Comparison of Methods for Solving Coupled Reactive Transport in Porous Media

Frederik Winkel Lehn

April 30, 2018

1 Abstract

This project investigates the performance of five different approaches to solving coupled reactive transport in porous media. Two different sequential methods and three global implicit approaches. The solvers are compared in terms of numerical efficiency, namely computational time and error of solution. It is found, that on the simple equilibrium reaction that is the basis for the implementations, that the sequential methods scale superiorly with increased refinement of the grid in terms of computational time, while not having a larger error compared to the fully coupled methods. From this it is concluded that the sequential methods are preferable for simple reactions, but that the conclusion can not be extended in general, and a similar comparisons would have to be done for more complex reactions, with more non-linearity in the formulations.

2 Introduction

There exists a series of coupled reactive transport (CRT) simulation methods, e.g. the Sequential Non-Iterative Approach (SNIA), Sequential Iterative Approach (SIA) and the Global Implicit Approach (GIA). SNIA and SIA are both sequential solvers, i.e. transport is solved at a point in time while reactions are assumed constant, and then reactions are solved at the same point in time while assuming the transport is constant. The assumption of one term of the system staying constant while the other changes is not representative of a true physical system and thus leads to errors. The SIA method utilizes an iterative method to reduce this error.
The GIA solves the transport equation and reactive equations simultaneously. Due to the non-linearity of the reaction equations this requires a linearization of the equations and an iterative method, such as the Newton-Raphson method to solve the system.
In this paper, the CRT system of a simple equilibrium reaction is solved using five different approaches, SNIA, SIA and three GIA formulations. The methods are tested in terms of precision and computational efficiency. A discussion is made on the advantages and disadvantages of fully coupled methods in opposition to sequential methods, in terms of computational requirements and precision.

3 Governing Equations

3.1 Transport Equation

The transport equation reads

$$\frac{\partial(\gamma_\alpha \psi_\alpha)}{\partial t} + \nabla \cdot (\mathbf{u}\psi_\alpha - \mathcal{D}_\alpha \eta_\alpha \nabla \psi_\alpha) + \beta_\alpha \psi_\alpha = q_\alpha \tag{1}$$

reading from left to right, it is composed of a transient term, advective term, diffusive term, linear source/sink term and constant source/sink term. The subscript α represent the phase of the transported quantity. This paper will focus on coupled reactive transport in porous media

$$\phi \frac{\partial C_\alpha}{\partial t} + \nabla \cdot (\mathbf{u}C_\alpha - \mathcal{D}_\alpha \phi \nabla C_\alpha) + r_\alpha = 0 \tag{2}$$

where ϕ is the porosity of the porous medium, C_α is the concentration of component α, $\mathbf{u}$ is the Darcy velocity, $\mathcal{D}_\alpha$ is the diffusion constant and r_α is the reaction rate. The reaction rate, r_α is the change in concentration of component α due to reactions between the components and it is shown later that r_α is a non-linear function of the components in the system. $\mathcal{D}_\alpha$ is considered independent of C_α.

4 Discretization

4.1 Spatial Discretization

The deriviations shown in the following chapter are largely based on (F. W. Lehn, 2016 [1]). The transport part of (2) is discretized in space using the Finite Volume Method (FVM). The advantage of using FVM over other discretization methods such as Finite Element Method (FEM) is that FVM is mass-conserving on a local scale, which is a desirable property for modelling fluid flow.

This means (2) holds true not only for the entire reservoir, but for any arbitrarily chosen control volume in the reservoir. This allows for considering each individual cell in the domain as a separate flow equation to be solved. Hence to solve the transport equation numerically it has to be discretized. Equation (2) can be solved using the discretized cells Ω_i as a control volume, which allows for calculating the flow through each cell by integrating over the volume of the cell

$$\int_{\Omega_i} \phi \frac{\mathrm{d}}{\mathrm{d}t} C_\alpha \, \mathrm{d}V + \int_{\Omega_i} \nabla \cdot (\mathbf{u}C_\alpha) \, \mathrm{d}V - \int_{\Omega_i} \nabla \cdot (\mathcal{D}_\alpha \phi \nabla C_\alpha) \, \mathrm{d}V = 0 \tag{3}$$

rearranging and using Gauss' Divergence Theorem the above can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_i} \phi C_\alpha \, \mathrm{d}V + \int_{\partial\Omega_i} (\mathbf{u}C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A - \int_{\partial\Omega_i} (\mathcal{D}_\alpha \phi \nabla C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A = 0 \tag{4}$$

where A is the area of a the cell faces and $\mathbf{n}$ is the outwards unit normal. Under the assumption that porosity is homogeneous within a cell, the integral over the temporal term can be approximated as

$$\int_{\Omega_i} \phi C_\alpha \, \mathrm{d}V \approx \phi C_{\alpha,i} V_i, \quad V_i = |\Omega_i| \tag{5}$$

