equillibrium Reactions

We can use any of the techniques described in Section 4.2 to eliminate the equillibrium reaction from (5.1.1). To keep a clear presentation, we use LU factorisation with manual pivoting, letting H^+ be the secondary species. Then, the matrix used for eliminating the equillibrium rates becomes

$$\mathbb{M} = \mathbb{C}^{-1} \begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ 1 & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix} \mathbb{C},$$

where only the nonzero elements are shown. $\mathbb{C}$ is the concentration scaling matrix introduced in Section 4.2.1. Let $\mathbb{M}_1$ be the first row of $\mathbb{M}$, and $\mathbb{M}_2$ the other rows, such that

$$\mathbb{M} = \begin{bmatrix} \mathbb{M}_1 \\ \mathbb{M}_2 \end{bmatrix}.$$

Using $\mathbb{M}_2$ to eliminate the equillibrium rates, we obtain the reduced system

$$\mathbf{0} = \mathbb{S}^{aq}_{eq} \ln \mathbf{y}_{aq} \tag{5.1.3a}$$

$$\mathbb{M}_2 \frac{\partial \mathbf{y}}{\partial t} = -\mathbb{M}_2 \left(\frac{\mathbf{u}}{\phi} \cdot \frac{\partial \mathbf{y}}{\partial x} \right) + \mathbb{M}_2 \mathbb{T}_{kin} \mathbf{q}_{kin}(\mathbf{y}). \tag{5.1.3b}$$

We can write this system of equations more compactly as

$$\mathbb{E} \frac{\partial \mathbb{M}\mathbf{y}}{\partial t} = -\frac{u}{\phi} \mathbb{F} \frac{\partial \mathbb{M}\mathbf{y}}{\partial x} + \mathbb{R}\mathbf{q}(\mathbf{y}),$$

where we have defined the diagonal matrices $\mathbb{E}$ and $\mathbb{F}$ as

$$\mathbb{E} = \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

$$\mathbb{F} = \begin{bmatrix} 0 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 0 & & \\ & & & & & & 0 & \\ & & & & & & & 0 \end{bmatrix},$$

the matrix $\mathbb{R}$ as

$$\mathbb{R} = \begin{bmatrix} \mathbb{I} & \mathbb{0} \\ \mathbb{0} & \mathbb{M}_2\mathbb{T}_{kin} \end{bmatrix},$$

and the function $\mathbf{q}$ as

$$\mathbf{q}(\mathbf{y}) = \begin{bmatrix} \mathbb{S}_{eq}^{aq} \ln \mathbf{y}_{aq} \\ \mathbf{q}_{kin}(\mathbf{y}) \end{bmatrix}.$$

Finally, the most compact form of (5.1.3) is obtained by using the substitution $\mathbf{v} = \mathbb{M}\mathbf{y}$, which results in the equation

$$\mathbb{E}\frac{\partial \mathbf{v}}{\partial t} = -\frac{u}{\phi}\mathbb{F}\frac{\partial \mathbf{v}}{\partial x} + \mathbb{R}\mathbf{q}\left(\mathbb{M}^{-1}\mathbf{v}\right). \tag{5.1.4}$$

At this point, we could have chosen to apply the techniques of 4.2 to reduce the condition number of the system's Jacobian matrix. However, for this test problem, the reaction rates are of similar magnitude, and the concentration differences are not too large. Thus, for the crude error tolerances we are considering, the integration algorithms converge satisfactorily if a simple row scaling of the Jacobian is used during integration.

5.1.3 Simple Operator Splitting

We first choose to solve the equation using a simple operator splitting scheme, where (5.1.4) is split into one pure transport step, and one pure reactive step. Applied to our problem, the transport step is then given by

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{u}{\phi}\mathbb{F}\frac{\partial \mathbf{v}}{\partial x},$$

and the reactive step is

$$\mathbb{E}\frac{\partial \mathbf{v}}{\partial t} = \mathbb{R}\mathbf{q}\left(\mathbb{M}^{-1}\mathbf{v}\right).$$

There is no need to apply the mass matrix $\mathbb{E}$ to the transport step, as the algebraic variable is not affected by this step anyway.

To solve the first substep, we use the upwind method for spatial discretisation, and the explicit Euler method for temporal integration. This results in the scheme

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \frac{u\Delta t}{\phi\Delta x}\cdot\mathbb{F}\left(\mathbf{v}_{i-1}^n - \mathbf{v}_i^n\right),$$

where subscripts denote the grid cell, superscripts denote the iteration index, and Δt is the step length. The dimensionless quantity $C = u\Delta t/\phi\Delta x$ is called the *Courant number*. Since our problem has a constant advection velocity, we can actually solve the step exactly by choosing Δt such that $C = 1$. With this choice, the method reduces to

$$\mathbf{v}_i^{n+1/2} = \left(\mathbb{I} - \mathbb{F}\right)\mathbf{v}_i^n + \mathbb{F}\mathbf{v}_{i-1}^n.$$

For the reaction substep, we choose the implicit Euler method. Since this substep is spatially decoupled, we can solve for each grid block separately,

$$\mathbb{E}\mathbf{v}_i^{n+1} = \mathbb{E}\mathbf{v}_i^{n+1/2} + \Delta t\,\mathbb{R}\mathbf{q}\left(\mathbb{M}^{-1}\mathbf{v}_i^{n+1}\right). \tag{5.1.5}$$

This is an implicit relation for $\mathbf{v}_i^{n+1}$, and must therefore be solved using iterative techniques. In grid cells where advection causes large concentration changes, the number of iterations will be large, but in cells where the change is small, only a few iterations is required. On parallel computers, the integration of each grid cell can be performed on separate processing units. In that case, the computational cost for this step is limited by the cell that requires the largest number of iterations.

We used the scheme described above, with a spatial resolution of 50 grid points, to simulate the evolution of the system for 20 years, given an advection velocity of $u = 0.1\,\mathrm{m/yr}$. In Figure 5.1.1, the distributions of CO_2, Dawsonite and Albite are shown at three different times, scaled by their respective initial concentrations. As evident from the figure, Dawsonite starts to dissolve once water with a low CO_2 concentration is introduced to the system. This causes Dawsonite-bounded carbon to be released into the infiltrating brine, rising its CO_2 content. On the other hand, some of the dissolved Dawsonite re-precipitates as Albite behind the front. These results are consistent with the findings of [21], from which this test case is constructed. The discrepancies that exist, are most likely due to the simplified mineralogy and activity model.

5.1.4 Numerical Diffusion

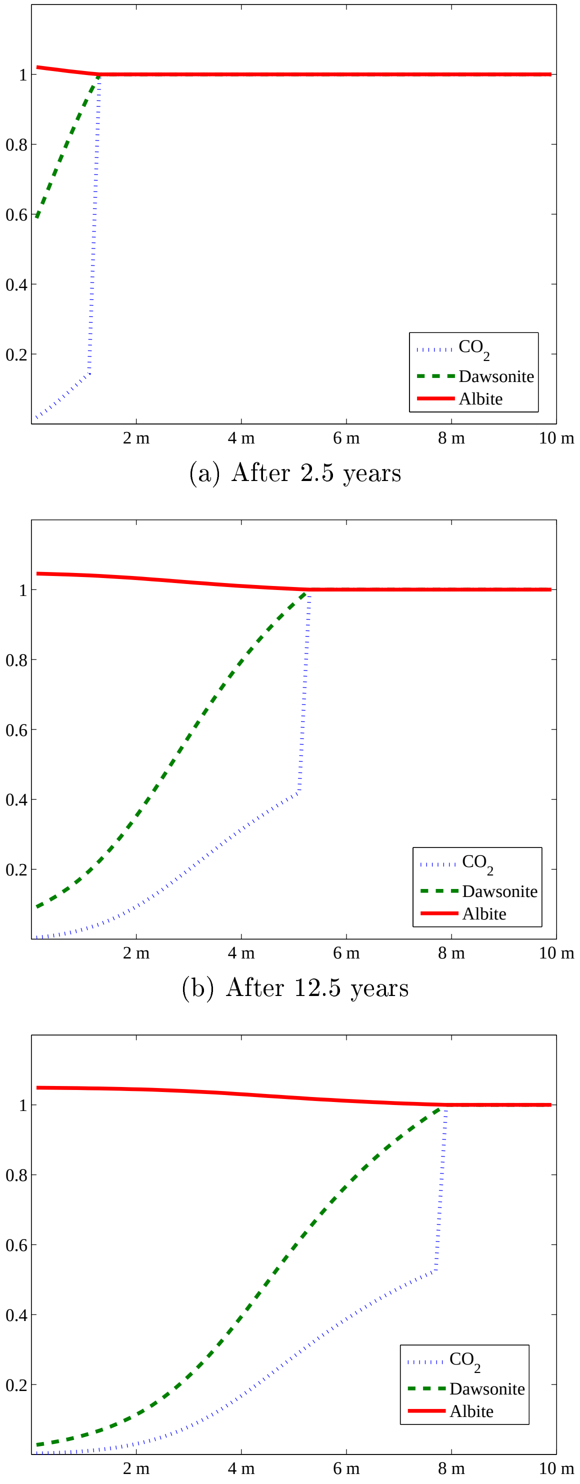
In our test problem, the difference between the initial concentrations and the inflow concentrations is large for some of the species. This is a commonly occuring situation in reaction-advection systems, and causes sharp concentration fronts to develop where the infiltrating fluid displaces the fluid originally present. It is well-known that sharp fronts are difficult to resolve numerically. In Section 5.1.3, we circumvented this problem by using an exact solution operator for the transport step. If the upwind discretisation scheme is used for non-constant velocities, however, the step length must be chosen such that $C < 1$ in all grid cells, for stability reasons. Thus, in many regions of the computational domain, C will be significantly smaller than 1. This introduces artificial diffusion to the solution, which destroys the solution accuracy near the front.

To illustrate the significance of numerical diffusion, we re-solved (5.1.4) using the same scheme as in the previous section, but with smaller time steps, corresponding to Courant numbers smaller than 1. The result is shown in Figure 5.1.2, along with the nondiffusive solution. We clearly see that the sharp CO_2 front is largely smeared out, even when the Courant number is close to 1. The concentration of Dawsonite, however, is not nearly as much affected, partially because the true solution of this concentration component is smoother than for CO_2.

One way of improving the resolution of the discontinuous front, is to use a second order discretisation scheme. For instance, consider the Lax-Wendroff scheme, which is given by

$$\mathbf{v}_i^{n+1/2} = \mathbf{v}_i^n + \frac{C}{2} \cdot \mathbb{F} \left(\mathbf{v}_{i+1}^n - \mathbf{v}_{i-1}^n \right) + \frac{C^2}{2} \left(\mathbf{v}_{i+1}^n - 2\mathbf{v}_{i+1}^n + \mathbf{v}_{i-1}^n \right).$$

This scheme is, however, known to introduce nonphysical oscillations to the solution. A straightforward application of this method may therefore cause the computed solution to become negative in some parts of the computational domain. Since the rate functions are undefined for negative concentrations, we must use a logarithm