Let $\mathcal{N}^{(i)}$ denote the set of indices of all neighbouring cells of cell i, and let $\partial\Omega_{ij} = \Omega_i \cap \Omega_j$ such that $\partial\Omega_{ij}$ is the face shared by cell i and j.

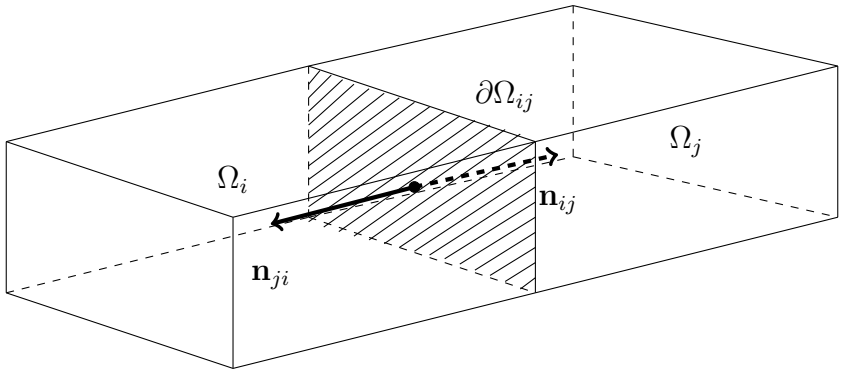


Figure 1: Two neighboring cells with outwards unit normals from their shared cell-face, $\partial\Omega_{ij}$

Fig. 1 shows cell i and one of its neighbours j. To relate this to the set of indices $\mathcal{N}^{(i)}$ the following will be true $j \in \mathcal{N}^{(i)}$ and $i \in \mathcal{N}^{(j)}$. Similarly $\partial\Omega_{ij}$ is the shaded surface and this is the cell-face that cell i and j share. Further, because the flux between cells is calculated at the cell interfaces, the vector s_{ij} is introduced

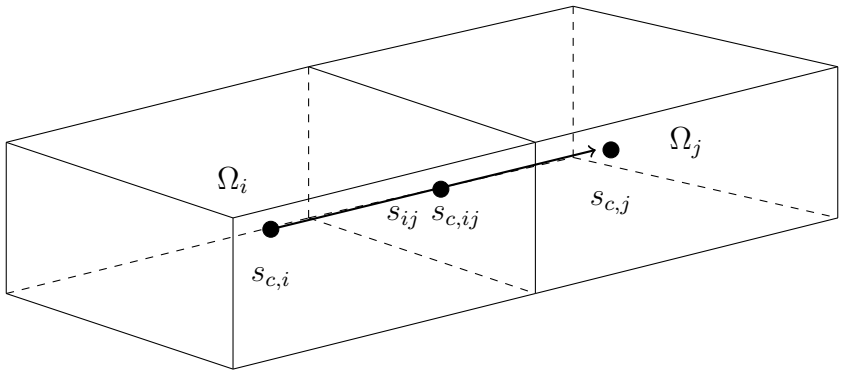


Figure 2: Vector between spatial centroids

where $s_{c,i}$ is the geometric cell centroid of Ω_i, $s_{c,j}$ similarly for Ω_j and $s_{c,ij}$ is the geometric face centroid between cells i and j. The vector joining the two centroids, s_{ij}, is then defined as

$$s_{ij} = s_{c,j} - s_{c,i} \tag{6}$$

and its length, Δs_{ij}, is defined as

$$\Delta s_{ij} = |s_{ij}| \tag{7}$$

similarly the distances from the cell centres to the shared cell face are defined as

$$\begin{aligned} \Delta s_i &= |s_{c,ij} - s_{c,i}| \\ \Delta s_j &= |s_{c,ij} - s_{c,j}| \end{aligned} \tag{8}$$

this vector will be used for the discretization of the advective and the diffusive terms.