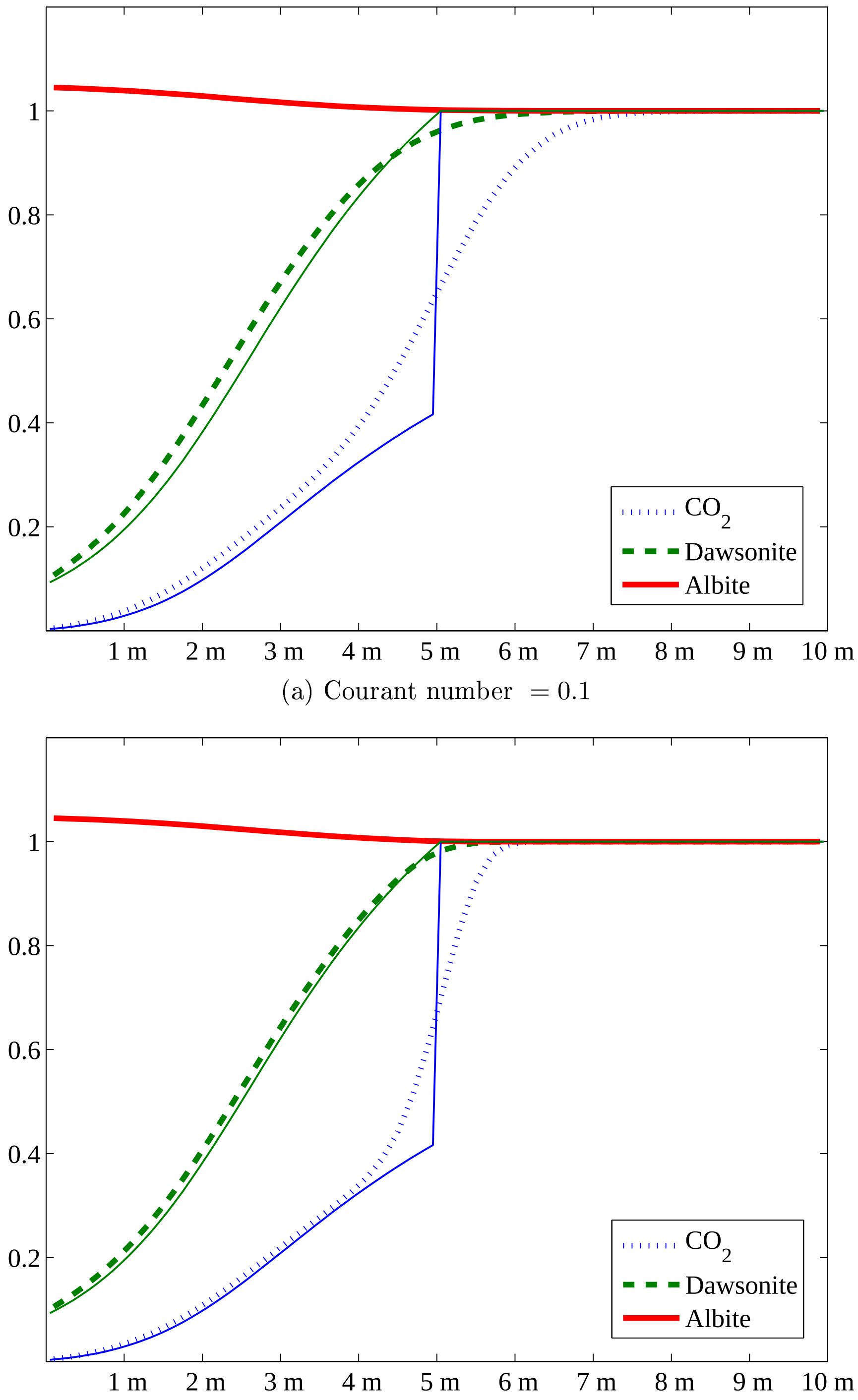


(a) After 2.5 years

(b) After 12.5 years

(c) After 19 years

Figure 5.1.1: Evolution of the concentrations



(a) Courant number $= 0.1$

(b) Courant number $= 0.8$

Figure 5.1.2: Artificial diffusion, upwind scheme

transformation to apply the method. Specifically, we use the change of variables $\boldsymbol{\ell} = \ln \mathbb{M}^{-1}\mathbf{v}$. The transport substep is then simply given by

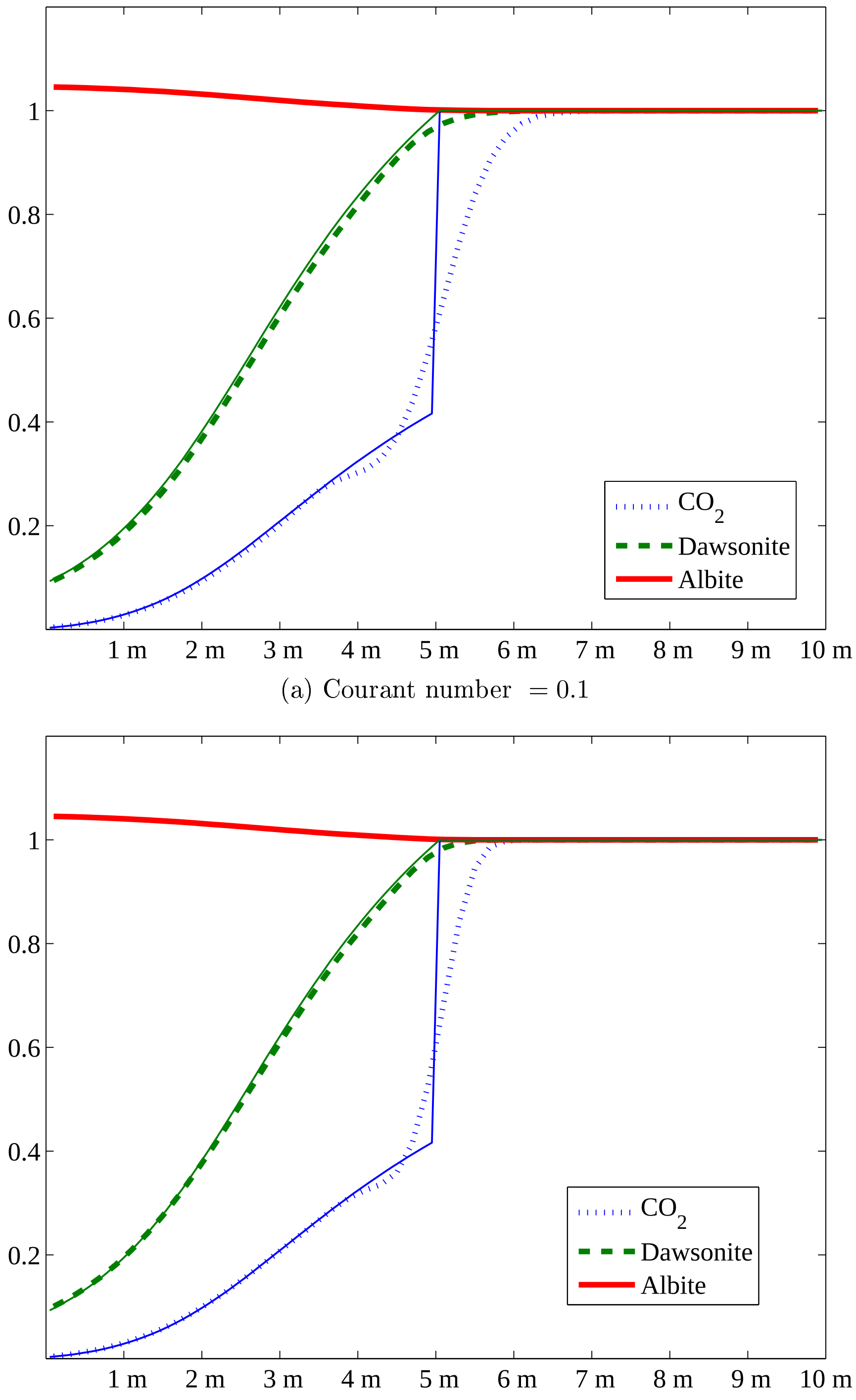
$$\frac{\partial \boldsymbol{\ell}}{\partial t} = -\frac{u}{\phi}\mathbb{F}\frac{\partial \boldsymbol{\ell}}{\partial t},$$

and the Lax-Wendroff discretisation can be applied directly. In Figure 5.1.3, the result of this approach is shown, along with the reference solution. We see that the front has become sharper, but some nonphysical oscillations are forming behind the front.

If we consider the concentration of Dawsonite specifically, the Lax-Wendroff scheme performs better than the upwind scheme for small Courant numbers, but the errors are of the same magnitude when the Courant number is larger. Since the Lax-Wendroff method is a second order scheme, it will invariably perform better than the upwind scheme when the grid resolution is increased. To demonstrate this, we re-solved the equation with increasing grid resolutions, with both the upwind and Lax-Wendroff discretisation, and measured the error of the Dawsonite component. As a reference, we solved the equation with a grid resolution of 256 points, using the exact solver for the transport substep. The results are shown in Figure 5.1.4. We remark that the upwind method gives the best performance for sparser grids, since the logarithm transformation is not required. At higher resolutions, however, the second order method is the most accurate.

5.1.5 Strang Splitting

In the previous sections, we solved the mass conservation equation using a simple sequential splitting scheme. As described in Section 4.1.2, it may be better to solve the mass conservation equation using the symmetric Strang splitting. With this procedure, the equation is solved using one half transport step, one full reaction step, and then yet another half transport step. To illustrate how this scheme can improve the accuracy of the solution, we solved (5.1.4) with a time step of $\Delta t = 2.5\,\mathrm{yr}$. Thus, the splitting interval is large, and the nature of the splitting errors becomes clearly visible. We used an exact solver for the transport step, and the implicit Euler method for the reaction substep, just as in Section 5.1.3. The computed solution with and without symmetric splitting is shown in Figure 5.1.5, along with the reference solution. Since the splitting interval is so large, both solutions have a jagged shape. Just behind the front, the dissolution rate of Dawsonite is large, and the amount of dissolved mineral is overestimated by the splitting algorithm. The opposite effect is seen at the inflow end, where the dissolution rate is slower.



(a) Courant number $= 0.1$

(b) Courant number $= 0.8$

Figure 5.1.3: Artificial diffusion, Lax-Wendroff scheme

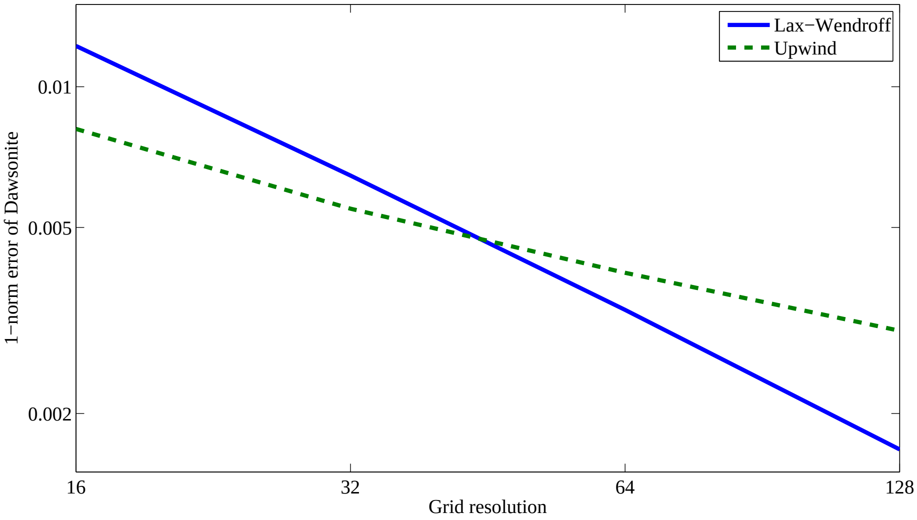
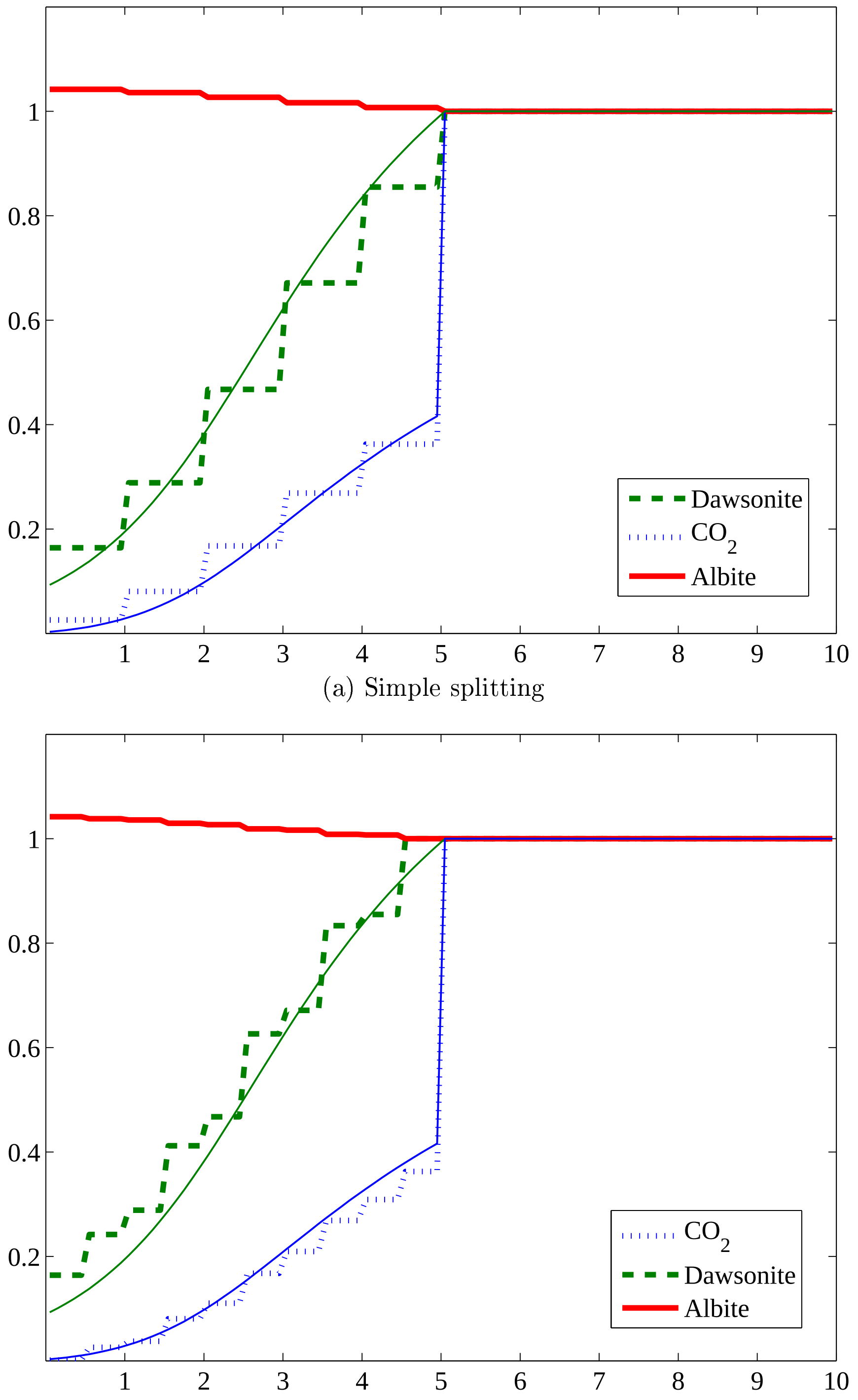


Figure 5.1.4: Numerical diffusion error for $C = 0.8$

Both the symmetric and nonsymmetric splitting schemes perform better in the presence of moderate numerical diffusion, as this smears out the jagged shape of the solution curves. To illustrate, we replaced the exact transport solver with the diffusive upwind scheme, using a time step corresponding to $C = 0.8$. The result is shown in Figure 5.1.6, where the improved accuracy is evident. Also, we repeated this procedure for different splitting intervals, and calculated the error of the Dawsonite concentration in each case. In Figure 5.1.7, the average grid cell error is plotted against the splitting interval. We see that the symmetric scheme yields the best performance, although the convergence is of first order for both methods, except at small splitting intervals. This is consistent with the findings of Sportisse[57], who discovered that the Strang splitting scheme may be of only first order when applied to stiff equations.

5.1.6 Integration of Stiff Terms

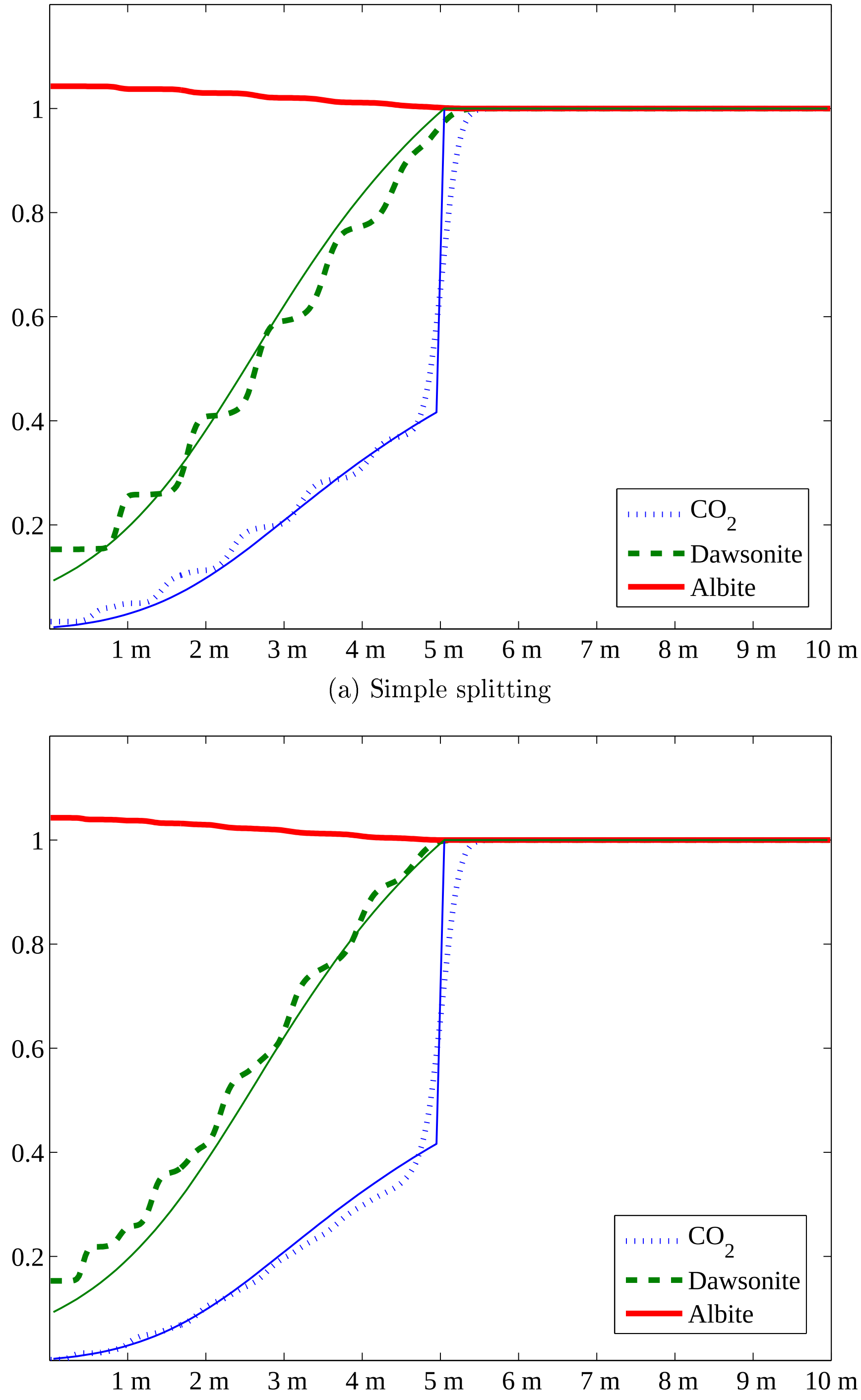
Until now, we have only chosen to integrate the reactive term using the implicit Euler method. As we have seen in Section 3.3, implicit integration methods of higher order also exists. To quantify the errors associated with choosing a low order integration scheme, we solved our test problem with the same basic setup as in Section 5.1.3,



(a) Simple splitting

(b) Strang splitting

Figure 5.1.5: Splitting errors, exact transport step



(a) Simple splitting

(b) Strang splitting

Figure 5.1.6: Splitting errors, diffusive transport step

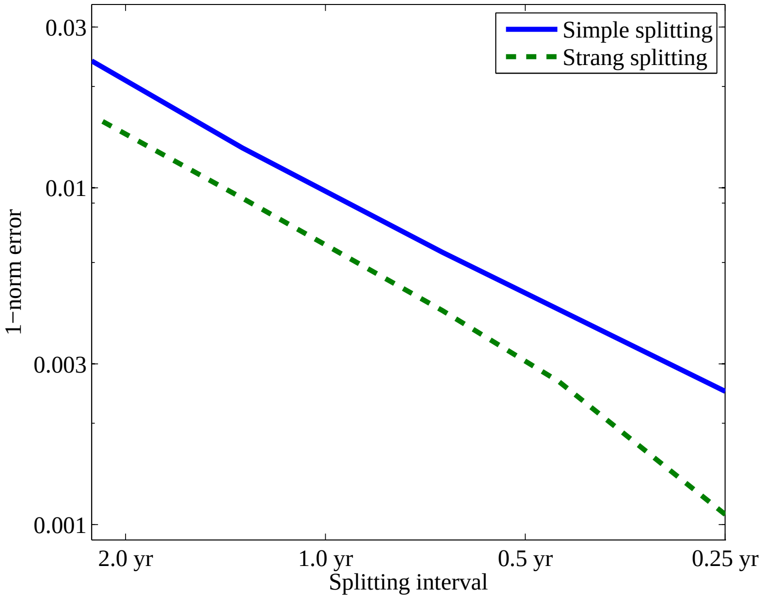


Figure 5.1.7: Splitting error for Dawsonite

but with different discretisation schemes for the reactive term. Instead of integrating the term using a single step, we subdivided the reaction step into several smaller integration intervals, increasing the accuracy. Also, we solved the step using both the implicit Euler method and the higher-order SDIRK methods from Table 3.3.2, disregarding their error estimation stages. In Figure 5.1.8, the resulting errors for the Dawsonite component is shown. The SDIRK4 method was used as a reference, and is not present in the figure. We see that the errors for the Euler method is of the same magnitude as the splitting and spatial discretisation errors. Also, the accuracy of the Euler method increases only slightly if more steps are used. The errors are much smaller for the other methods. For instance, it would require approximately 100 Euler iterations to acheive the accuracy of a single SDIRK2 step. However, this level of accuracy might not be needed, as other parts of the numerical solution introduce errors that are of greater importance.

5.1.7 Fully Coupled Solution

In all the previous sections, we have solved (5.1.4) using a sequential approach, solving for transport and reaction separately. We now try and measure the performance of the coupled approach. To do this, we first discretise (5.1.4) in space without

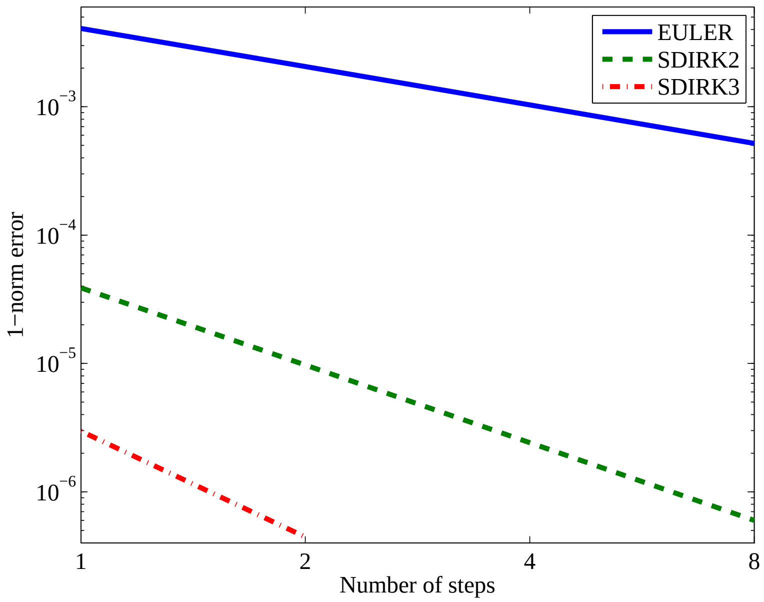


Figure 5.1.8: Reactive term error for Dawsonite

removing the reaction term. This results in the DAE system

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix}\mathbb{E} & & & \\ & \mathbb{E} & & \\ & & \ddots & \\ & & & \mathbb{E}\end{bmatrix}\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \vdots\\ \mathbf{v}_N\end{bmatrix} &= \frac{u}{\phi\Delta x}\left(\begin{bmatrix}-\mathbb{F} & & & \\ \mathbb{F} & -\mathbb{F} & & \\ & \ddots & \ddots & \\ & & \mathbb{F} & -\mathbb{F}\end{bmatrix}\begin{bmatrix}\mathbf{v}_1\\ \mathbf{v}_2\\ \vdots\\ \mathbf{v}_N\end{bmatrix} + \begin{bmatrix}\mathbb{F}\mathbf{v}_0\\ \mathbf{0}\\ \vdots\\ \mathbf{0}\end{bmatrix}\right)\\ &\quad + \begin{bmatrix}\mathbb{R} & & & \\ & \mathbb{R} & & \\ & & \ddots & \\ & & & \mathbb{R}\end{bmatrix}\begin{bmatrix}\mathbf{q}\left(\mathbb{M}^{-1}\mathbf{v}_1\right)\\ \mathbf{q}\left(\mathbb{M}^{-1}\mathbf{v}_2\right)\\ \vdots\\ \mathbf{q}\left(\mathbb{M}^{-1}\mathbf{v}_N\right)\end{bmatrix}\end{aligned} \tag{5.1.6}$$

where the subscripts once again denote the grid cell number, and $\mathbf{v}_0$ is the value of $\mathbf{v}$ at the inflow boundary. We can now solve the system directly using any of the implicit integration methods from Chapter 3. The computed solution will be plagued by numerical diffusion, however, since the transport part is solved implicitly. In Figure 5.1.9, the solution is shown for a grid resolution of 50 cells, along with the reference solution computed by operator splitting. It is evident that numerical diffusion dominates the error of the solution. As we have seen, numerical diffusion can be partially combated by increasing the spatial grid resolution. In Figure 5.1.10,

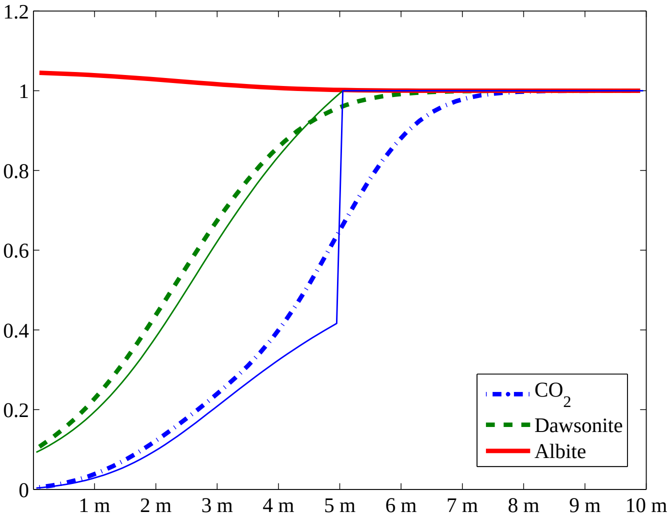


Figure 5.1.9: Implicit solution, 50 grid points

we have plotted the error of the Dawsonite component for different grid resolutions, using a step length of $\Delta t = 1.2\,\text{yr}$. Only results for the Euler and SDIRK2 methods are shown, as the results for SDIRK3 and SDIRK4 were similar to that of SDIRK2. We clearly see that the errors are reduced when the higher-order integration schemes are applied. However, the accuracy improvement is not as large as in Figure 5.1.8, since the diffusion error dominate in the solution.