The integral of the spatial terms over the boundary may be split up as sums over the neighbouring cell faces. This is shown for the advective and diffusive terms respectively

4.1.1 Advection

For advection the sum over the neighbouring cell faces become

$$\int_{\partial\Omega_i} (\mathbf{u}C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A = \sum_{j \in \mathcal{N}^{(i)}} \int_{\partial\Omega_{ij}} (\mathbf{u}C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A \tag{9}$$

assuming that the velocity is homogeneous over the entire cell-face, the integral can be approximated as

$$\int_{\partial\Omega_{ij}} (\mathbf{u}C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A \approx (\mathbf{u}_{ij}C_{\alpha,ij}) \cdot \mathbf{n}_{ij} A_{ij} = (\mathbf{u}_{ij} \cdot \mathbf{n} A_{ij}) C_{\alpha,ij} \tag{10}$$

where $\mathbf{n}_{ij}$ is the unit-normal outwards from the face $\partial\Omega_{ij}$ and $A_{ij} = |\partial\Omega_{ij}|$. $C_{\alpha,ij}$ is the concentration at the interface of the cells, cell i and j, which is found using a linear interpolation, hence

$$(\mathbf{u}_{ij} \cdot \mathbf{n} A_{ij}) C_{\alpha,ij} \approx (\mathbf{u}_{ij} \cdot \mathbf{n} A_{ij}) \frac{C_{\alpha,i}\Delta s_i + C_{\alpha,j}\Delta s_j}{\Delta s_{ij}} \tag{11}$$

the dot-product $\mathbf{u}_{ij} \cdot \mathbf{n}_{ij}$ results in the value of the velocity at the cell interface, and a sign dependent on whether it is flowing in or out of the cell (positive for out and negative for in). Writing this in terms of linear algebra becomes a system matrix with a diagonal and $2n_{dims}$ bidiagonals.

$$\mathbf{M}_{adv}\mathbf{C} = \mathbf{0} \tag{12}$$

where

$$\begin{aligned} \mathbf{M}_{adv,ij} &= \begin{cases} (\mathbf{u}_{ij} \cdot \mathbf{n}_{ij}) \frac{A_{ij}\Delta s_i}{\Delta s_{ij}}, & \Delta C_{ij} > 0 \\ (\mathbf{u}_{ji} \cdot \mathbf{n}_{ji}) \frac{A_{ji}\Delta s_j}{\Delta s_{ij}}, & \Delta C_{ij} < 0 \end{cases}, \quad j \in \mathcal{N}^{(i)} \\ \mathbf{M}_{adv,ii} &= \sum_{j \in \mathcal{N}^{(i)}} \mathbf{M}_{adv,ij} \end{aligned} \tag{13}$$

i.e. positive in the upwind direction and negative in the downwind direction (ΔC_{ij} is introduced in the diffusion section), also subscript ij refers to the cell interface in the upwind direction and ji in the downwind direction.

4.1.2 Diffusion

The discretization of the diffusion is very similar to the advection for the first part

$$\int_{\partial\Omega_i} (\mathcal{D}_\alpha \phi \nabla C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A = \sum_{j \in \mathcal{N}^{(i)}} \int_{\partial\Omega_{ij}} (\mathcal{D}_\alpha \phi \nabla C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A \tag{14}$$

assuming that the velocity is homogeneous over the entire cell-face, the integral can be approximated as

$$\int_{\partial\Omega_{ij}} (\mathcal{D}_\alpha \phi \nabla C_\alpha) \cdot \mathbf{n} \, \mathrm{d}A \approx (\mathcal{D}_{\alpha,i} \phi \nabla C_{\alpha,i}) \cdot \mathbf{n}_{ij} A_{ij} \tag{15}$$

however for the diffusion, (15) has to be discretized further, as it is a function of the gradient of $C_{\alpha,i}$. ∇C can be approximated by the change in concentration between cell i and j, ΔC_{ij}, and the distance between the cells

$$\nabla C_{ij} \approx \frac{s_{ij}}{\Delta s_{ij}} \frac{1}{\Delta s_{ij}} \Delta C_{ij}, \quad \Delta C_{ij} = C_j - C_i \tag{16}$$