Compared to the sequential solution methods, the nonlinear systems that must be solved when using the coupled approach, are very large. This disadvantage is, however, not as severe as it may seem. From equation (5.1.6), it is clear that the Jacobian of the system will be sparse and banded. Therefore, the systems are solved quite efficiently if sparse techniques are used in a clever way. Also, the increased stability of the coupled method allows larger time steps to be taken, and faster convergence of the Newton iterations. Whether or not the coupled method can compete with operator splitting, therefore depends on a lot of different factors, such as the smoothness of the solution, the efficiency of the sparse solver, the possibilities for parallelisation, and the programming environment in which the algorithms are implemented. A direct comparison is not possible without doing a range of optimised benchmark tests. We will therefore not pursue this topic any further.

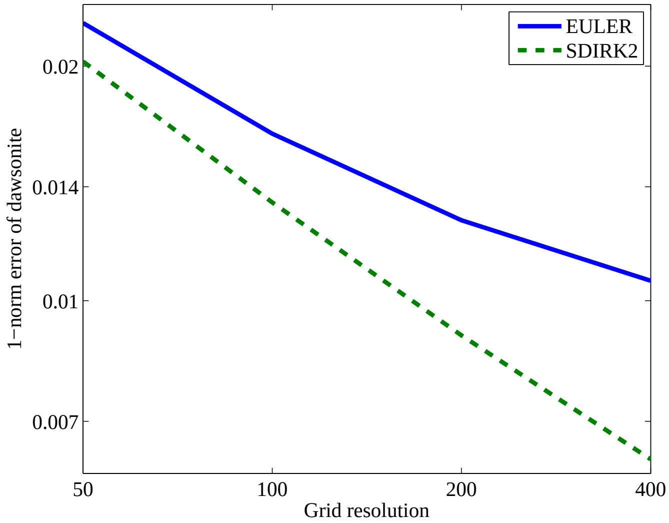


Figure 5.1.10: Convergence of the coupled method

5.1.8 Condition Number of the Jacobian

In Section 4.2.2, we stated that linear recombinations of the mass conservation equations could ease the condition number of the system's Jacobian matrix, which is important if the equation is to be solved implicitly. To demonstrate this, we calculated the condition number of the Jacobian before and after these techniques were used. Specifically, we considered the Jacobian for the implicit Euler method, applied to the fully coupled mass conservation equation (5.1.6), which is given by

$$\mathbb{J} = -\begin{bmatrix} \mathbb{E} & & & \\ & \mathbb{E} & & \\ & & \ddots & \\ & & & \mathbb{E} \end{bmatrix} + \frac{u\Delta t}{\phi \Delta x}\begin{bmatrix} -\mathbb{F} & & & \\ \mathbb{F} & -\mathbb{F} & & \\ & \ddots & \ddots & \\ & & \mathbb{F} & -\mathbb{F} \end{bmatrix} \\ + \Delta t \begin{bmatrix} \mathbb{R} & & & \\ & \mathbb{R} & & \\ & & \ddots & \\ & & & \mathbb{R} \end{bmatrix} \begin{bmatrix} \frac{\partial \mathbf{q}_1}{\partial \mathbf{v}_1} \\ \vdots \\ \frac{\partial \mathbf{q}_N}{\partial \mathbf{v}_N} \end{bmatrix} \mathbb{M}^{-1}.$$

To demonstrate the effect of stiffness on the condition number, we let the value of u vary, and set the integration time scale to match the advection. That is, we adjust the time step such that $u\Delta t$ always stays at the same value as in the previous

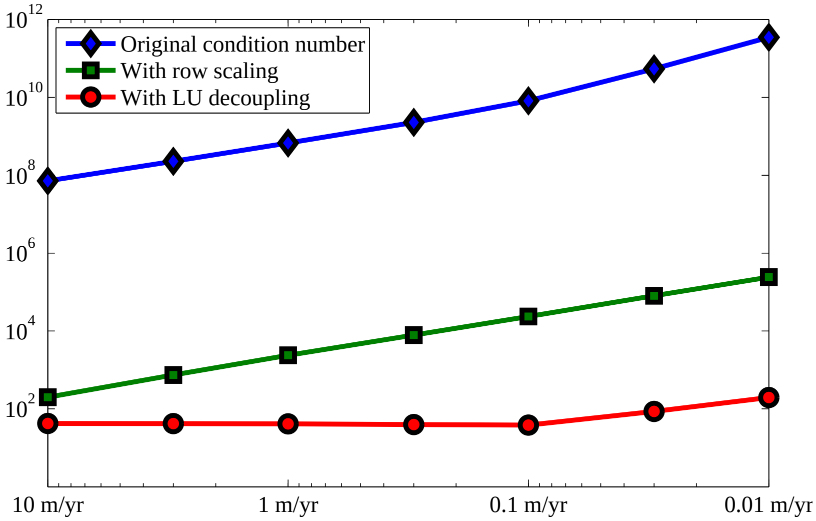


Figure 5.1.11: Condition number of Jacobian

section. For large values of u, the system is dominated by advection, and for small u values, the reaction term is dominant. The resulting condition numbers of $\mathbb{J}$, for u between 10 m/yr and 0.01 m/yr, are shown in Figure 5.1.11. We see that the condition numbers become large very quickly. Indeed, if no methods are used to reformulate the system, $\mathbb{J}$ may be so ill-conditioned that the implicit integration methods are unable to converge. A simple row scaling, where the rows of $\mathbb{J}$ are scaled according to their maximum norm, is a great aid, and reduces the condition number to within acceptable limits. Even better conditioning is obtained if any of the reduction methods of Section 4.2 are used. In the figure, the result of using the automatically pivoted LU method is shown. This time, the condition number of the matrix is almost unaffected by the stiffness, and stays below 100 for the majority of the test interval. The results for the QR and SVD reduction techniques are similar, but slightly better.

5.2 The Utsira Mineralogy

To explore the behaviour of the numerical codes on a more complex scenario, we constructed another test case, involving an extensive mineralogy, and two-phase, two-dimensional flow. Specifically, we wanted to investigate how the codes performed when simulating a bouyantly migrating CO_2 plume. For this purpose, we used the

Mineral	Initial volume fraction
Calcite	0.039
Magnesite	minimal
Siderite	minimal
Dawsonite	minimal
Albite	0.02
K-Feldspar	0.085
Quartz	0.488
Chalcedony	minimal
Kaolinite	minimal
Chlinochlore	0.003
Daphnite	0.003
Muscovite	0.006
Phlogopite	minimal
Annite	0.006
Labradorite	minimal
Gibbsite	minimal

Table 5.2.1: Initial volume fractions

open-source Matlab Reservoir Simulation Toolbox (MRST) developed at SINTEF ICT and available at http://www.sintef.no/Projectweb/MRST/. We used an initial distribution of CO_2 as shown in Figure 5.2.1, and assumed that no fluid could pass through the boundaries of the computational domain (Neumann conditions). The pressure at the top of the reservoir was set to a constant value of 100 bar. In a more realistic case, the reservoir would extend much further in the lateral direction, and the CO_2 plume would have originated from a well source. However, our intention is not to calculate the plume propagation accurately, but rather to test the interaction between numerical codes for chemistry and advective transport. In this respect, the test case described above is very illustrative.

As explained in Section 1.2, the volume of a two-phase CO_2/brine mixture is significantly decreased when gaseous CO_2 dissolve into the brine. Also, the volume of gaseous CO_2 is influenced by pressure changes. Thus, theory suggests that we should use the pressure equation for compressible fluids (1.2.3) to find the advection velocities. Since accurate advection velocities are not important for us, however, we use the approach of Obi and Blunt[43], and define the gaseous CO_2 phase as incompressible. We then obtain a simpler set of equations that is faster to solve. The density of the gas phase is set to $0.23\,\mathrm{kg/m^3}$, and the density of the aqueous phase is set to $1\,\mathrm{kg/m^3} + c_{\mathrm{CO_2(aq)}} \cdot \mathrm{dm^3/mol} \cdot 0.01\,\mathrm{kg/m^3}$, where $c_{\mathrm{CO_2(aq)}}$ is the concentration of CO_2 dissolved in the brine. We let the relative permeabilities of the gas and aqueous phase be given by the simple analytical expressions

$$\begin{aligned} k_g^{rel} &= \frac{1}{2}\left(0.7 - \min(S,\, 0.7)\right)^2 \\ k_l^{rel} &= S^2, \end{aligned}$$

while the rock permeability is set to 10 mD uniformly across the reservoir. If we were to calculate the plume migration speed more accurately, we could have used experimentally determined expressions like the ones found in[6], but approximate values are sufficient for our purposes. The viscosity of the aqueous and gaseous phase was set to 1 cP and 0.03 cP, respectively. Alternatively, these values could have been calculated from an equation of state.

The geochemical reactions that we include in the model, are the same as those described in Table 2.1.2. The initial mineral concentrations are taken from [22], and are echoed in Table 5.2.1 for easier reference. “Minimal” volume fractions correspond to a bulk concentration of $0.001\,\mathrm{mol/dm^3}$, at which the dissolution rate was assumed to be zero. The mineral assemblage resembles that of the Utsira formation, into which Statoil has injected captured CO_2 since 1996[32]. For simplicity, we assume constant activity coefficients for all the species in the model. In particular, we assume that noncharged species have unit activity coefficients, monovalent ions have an activity coefficient of $\gamma = e^{-0.5}$, divalent ions have $\gamma = e^{-2}$ and trivalent ions have $\gamma = e^{-4.5}$. This is a crude activity model, so the we expect the concentrations we calculate to be somewhat inaccurate. Nevertheless, the convergence properties of the numerical methods will be the same as if more precice values where used.

5.2.1 Evolution of the System

We solved the system depicted above using a spatial resolution of 32×32 grid blocks. The flow equations were solved in a sequential manner. First, we solved for pressure, obtaining a pressure gradient that was used to calculate the advection velocities. Secondly, the mass conservation equation was solved, using Strang splitting to separate reaction from advection. The advective part of the equation was solved using a simple explicit upwind scheme, while the reaction part was solved using the ESDIRK32 scheme from Table 3.3.3. To be able to integrate this very stiff geochemical system, we also used the SVD reduction technique that was introduced in Section 4.2.5. Finally, the temperature was assumed to be constant and equal to 80 °C during the entire simulation period.

For the first 5 years of the simulation, we used a splitting interval of 0.25 years. The gas plume migration pattern during this period is shown in Figure 5.2.1. In this period, bouyancy forces act on the CO_2 plume, pushing it towards the top of the aquifer. While migrating upwards, the CO_2 plume dissolves into the aquifer brine, as shown in Figure 5.2.2. After approximately 5 years, the gas plume becomes immobilised due to residual trapping. Thus, we observe very few changes in the saturation from this point onwards, and the splitting interval can be increased. Continuing the simulation another 200 years, we see that the CO_2-rich brine begins to sink down towards the bottom of the reservoir, due to bouyancy effects. This slower migration process helps distribute the CO_2 over a larger area. At the same time, precipitating minerals capture some of the CO_2 that has been injected.