where Δs_{ij} is used to normalize the distance in both cases. Looking at Fig. 1 & 2 it can be seen that s_{ij} is parallel to $\mathbf{n}_{ij}$ and since the normal vector, $\mathbf{n}_{ij}$, has unit length, this results in

$$\frac{s_{ij}}{\Delta s_{ij}} = \mathbf{n}_{ij} \tag{17}$$

the dot-product of (15) can now be written as

$$(\mathcal{D}_{\alpha,i} \phi \nabla C_{\alpha,i}) \cdot \mathbf{n}_{ij} A_{ij} \approx \left(\mathcal{D}_{\alpha,ij} \phi \frac{s_{ij}}{\Delta s_{ij}} \frac{\Delta C_{ij}}{\Delta s_{ij}} \right) \cdot \mathbf{n}_{ij} A_{ij} \tag{18}$$

but since

$$\frac{s_{ij}}{\Delta s_{ij}} \cdot \mathbf{n}_{ij} = \mathbf{n}_{ij} \cdot \mathbf{n}_{ij} = 1 \tag{19}$$

(18) reduces to

$$(\mathcal{D}_{\alpha,i} \phi \nabla C_{\alpha,i}) \cdot \mathbf{n}_{ij} A_{ij} \approx \mathcal{D}_{\alpha,ij} \phi \frac{A_{ij}}{\Delta s_{ij}} \Delta C_{ij} \tag{20}$$

Writing this in terms of linear algebra becomes

$$\mathbf{M}_{diff} \mathbf{C} = \mathbf{0} \tag{21}$$

where

$$\begin{aligned} \mathbf{M}_{diff,ij} &= \begin{cases} \mathcal{D}_{\alpha,ij} \frac{A_{ij}}{\Delta s}, & \Delta C_{ij} > 0 \\ \mathcal{D}_{\alpha,ji} \frac{A_{ji}}{\Delta s}, & \Delta C_{ij} < 0 \end{cases}, \quad j \in \mathcal{N}^{(i)} \\ \mathbf{M}_{diff,ii} &= - \sum_{j \in \mathcal{N}^{(i)}} \mathbf{M}_{diff,ij} \end{aligned} \tag{22}$$

4.1.3 Boundary Conditions

4.2 Temporal Discretization

Some equilibrium reactions occur almost instantaneously. For this reason a high degree of numerical stability is required even at small time steps. In order to make solutions computationally feasible a implicit time-stepping scheme is used for the transport part of the equation, and also for the reaction part in the GIA cases.

4.2.1 Forward Differencing

The time-derivative is approximated using a forward differencing

$$\gamma \frac{\partial C}{\partial t} \approx \gamma \frac{C_{t+1} - C_t}{\Delta t} = \frac{\gamma}{\Delta t}(C_{t+1} - C_t), \quad \Delta t = t_{i+1} - t_i \tag{23}$$

at a given time-step, C_t is known, and C_{t+1} is the unknown variable, hence writing this in terms of linear algebra yields

$$\mathbf{M}_{t,\gamma}\mathbf{C} = \mathbf{b} \Leftrightarrow \begin{bmatrix} 0 & & & \\ & \frac{\gamma}{\Delta t} & & \\ & & \ddots & \\ & & & \frac{\gamma}{\Delta t} \\ & & & 0 \end{bmatrix} \begin{bmatrix} C_{t+1,1} \\ C_{t+1,2} \\ \vdots \\ C_{t+1,n+1} \\ C_{t+1,n+2} \end{bmatrix} = \frac{\gamma}{\Delta t} \begin{bmatrix} C_{t,1} \\ C_{t,2} \\ \vdots \\ C_{t,n+1} \\ C_{t,n+2} \end{bmatrix} \tag{24}$$