5.2.2 Mineral Reaction Paths

Significant changes in the mineral composition is seen in the grid cells approximately 1 year after CO_2 has been introduced. In Figure 5.2.3, we have displayed how the concentrations of different minerals will typically evolve. The figure is obtained by

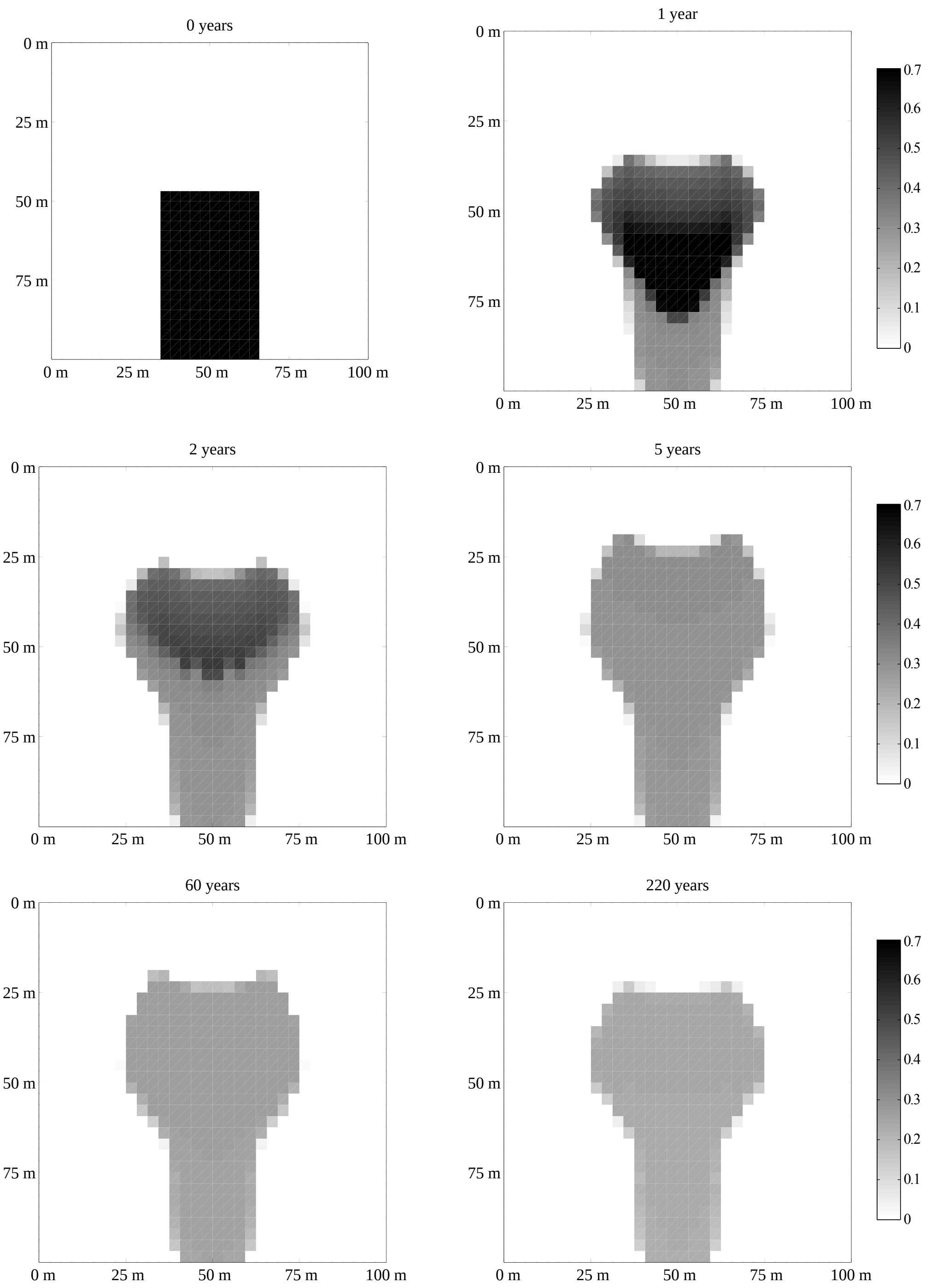


Figure 5.2.1: Saturation of CO_2(g)

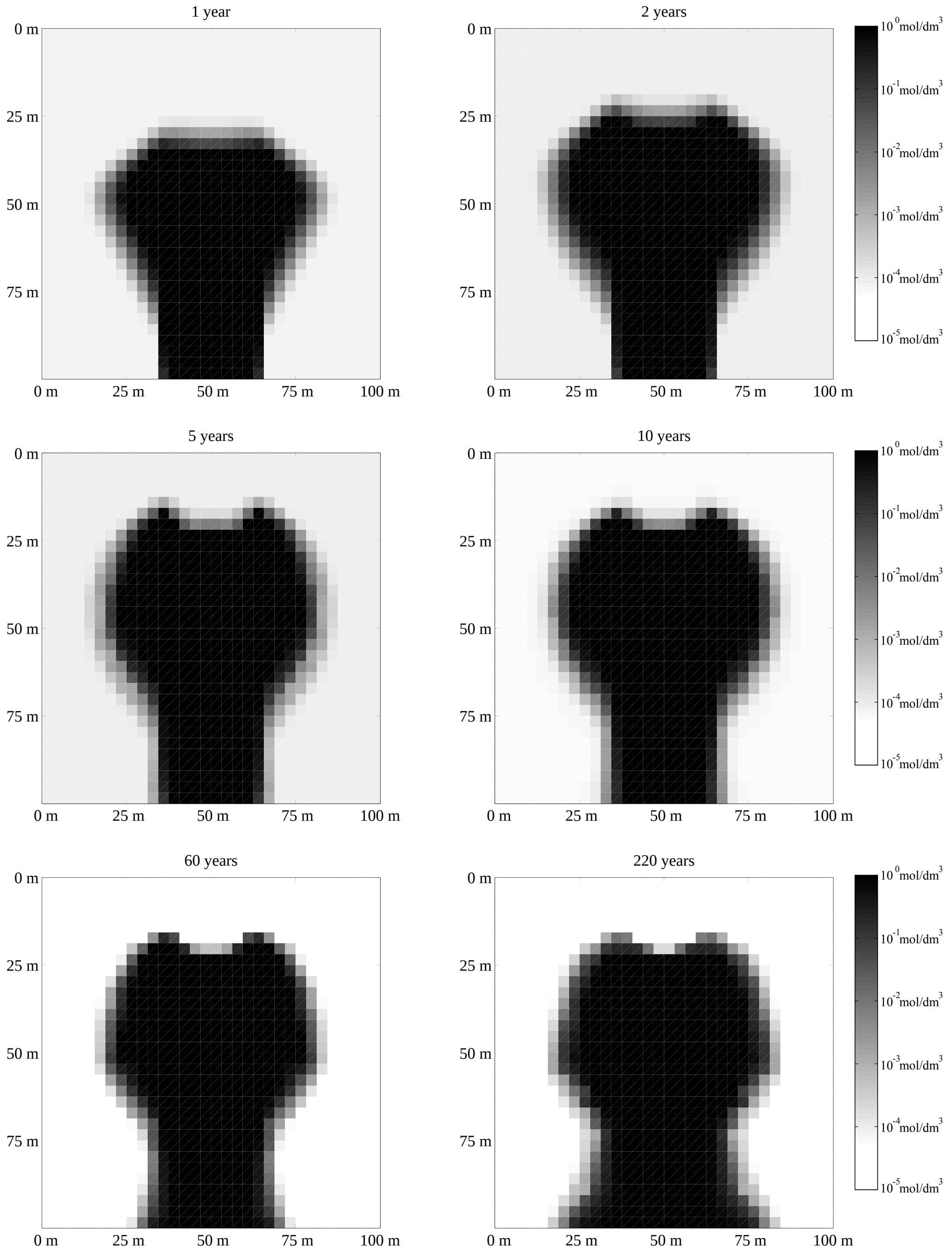


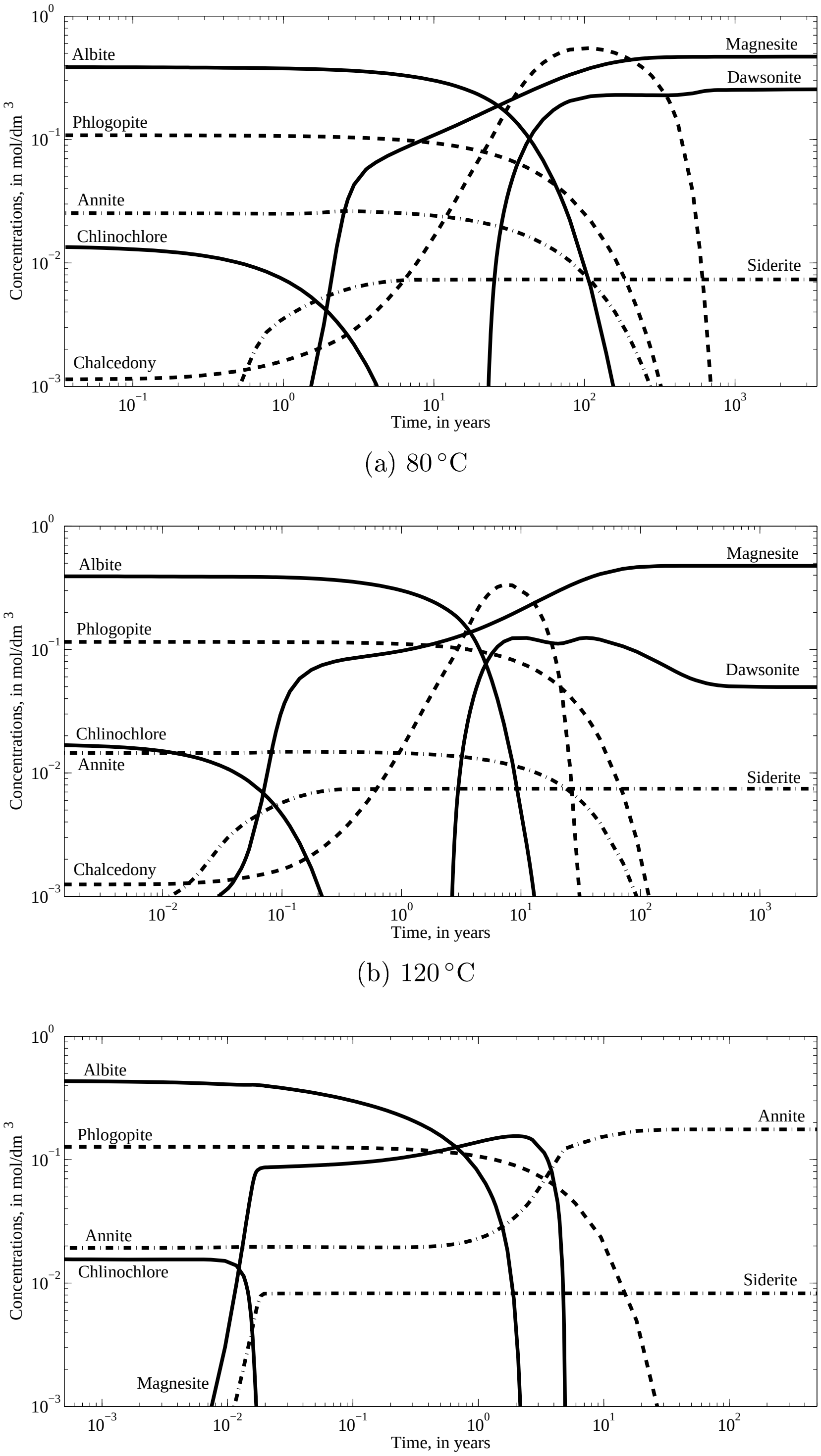
Figure 5.2.2: Interphasial concentration of CO_2(aq)

simulating a grid cell that has the initial mineralogy of Table 5.2.1, but is then flooded by gaseous CO_2 at a pressure of 100 bar. To illustrate how the temperature may influence the chemical kinetics, we performed the calculation three times, using different temperatures. At 80 °C, we see that a significant amount of CO_2 becomes trapped by precipitation of Magnesite and Dawsonite, within a time frame of 50-100 years. If the temperature is increased to 120 °C, precipitation is faster, but the amount of stable Dawsonite is smaller than at 80 °C. At 200 °C, neither Magnesite nor Dawsonite are stable, and the initial amount of Albite is quickly dissolved, releasing more carbonate into the brine. This calculation illustrates that temperature changes can affect the stability of the minerals, as well as the time frame in which dissolution and precipitation occurs. Thus, temperature effects may be important when considering storage sites with a significant temperature gradient, or when considering the injection of CO_2 with higher temperature than the reservoir brine.