4.2.2 Backwards Euler Method

(2) can be written on the general ODE form

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(t, y) \tag{25}$$

integrating that from $[t_n, t_{n+1}]$ where $t_{n+1} = t_n + \Delta t$ yields

$$y_{n+1} - y_n = \int_{t_n}^{t_{n+1}} f(t, y)\ \mathrm{d}t \approx \Delta t\ f(t_{n+1}, y_{n+1}) \tag{26}$$

where $y_n = y(t_n)$. This can be rearranged as

$$y_{n+1} - \Delta t\ f(t_{n+1}, y_{n+1}) = y_n \tag{27}$$

as can be seen, the Backwards Euler method is an implicit method, requiring an iterative approach to solve. Using linearized formulations of the reaction rates (shown later), (2) becomes a linear function, which allows for solving it as linear system of equations. GIA applies the Newton-Raphson method to solve the linearized system of equations iteratively. The solution is updated

in each iteration by an approximated step

$$\begin{aligned} \mathbf{MC} - \frac{\partial \mathbf{M}}{\partial \mathbf{C}}\mathbf{C} &= \mathbf{b} \\ \mathbf{MC} - \mathbf{JC} &= \mathbf{b} \\ (\mathbf{M} - \mathbf{J})\mathbf{C} &= \mathbf{b} \end{aligned} \tag{28}$$

where $\mathbf{J}$ is the Jacobian matrix. The matrices $\mathbf{M}$ and $\mathbf{J}$ are both block-matrices, with $\mathbf{M}$ being a diagonal block-matrix, of which the respective components are written

$$\mathbf{M}_\alpha = \mathbf{M}_{t_\alpha,\phi} + \mathbf{M}_{adv} - \mathbf{M}_{diff_\alpha} + \mathbf{M}_{bc_\alpha}, \quad \alpha \in \{A, B, C\} \tag{29}$$

Written for the specific problem this becomes

$$\begin{bmatrix} \mathbf{M}_A - \nu_A \mathbf{J}_{A_0} & -\nu_A \mathbf{J}_{B_0} & -\nu_A \mathbf{J}_{C_0} \\ -\nu_B \mathbf{J}_{A_0} & \mathbf{M}_B - \nu_B \mathbf{J}_{B_0} & -\nu_B \mathbf{J}_{C_0} \\ \nu_C \mathbf{J}_{A_0} & \nu_C \mathbf{J}_{B_0} & \mathbf{M}_C + \nu_C \mathbf{J}_{C_0} \end{bmatrix} \begin{bmatrix} \mathbf{C}_A \\ \mathbf{C}_B \\ \mathbf{C}_C \end{bmatrix} = \begin{bmatrix} \mathbf{b}_A \\ \mathbf{b}_B \\ \mathbf{b}_C \end{bmatrix} \tag{30}$$

where ν_α are stoichiometric coefficients. The Jacobian terms, $\mathbf{J}_{\alpha_0}$ and the RHS, $\mathbf{b}_\alpha$, are specific to each individual GIA, and will be explained in detail in their respective sections.

5 Reaction Equations

As stated in the introduction, there exists multiple different ways of simulating CRT. In this section five different approaches will be introduced, and later tested. This is the SNIA, SIA, GIA using rate of conversion notation and using two different kinematic reaction rate formulations. In all cases the following simple equilibrium reaction is considered

$$\nu_A \mathrm{A} + \nu_B \mathrm{B} \rightleftharpoons \nu_C \mathrm{C} \tag{31}$$

5.1 SNIA

The mass balance for the concentrations of (31) may be written as

$$\begin{aligned} C_A &= C_{A_0} - \nu_A \epsilon \\ C_B &= C_{B_0} - \nu_B \epsilon \\ C_C &= C_{C_0} + \nu_C \epsilon \end{aligned} \tag{32}$$

or more compactly

$$C_\alpha = C_{\alpha_0} + \nu_\alpha \epsilon, \quad \alpha \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}\} \tag{33}$$

where ϵ is the change in concentration. Notice here that the stoichiometric coefficients, ν_i, are negative for reactants and positive for products. The equilibrium equation becomes

$$K_{eq} = \frac{C_C}{C_A \cdot C_B} = \frac{C_{C_0} + \nu_C \epsilon}{(C_{A_0} - \nu_A \epsilon) \cdot (C_{B_0} - \nu_B \epsilon)} \tag{34}$$

Solving (34) for ϵ results in two expressions, where of only one has a physical meaning, hence the expression for ϵ becomes

$$\epsilon = \frac{1}{2\nu_A\nu_B K_{eq}}\Bigg(\nu_A C_B K_{eq} + \nu_B C_A K_{eq} + \nu_C - \sqrt{\nu_A^2 C_B^2 K_{eq}^2 - 2\nu_A\nu_B C_A C_B K_{eq}^2 + \nu_B^2 C_A^2 K_{eq}^2 + 4\nu_A\nu_B C_C K_{eq} + 2\nu_A\nu_C C_B K_{eq} + 2\nu_B\nu_C C_A K_{eq} + \nu_C^2}\Bigg) \tag{35}$$

Using (35), (2) is solved sequentially, by first solving the transport equation for $C_{\alpha,trans}$, then solving the reactive equation

$$C_\alpha = C_{\alpha,trans} + \nu_\alpha \epsilon(C_{A,trans}, C_{B,trans}, C_{C,trans}) \tag{36}$$

this procedure is carried out for each time-step.