5.2.3 Influence of Chemical Reactions on Flow

The chemical reaction that has the biggest impact on advection velocities, is the dissolution of gaseous CO_2 into brine. This reaction alters the saturations of the mobile phases, which in turn changes their relative permeabilities. The other reactions in the system may influence this process implicitly, by transforming dissolved CO_2 into other chemical species. This leaves room for more gaseous CO_2 to dissolve. However, since the mineral dissolution reactions are slow, we do not expect them to significantly influence the gas saturation during the period where the gas plume is mobile. To test this hypothesis, we performed additional two-dimensional tests, one including only the equillibrium reactions, and another where all chemical reactions were disregarded. The results are shown in Figure 5.2.4. As we can see, the gas migration speed is practically unchanged when the kinetic reactions are removed. On the other hand, if the equillibrium reactions are removed as well, the plume is migrating much too fast. Thus, for most CO_2 injection scenarios, mineral reactions can probably be disregarded during the initial phase of the simulation. If this is done, the first years of the simulation can be computed within a fraction of the time that must be used otherwise. The equillibrium reactions, however, must be retained. Otherwise, the saturations and advection velocities are greatly affected.

Another way of which chemical reactions may potensially change the advection velocities, is by increasing or decreasing the porosity of the rock, due to dissolution or precipitation of minerals. To quantify this effect for the mineralogy we are considering, we calculated the permeability change according to the Kozeny-Carman

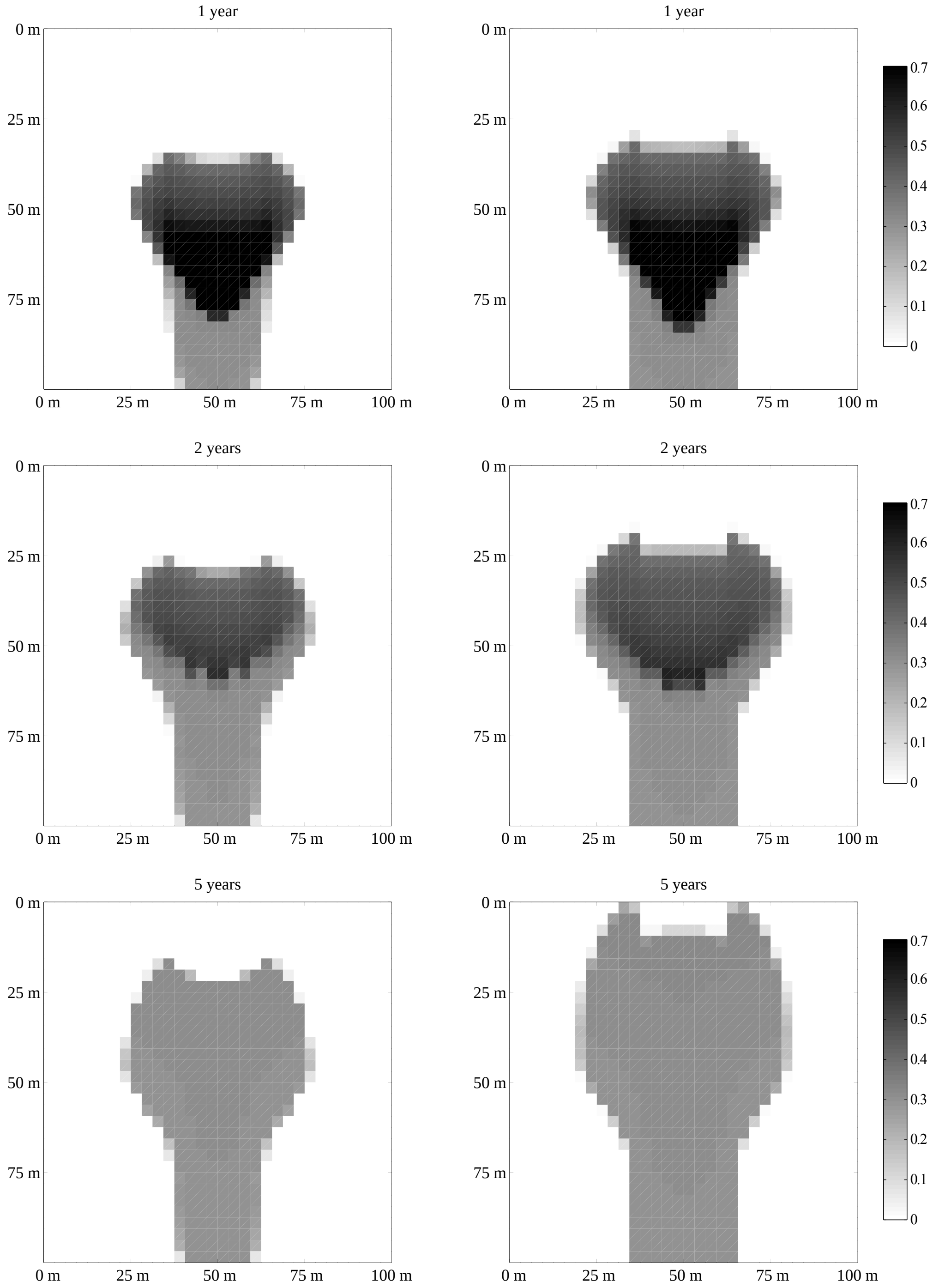


(a) 80 °C

(b) 120 °C

(c) 200 °C

Figure 5.2.3: Mineral evolution after a sudden CO_2 flooding



(a) Only equillibrium reactions

(b) No chemical reactions

Figure 5.2.4: Plume migration patterns

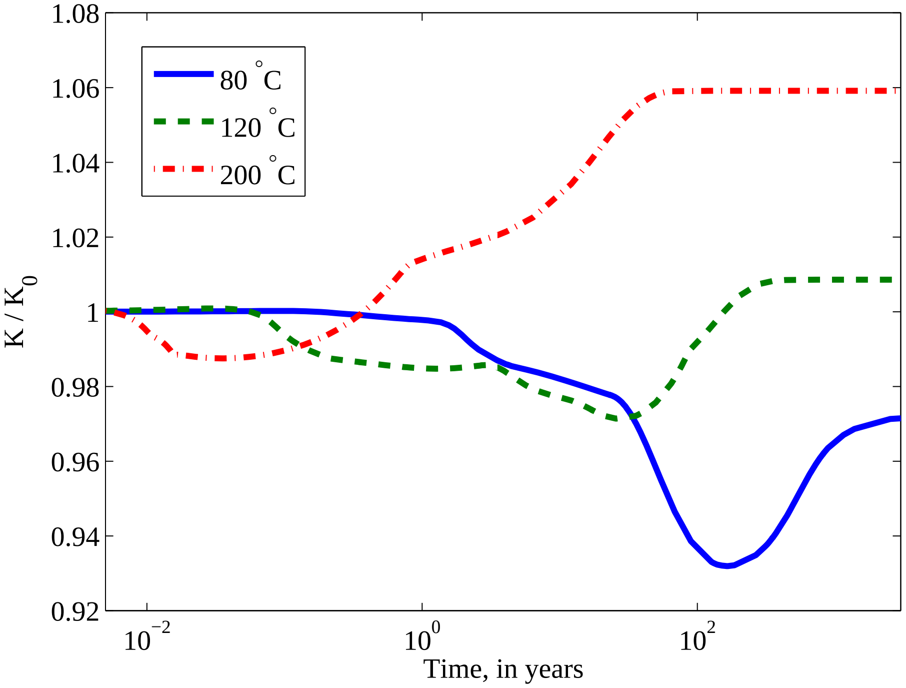


Figure 5.2.5: Change of rock permeability after CO_2 flooding

equation (1.3.4), for the grid cell considered in Section 5.2.2. The simulation was repeated for three different temperatures, and the result is seen in Figure 5.2.5. For high temperatures, more minerals dissolve, and the permeability increases. For lower temperatures, the increased CO_2 concentration causes minerals to precipitate, thereby decreasing the permeability. For the mineralogy we are considering, however, the changes are relatively small. Disregarding porosity effects may therefore be justified, as the uncertainty of other flow parameters is of greater importance.

5.2.4 Instabilities due to Transient Behaviour

For the complicated mineralogy we are considering in this example, stability is a major issue during integration of the reactive term. As evident from Figure 5.2.2, the area which is flooded with CO_2 increases during the entire integration interval. When CO_2 is introduced to a new region of the computational domain, the chemical properties of that region changes abruptly, and the concentrations of the aqueous species

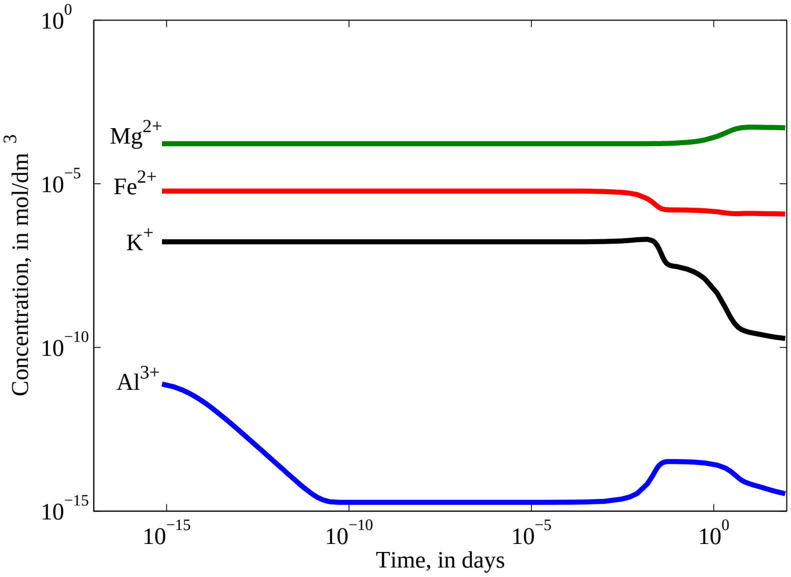


Figure 5.2.6: Typical evolution of concentrations when using Strang splitting

may change by orders of magnitude. Thus, it is difficult to choose a proper scaling for the variables involved, which affects the performance of the implicit methods. Since the concentration values at the end of the time step are so far from their initial values, the simplified Newton iteration may diverge, or require an excessive number of iterations. Therefore, integrating the reactive term in a single step, as we did in Section 5.1.3, is out of the question. In line with the recommendations of Hairer and Wanner[18], we chose to abort integration and reduce the time step whenever more than 10 iterations was required for convergence. Furthermore, we chose the initial time step to match the time scale of the fastest reaction. For every successful time step, we estimated the error of the ESDIRK32 method using the second last stage, which is an L-stable second-order approximation to the solution. We remark that even though the last stage of this method is implicit, it is cheap compared to the others. This is because the LU factorisation of the Newton iteration matrix is already available, and the second stage value can be used as a close starting value for the iteration. After a successful step, the next time step length was calculated based on (3.3.8), with a relative tolerance limit of 10^{-3}, and absolute tolerance limit of 10^{-5}, which is well below the experimental uncertainty level for typical geochemical parameters. A rescaling was performed whenever the change in concentrations or time step length was large. Otherwise, the algorithm failed to converge.

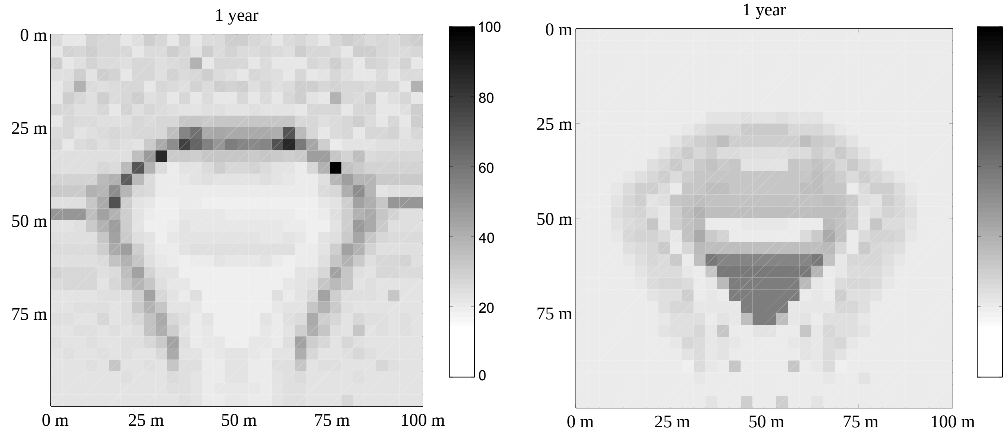
We performed a thorough analysis of the grid cells that required the most number of iterations for convergence. It was revealed that the concentration evolution in these cells often showed a curious transient behaviour, as in Figure 5.2.6. Here, we have plotted the calculated concentrations for four different metal ions in a single grid cell. The cell contains much CO_2 initially, but in the previously calculated transport step, it has been infiltrated with brine that has lower CO_2 content. Thus, at the beginning of the reactive step, the concentrations of the metal ions are far from their equillibrium values. Already within fractions of a second, the aluminium concentration starts to converge towards a value that represents equillibrium with the minerals and the higher-concentration ions. At this point, kinetic reaction rates are small compared to the time scale (10^{-15} days). Therefore, the metal ions with higher concentrations are not significantly affected.

Since the change in aluminium concentration is very large, small time steps must be taken to ensure convergence. After aluminium has equillibrated, the concentrations remain constant for some period, until the step length has reached about 10^{-2} days. At this time scale, the rates of the mineral reactions are so large that the concentrations of iron and potassium are affected. These quickly attain equillibrium with the minerals, and the sudden concentration drop requires the step length to be greatly reduced, resulting in a lot of failed steps. We also see that when the concentrations of iron and potassium changes, the equillibrium level for aluminium changes as well. Finally, after about 1 day, the concentration of magnesium starts to rise due to dissolution reactions. This affects the equillibrium values of potassium and aluminium, leading to large changes in these concentrations during the last part of the integration interval.

To summarise, the splitting scheme we have chosen, disturbs the initial chemical equillibrium of the cell during the transport substep, by introducing chemical species from neighbouring cells. To re-establish the equillibrium state, the cell undergoes a complex reaction path, which forces the integration algorithm to use many, small time steps in order to converge. This makes the solution of the reactive step extremely time-consuming.

Since the instabilities observed above are caused by "instantly" flushing the cell with components from neighbouring locations, we might try to alleviate the situation by flushing the cell "gradually" during the reactive step. This can be obtained by using a variant of the iterative splitting technique of Section 4.1.3, instead of the Strang splitting. Specifically, we first solve for advective transport,

$$\frac{\partial \mathbf{c}}{\partial t} = \boldsymbol{\mathcal{L}}(\mathbf{c}), \quad \mathbf{c}(0) = \mathbf{c}_0$$



(a) Strang splitting

(b) Iterative splitting

Figure 5.2.7: Number of LU factorisations per reactive step

using the explicit Euler method and upwind discretisation, as before. The difference between the initial and final concentrations, $\Delta\mathbf{c} = \mathbf{c}_{final} - \mathbf{c}_0$, is then used as a constant source term for the reactive step,

$$\frac{\partial \mathbf{c}}{\partial t} = \frac{\Delta \mathbf{c}}{\Delta t} + \mathcal{R}(\mathbf{c}), \quad \mathbf{c}(0) = \mathbf{c}_0.$$

After this step, integration is stopped. Thus, it corresponds to iterative splitting with only one iteration used. Not only does this scheme resemble the true physical situation more closely than the Strang splitting, the reactive step is much more stably computed as well, since the initial conditions correspond to chemical equillibrium. With this new scheme, the concentrations converge monotonically to their respective equillibrium values, and the time steps can be chosen much larger.

5.2.5 Integrating Over Discontinuities

We re-solved the equation using the new splitting scheme, and noticed a big improvement in the computational requirements. In Figure 5.2.7, we have shown the number of required LU factorisations per reaction step after 1 year, using both of the splitting techniques. However, there are still some cells that require a large amount of iterations. Analyses shows that the problem in this case is related to the integration of discontinuous functions. For instance, consider a cell that is gradually flushed with CO_2. Initially, the problem is well-behaved, and the variables are increasing or decreasing monotonically as CO_2 is introduced. At this point, the amount of the gaseous CO_2 component itself is at zero, since the gas is immediately dissolved. However, when the concentration of aqueous CO_2 reaches a certain point, the cell becomes saturated, and the amount of gaseous CO_2 starts to rise. This creates a discontinuity in the derivative of the solution.

As the integration methods we are using assumes that the problem is smooth, we will have big difficulties integrating past the discontinuity. First of all, the simplified Newton iterations may diverge, since the Jacobian matrix may be very different at saturated and subsaturated conditions. Secondly, the error estimation and step size control mechanism depend on the smoothness of the solution. Therefore, we experience a large number of unnecessary step rejections at the discontinuity.

As a first remedy to handle this problem, we switched from simplified to true Newton iterations, and disabled the error estimation device, whenever a lot of consecutive step rejections occured. Since the step length near the discontinuity is reduced to very small values, this approach does not contribute significantly to the numerical error of our solution. However, it is not an ideal solution, as the computational cost becomes very large whenever this situation occurs. A better way is to use some kind of event detection algorithm. For instance, one can use the *discontinuity locking* method, as described in [45].

We will briefly outline the idea behind the method. First, we remark that the function expressions used at one side of the discontinuities of our problem, is mathematically well-behaved (although physically incorrect) at the other side as well. For instance, consider a cell that is being flushed with gaseous CO_2. Let c_g and c_d be the concentrations of gaseous and dissolved CO_2, respectively. Initially, the correct expression describing chemical equillibrium is $c_g = 0$, that is, all the gas is dissolved into the brine. As CO_2 is introduced into the system, the level of c_d is rising, but c_g stays at zero. After saturation is reached, the physically correct equillibrium expression is $c_d = c_{sat}$, where c_{sat} is the maximum concentration of CO_2 the brine can dissolve. When applying a discontinuity locking mechanism, the initial equillibrium expression $c_g = 0$ is retained even *after* saturation has been reached. Thus, the calculated concentrations varies smoothly, although they attain unphysical (supersaturated) values past the saturation point. After the algorithm has sucessfully integrated past the discontinuity, interpolation techniques are then applied to locate the exact point where saturation was reached. The integration is then aborted, and restarted at the saturation point using the physically correct equillibrium expression.

5.2.6 Performance of Higher-order Methods

It is possible that higher-order integration schemes perform better than the ESDIRK32 method we have used this far. To test this hypothesis, we compared the performance of the ESDIRK methods of Table 3.3.3 on the situation that we have found to cost the most number of iterations, namely, the flushing of a grid cell with gaseous CO_2. To avoid complications arising from discontinuities, we only compared the performance on the smooth interval, prior to the saturation point. The results are shown in Table 5.2.2a. It is clear that the lower-order method perform best in this case, with respect to all the parameters. Another striking result of Table 5.2.2a is the large number of failed steps, for all the methods tested. The reason for the failed steps is the large change of concentrations seen when introducing CO_2 into a reservoir region. Thus, the step size is restricted by the convergence of the Newton iterations, rather than the accuracy of the solution. Since the lower-order method is also the most stable one, ESDIRK32 comes out ahead. The situation changes if the accuracy requirement is increased. To illustrate this, we re-solved the problem with a relative tolerance level of 10^{-6} and absolute tolerance of 10^{-8}, and printed the results in Table 5.2.2b. The higher order methods now requires the least number of function evaluations and LU factorisations to acheive a solution within the error tolerance. This high level of accuracy is not needed for geochemical simulations, however, since the uncertainty of geochemical parameters are usually of some percents or more.

	ESDIRK32	ESDIRK43	ESDIRK54
LU factorisations	64	82	96
Failed steps	24	33	40
Successful steps	40	49	56
Function evaluations	575	868	1435
Solutions of linear systems	593	896	1470

(a) RelTol $= 10^{-3}$, AbsTol $= 10^{-5}$

	ESDIRK32	ESDIRK43	ESDIRK54
LU factorisations	563	328	313
Failed steps	261	163	160
Successful steps	302	165	153
Function evaluations	3757	3003	4071
Solutions of linear systems	3829	3087	4177

(b) RelTol $= 10^{-6}$, AbsTol $= 10^{-8}$

Table 5.2.2: Performance of different integration schemes

Chapter 6

Summary and Conclusions

In this thesis, we have shown how to include chemical reactions in the common equations for multi-component, multi-phase flow in porous media. As we have seen, three major modifications must be made:

1. A source term must be added to the mass conservation equations, describing the transition from one component to another due to chemical reactions.

2. Since the components have different formation energies, a chemical reaction can rise or lower the temperature of the fluid. This can be described by adding a term to the temperature equation.

3. Reactions may change the saturations and the total volume of the pore fluid, thus affecting the pressure gradient and advection velocities. This effect can be captured by small modifications to the pressure equation.

The modification that poses the greatest numerical difficulties, is the first one. Without reactions, the mass conservation equations are easily scalable, and can be solved quickly by explicit integration methods. When chemistry is introduced, however, we face a number of challenges. First of all, the number of equations are usually greatly increased, since we will have to consider the concentrations of components that is present in small amounts only. Since the reaction speeds and the involved concentrations are of very different magnitudes, the chemistry term is stiff, and must be solved with implicit integration methods. Such methods require a system of equations to be solved iteratively in each time step. As we have seen, the rate equations are strongly nonlinear, which often require small time steps to be taken in order to ensure convergence. In addition, it is difficult to scale the equations properly, since the magnitudes of the concentrations are changing rapidly when a reservoir region is flushed with gaseous CO_2. Thus, the system must be frequently rescaled in order to keep it well-conditioned.

We have introduced a number of techniques to deal with these issues. To integrate the stiff equations, we utilised implicit Runge-Kutta methods, which were able to integrate the chemical terms efficiently and stably. As the error tolerances for geochemical simulations are crude, due to large uncertainties in the parameters used, lower-order methods often show the best performance. We have also shown that linear recombinations of the mass conservation equations may be necessary in order to reduce the condition number of the Jacobian matrix used for implicit calculations. When complex mineralogy is involved, such reformulations are necessary in order to acheive a numerically stable integration. In addition, frequent rescalings may also be necessary.

We have also shown that decoupling advection and reaction by operator splitting may be computationally efficient. Not only is each step computed faster than with coupled methods, the approach also allows explicit methods to be used for the advection operator, minimising numerical diffusion. Decoupled reactive steps may be computed in parallel, since the grid cells are spatially decoupled when it comes to reactions. However, as we have seen, simple splitting schemes may cause instable behaviour when complex mineralogy is involved. Specifically, splitting schemes may disturb the initial chemical equillibrium of the grid cells prior to integration of the reactive term. To avoid this situation, we have proposed a variant of iterative splitting that seems to solve the problem.

Mineral changes are observed in the reservoir approximately one year after the introduction of CO_2. As we have seen, mineral reactions may have a large impact on the brine's ion composition, but they do not necessarily influence the velocity of the migrating CO_2 plume. Within the time frame of plume migration, the porosity changes are small, and the reactions do not influence the dissolution of CO_2 significantly. Thus, it is possible that mineral reactions may be disregarded during the initial phase of CO_2 injection into a reservoir.

Finally, we have observed that the integration methods we are using, may require an excessive amount of steps to be able to integrate over discontinuities caused by phase appearances and disappearances. Further work is required to identify suitable algorithms for stopping and restarting the integration at the location of the discontinuity.

Nomenclature

Lowercase letters

a	Activity
c	(Bulk) concentration
$\hat{c}$	Interphasial concentration
e	Molar internal energy
k_F	Thermal conductivity
k^{rel}	Relative permeability
k_D	Dispersion coefficient
k	Chemical rate constant
p	Pressure
q	Scaled chemicar reaction rate
r	Chemical reaction rate
t	Time
u	Flow velocity, Darcy velocity
y	Scaled concentration
	General variable designation

Uppercase letters

D	Damköhler number
E	Internal energy
E_a	Apparent activation energy
H	Enthalpy
K	Equillibrium constant
	Rock permeability
N	Molar amounts

R	Molar production rate of a component due to chemical reactions Gas constant
S	Saturation
T	Temperature
V	Volume

Greek symbols

α	Rock compressibility
ϕ	Porosity
φ	Fugacity coefficient
γ	Activity coefficient
λ	Eigenvalue
μ	Viscosity
ψ	Phase volume fraction
ρ	Density
ξ	Extent of reaction
Ω	Reservoir region

Blackboard bold symbols

$\mathbb{0}$	Zero matrix
$\mathbb{1}$	Vector of ones
$\mathbb{A}$	Butcher tableau of Runge-Kutta methods
$\mathbb{I}$	Identity matrix
$\mathbb{J}$	Jacobian matrix
$\mathbb{M}$	Mass matrix
$\mathbb{N}$	Change-of-variables matrix
$\mathbb{S}$	Stoichiometry matrix
$\mathbb{T}$	Scaled and transposed stoichiometry matrix

Calligraphic symbols

$\mathcal{L}$	Convective term of mass conservation equation
$\mathcal{M}$	Mass conservation equation
$\mathcal{P}$	Pressure equation
$\mathcal{R}$	Reactive term of mass conservation equation
$\mathcal{T}$	Temperature equation

Bibliography

[1] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff ODE's. *SIAM Journal on Numerical Analysis*, 14(6):1006–1021, 1977.