5.2 SIA

The upside of SNIA, is that it is simple and fast. The downside is that by solving sequentially, the couplings between the transport equation and the reaction equation are ignored, which may introduce errors. A way to reduce these error is by introducing an iterative approach. SIA, like SNIA solves the transport equation and reaction equations sequentially in a decoupled manner, but SIA applies an inner loop in the time-stepping, where the solution is recomputed a number of times, until a convergence criteria is met

$$||\mathbf{C}_{t+1}^i - \mathbf{C}_{t+1}^{i-1}||_1 < \text{tol} \tag{37}$$

where tol is some user-defined tolerance, subscript t is time-step and superscript i is the inner loop iteration.

5.3 GIA

5.3.1 Rate of Conversion

Similar to the SNIA, where the transport equation is solved first, and then the reactions solved secondly using the rate of conversion, ϵ, the coupled reactive transport may also be solved fully coupled using ϵ. This means ϵ has to be linearized

$$r = \nu_\alpha \epsilon \approx \nu_\alpha \left[\epsilon_0 + \epsilon'_{A_0}(C_A - C_{A_0}) + \epsilon'_{B_0}(C_B - C_{B_0}) + \epsilon'_{C_0}(C_C - C_{C_0})\right] \tag{38}$$

where the terms ϵ'_{A_0}, ϵ'_{B_0} and ϵ'_{C_0} can be found in appendix A. Writing this in discretized form the RHS, $\mathbf{b}_\alpha$, is

$$\mathbf{b}_\alpha = \mathbf{b}_{t_\alpha,\phi} + \mathbf{b}_{bc_\alpha} + \nu_\alpha \left[-\epsilon_0 + \epsilon'_{A_0} C_{A_0} + \epsilon'_{B_0} C_{B_0} + \epsilon'_{C_0} C_{C_0}\right], \quad \alpha \in \{A, B, C\} \tag{39}$$

and the Jacobian's are given by

$$\begin{aligned} \mathbf{J}_{A_0} &= \epsilon'_{A_0} \\ \mathbf{J}_{B_0} &= \epsilon'_{B_0} \\ \mathbf{J}_{C_0} &= \epsilon'_{C_0} \end{aligned} \tag{40}$$

(30) is solved until convergence is reached, defined by the convergence criteria

$$||\mathbf{C}_{t+1}^{i} - \mathbf{C}_{t+1}^{i-1}||_1 < \text{tol} \tag{41}$$

where tol is some user-defined tolerance, subscript t is time-step and superscript i is the Newton-iteration.

5.3.2 Kinetic Reaction Rate (K_{eq})

With this approach, the reaction rate, r_α in (2) may be written as

$$r_\alpha = \nu_\alpha \left(K_{eq} C_A^a C_B^b - C_C^c \right) \tag{42}$$

where a, b and c are experimentally determined exponents and

$$\text{order of reaction} = a + b + c \tag{43}$$

Both terms of (42) are linearized using Taylor expansion

$$K_{eq} C_A^c C_B^b \approx K_{eq} C_{A_0}^a C_{B_0}^b + a K_{eq} C_{A_0}^{a-1} C_{B_0}^b (C_A - C_{A_0}) + b K_{eq} C_{A_0}^a C_{B_0}^{b-1} (C_B - C_{B_0}) \tag{44}$$

and

$$C_C^c \approx C_{C_0}^c + c C_{C_0}^{c-1} (C_C - C_{C_0}) \tag{45}$$

Writing this in discretized form results in the RHS, $\mathbf{b}_\alpha$, being

$$\begin{aligned} \mathbf{b}_\alpha &= \mathbf{b}_{t_\alpha,\phi} + \mathbf{b}_{bc_\alpha} - K_{eq} \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b + a K_{eq} \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b + b K_{eq} \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b + \mathbf{C}_{C_0}^c - c \mathbf{C}_{C_0}^c \\ &= \mathbf{b}_{t_\alpha,\phi} + \mathbf{b}_{bc_\alpha} + \nu_\alpha \left[-K_{eq} \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b (1 - a - b) + \mathbf{C}_{C_0}^c (1 - c) \right], \quad \alpha \in \{A, B, C\} \end{aligned} \tag{46}$$