[2] R. Alexander. Design and implementation of dirk integrators for stiff systems. *Applied numerical mathematics*, 46(1):1–17, 2003.

[3] J.W. Ball and D.K. Nordstrom. *User's manual for WATEQ4F.* US Geological Survey, 1991.

[4] B.A. Barshop, R.F. Wrenn, and C. Frieden. Analysis of numerical methods for computer simulation of kinetic processes: development of KINSIM–a flexible, portable system. *Analytical biochemistry*, 130(1):134–145, 1983.

[5] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer methods in applied mechanics and engineering*, 1(1):1–16, 1972.

[6] B. Bennion and S. Bachu. Relative permeability characteristics for supercritical CO2 displacing water in a variety of potential sequestration zones. In *SPE Annual Technical Conference and Exhibition*, 2005.

[7] C. Bethke. *Geochemical reaction modeling: Concepts and applications.* Oxford University Press, USA, 1996.

[8] R.B. Bird, W.E. Stewart, and E.N. Lightfoot. *Transport phenomena (2nd Ed.).* John Wiley & Sons, New York, NY, 2002.

[9] Z. Duan, R. Sun, C. Zhu, I. Chou, et al. An improved model for the calculation of CO2 solubility in aqueous solutions containing Na+, K+, Ca2+, Mg2+, Cl-, and SO42. *Marine Chemistry*, 98(2-4):131–139, 2006.

[10] A. Ebigbo, H. Class, and R. Helmig. CO2 leakage through an abandoned well: Problem-oriented benchmarks. *Computational Geosciences*, 11(2):103–115, 2007.

[11] J. Ennis-King and L. Paterson. Role of convective mixing in the long-term storage of carbon dioxide in deep saline formations. *SPE Journal*, 10(3):349–356, 2005.

[12] J. Ennis-King and L. Paterson. Coupling of geochemical reactions and convective mixing in the long-term geological storage of carbon dioxide. *International Journal of Greenhouse Gas Control*, 1(1):86–93, 2007.

[13] I. Faragó, B. Gnandt, and Á. Havasi. Additive and iterative operator splitting methods and their numerical investigation. *Computers & Mathematics with Applications*, 55(10):2266–2279, 2008.

[14] J.C. Friedly and J. Rubin. Solute transport with multiple equilibrium-controlled or kinetically controlled chemical reactions. *Water resources research*, 28(7):1935–1953, 1992.

[15] S.K. Garg and J.W. Pritchett. On pressure-work, viscous dissipation and the energy balance relation for geothermal reservoirs. *Advances in Water Resources*, 1(1):41–47, 1977.

[16] D.W. Green et al. *Perry's chemical engineers' handbook*. McGraw-Hill, 2008.

[17] K. Gustafsson. Control-theoretic techniques for stepsize selection in implicit Runge-Kutta methods. *ACM Transactions on Mathematical Software (TOMS)*, 20(4):496–517, 1994.

[18] E. Hairer and G. Wanner. *Solving ordinary differential equations II: Stiff and differential-algebraic problems*. Springer, 2010.

[19] B.O. Heimsund. *Mathematical and Numerical Methods for Reservoir Fluid Flow Simulation*. PhD thesis, University of Bergen, 2005.

[20] H. Hellevang. *Interactions between CO2, saline water and minerals during geological storage of CO2*. PhD thesis, The University of Bergen, 2006.

[21] H. Hellevang, P. Aagaard, E.H. Oelkers, and B. Kvamme. Can dawsonite permanently trap CO2? *Environmental science & technology*, 39(21):8281–8287, 2005.

[22] H. Hellevang, S.K. Khattri, G.E. Fladmark, and B. Kvamme. CO2 storage in the Utsira Formation: ATHENA 3-D reactive transport simulations. *Submitted to Basin Research*, 2005.

[23] E. Holzbecher. Reactive transport in porous media – concepts and numerical approaches. In D.B. Ingham and I.I. Pop, editors, *Transport phenomena in porous media.*

[24] S. Hoops, S. Sahle, R. Gauges, C. Lee, et al. COPASI - a complex pathway simulator. *Bioinformatics*, 22(24):3067, 2006.

[25] M. E. Hosea and L. F. Shampine. Analysis and implementation of TR-BDF2. *Applied Numerical Mathematics*, 20(1-2):21–37, 1996.

[26] W.H. Hundsdorfer and J.G. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations.* Springer Verlag, 2003.

[27] J.W. Johnson, J.J. Nitao, and K.G. Knauss. Reactive transport modelling of CO2 storage in saline aquifers to elucidate fundamental processes, trapping mechanisms and sequestration partitioning. *Geological Society, London, Special Publications*, 233(1):107, 2004.

[28] R.J. Kee, F.M. Rupley, E. Meeks, and J.A. Miller. CHEMKIN-III: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics. Technical report, Sandia National Laboratories, 1996.

[29] P.B. Kelemen and J. Matter. In situ carbonation of peridotite for CO2 storage. *Proceedings of the National Academy of Sciences*, 105(45):17295, 2008.

[30] C.A. Kennedy and M.H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Applied Numerical Mathematics*, 44(1-2):139–181, 2003.

[31] S.K. Khattri, G.E. Fladmark, H. Hellevang, and B. Kvamme. Simulation of long-term fate of CO2 in the sand of Utsira. *Journal of Porous Media*, 14(2), 2011.

[32] R. Korbøl and A. Kaddour. Sleipner vest CO2 disposal-injection of removed CO2 into the Utsira formation. *Energy Conversion and Management*, 36(6-9):509–512, 1995.

[33] S. Kräutle. *General multi-species reactive transport problems in porous media: Efficient numerical approaches and existence of global solutions.* PhD thesis, University of Erlangen-Nuremberg, 2008.

[34] S. Kräutle and P. Knabner. A new numerical reduction scheme for fully coupled multicomponent transport-reaction problems in porous media. *Water resources research*, 41(9):W09414, 2005.

[35] M.R. Kristensen, M.G. Gerritsen, P.G. Thomsen, M.L. Michelsen, and E.H. Stenby. Efficient integration of stiff kinetics with phase change detection for reactive reservoir processes. *Transport in porous media*, 69(3):383–409, 2007.

[36] A. Kværnø. Singly diagonally implicit Runge-Kutta methods with an explicit first stage. *BIT Numerical Mathematics*, 44(3):489–502, 2004.

[37] K. Laidler, J. Meiser, and B. Sanctuary. *Physical Chemistry.*

[38] D. Lanser and J.G. Verwer. Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling. *Journal of computational and applied mathematics*, 111(1-2):201–216, 1999.

[39] R.J. LeVeque. *Finite difference methods for ordinary and partial differential equations.* Society for Industrial and Applied Mathematics, 2007.

[40] L. Luquot and P. Gouze. Experimental determination of porosity and permeability changes induced by injection of CO2 into carbonate rocks. *Chemical Geology*, 265(1-2):148–159, 2009.

[41] L. Nghiem, P. Sammon, J. Grabenstetter, and H. Ohkuma. Modeling CO2 storage in aquifers with a fully-coupled geochemical EOS compositional simulator. In *SPE/DOE Symposium on Improved Oil Recovery*, 2004.

[42] J.M. Nordbotten, M.A. Celia, S. Bachu, and H.K. Dahle. Semianalytical solution for CO2 leakage through an abandoned well. *Environ. Sci. Technol*, 39(2):602–611, 2005.

[43] E.O.I. Obi and M.J. Blunt. Streamline-based simulation of carbon dioxide storage in a North Sea aquifer. *Water Resources Research*, 42(3):W03414, 2006.

[44] J.L. Palandri. A compilation of rate parameters of water-mineral interaction kinetics for application to geochemical modeling. Technical report, DTIC Document, 2004.

[45] T. Park and P.I. Barton. State event location in differential-algebraic models. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 6(2):137–165, 1996.

[46] D.L. Parkhurst. *User's guide to PHREEQC: A computer program for speciation, reaction-path, advective-transport, and inverse geochemical calculations.* Geological Survey, 1995.

[47] L.N Plummer, T.M.L. Wigley, and D.L. Parkhurst. The kinetics of calcite dissolution in CO2-water systems at 5 °c to 60 °c and 0.0 to 1.0 atm. *American Journal of Science*, 278, 1978.

[48] T. Poinsot and D. Veynante. *Theoretical and numerical combustion.* RT Edwards, Inc., 2005.

[49] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Mathematics of computation*, 28(125):145–162, 1974.

[50] K. Pruess. Numerical simulation of CO2 leakage from a geologic disposal reservoir including transitions from super-to sub-critical conditions, and boiling of liquid of CO2. 2003.

[51] K. Pruess, J. García, T. Kovscek, C. Oldenburg, J. Rutqvist, C. Steefel, and T. Xu. Code intercomparison builds confidence in numerical simulation models for geologic disposal of CO2. *Energy*, 29(9-10):1431–1444, 2004.

[52] K. Pruess, T. Xu, J. Apps, and J. Garcia. Numerical modeling of aquifer disposal of CO2. *SPE Journal*, 8(1):49–60, 2003.

[53] A. Riaz, M. Hesse, H.A. Tchelepi, and F.M. Orr. Onset of convection in a gravitationally unstable diffusive boundary layer in porous media. *Journal of Fluid Mechanics*, 548:87–111, 2006.

[54] M.W. Saaltink, C. Ayora, and J. Carrera. A mathematical formulation for reactive transport that eliminates mineral concentrations. *Water Resources Research*, 34(7):1649–1656, 1998.

[55] A. Sandu, J.G. Verwer, J.G. Blom, E.J. Spee, G.R. Carmichael, and F.A. Potra. Benchmarking stiff ODE solvers for atmospheric chemistry problems II: Rosenbrock solvers. *Atmospheric environment*, 31(20):3459–3472, 1997.

[56] A. Sandu, J.G. Verwer, M. Van 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(19):3151–3166, 1997.

[57] B. Sportisse. An analysis of operator splitting techniques in the stiff case. *Journal of Computational Physics*, 161(1):140–168, 2000.

[58] C.I. Steefel and A.C. Lasaga. A coupled model for transport of multiple chemical species and kinetic precipitation/dissolution reactions with applications to reactive flow in single phase hydrothermal systems. *American Journal of Science*, 294(5):529–592, 1994.

[59] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.

[60] J.G. Verwer, W.H. Hundsdorfer, and B.P. Sommeijer. Convergence properties of the Runge-Kutta-Chebyshev method. *Numerische Mathematik*, 57(1):157–178, 1990.

[61] J.W. Watts. A compositional formulation of the pressure and saturation equations. *SPE Reservoir Engineering*, 1(3):243–252, 1986.

[62] T. Wellman, R. Grigg, B. McPherson, R. Svec, and C. Peter. Evaluation of CO2-brine-reservoir rock interaction with laboratory flow tests and reactive transport modeling. In *International Symposium on Oilfield Chemistry*, 2003.

[63] T. Xu, J.A. Apps, and K. Pruess. Reactive geochemical transport simulation to study mineral trapping for CO2 disposal in deep arenaceous formations. *Journal of Geophysical Research*, 108(B2):2071, 2003.

[64] T. Xu, J.A. Apps, and K. Pruess. Numerical simulation of CO2 disposal by mineral trapping in deep aquifers. *Applied geochemistry*, 19(6):917–936, 2004.

[65] T. Xu, E. Sonnenthal, N. Spycher, and K. Pruess. TOUGHREACT–A simulation program for non-isothermal multiphase reactive geochemical transport in variably saturated geologic media: applications to geothermal injectivity and CO2 geological sequestration. *Computers & geosciences*, 32(2):145–165, 2006.

[66] X. Xu, S. Chen, and D. Zhang. Convective stability analysis of the long-term storage of carbon dioxide in deep saline aquifers. *Advances in water resources*, 29(3):397–407, 2006.

[67] G.T. Yeh and V.S. Tripathi. A model for simulating transport of reactive multispecies components: model development and demonstration. *Water Resources Research*, 27(12):3075–3094, 1991.