and the Jacobian's given by

$$\begin{aligned} \mathbf{J}_{A_0} &= a K_{eq} \mathbf{C}_{A_0}^{a-1} \mathbf{C}_{B_0}^b \\ \mathbf{J}_{B_0} &= b K_{eq} \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^{b-1} \\ \mathbf{J}_{C_0} &= -c \mathbf{C}_{C_0}^{c-1} \end{aligned} \tag{47}$$

(30) is solved until convergence is reached, defined by the convergence criteria

$$||\mathbf{C}_{t+1}^{i} - \mathbf{C}_{t+1}^{i-1}||_1 < \text{tol} \tag{48}$$

where tol is some user-defined tolerance, subscript t is time-step and superscript i is the Newton-iteration.

5.3.3 Kinetic Reaction Rate (k_f & k_b)

With this approach, the reaction rate, r_α in (2) may be written as

$$r_\alpha = \nu_\alpha \left(k_f C_A^a C_B^b - k_b C_C^c \right) \tag{49}$$

where a, b and c are experimentally determined exponents and

$$\text{order of reaction} = a + b + c \tag{50}$$

Notice here that only k_f is unknown, and k_b is explicitly defined by

$$k_b = \frac{k_f}{K_{eq}} \tag{51}$$

Both terms of (42) are linearized using Taylor expansion

$$k_f C_A^c C_B^b \approx k_f C_{A_0}^a C_{B_0}^b + a k_f C_{A_0}^{a-1} C_{B_0}^b (C_A - C_{A_0}) + b k_f C_{A_0}^a C_{B_0}^{b-1} (C_B - C_{B_0}) \tag{52}$$

and

$$k_b C_C^c \approx k_b C_{C_0}^c + c k_b C_{C_0}^{c-1} (C_C - C_{C_0}) \tag{53}$$

the RHS, $\mathbf{b}_\alpha$, is

$$\begin{aligned} \mathbf{b}_\alpha &= \mathbf{b}_{t_\alpha,\phi} + \mathbf{b}_{bc_\alpha} - k_f \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b + a k_f \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b + b k_f \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b + k_b \mathbf{C}_{C_0}^c - c k_b \mathbf{C}_{C_0}^c \\ &= \mathbf{b}_{t_\alpha,\phi} + \mathbf{b}_{bc_\alpha} + \nu_\alpha \left[-k_f \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^b (1 - a - b) + k_b \mathbf{C}_{C_0}^c (1 - c) \right], \quad \alpha \in \{A, B, C\} \end{aligned} \tag{54}$$

and the Jacobian's are given by

$$\begin{aligned} \mathbf{J}_{A_0} &= a k_f \mathbf{C}_{A_0}^{a-1} \mathbf{C}_{B_0}^b \\ \mathbf{J}_{B_0} &= b k_f \mathbf{C}_{A_0}^a \mathbf{C}_{B_0}^{b-1} \\ \mathbf{J}_{C_0} &= -c k_b \mathbf{C}_{C_0}^{c-1} \end{aligned} \tag{55}$$

(30) is solved until convergence is reached, defined by the convergence criteria

$$||\mathbf{C}_{t+1}^i - \mathbf{C}_{t+1}^{i-1}||_1 < \text{tol} \tag{56}$$

where tol is some user-defined tolerance, subscript t is time-step and superscript i is the Newton-iteration.

It should be noted that this method is sensitive to low values of k_f and k_b, so k_f should be selected to be sufficiently high.

6 Results

When testing numerical methods, two main factors are important, error and time. In this case the normal approach to testing error, i.e. against an analytical solution, is not a possibility, as no analytical solution exists for the problem. Hence an alternative approach has to be used. The second parameter is time, i.e. the computational effort required to run the simulation.

Usually it is the case that the simpler approaches (SNIA and SIA) will be less precise, but faster than the complex approach (GIA), and eventually the method of choice is based on a trade-off

between precision and time.

The implementation used to solve the (2) using the five different approaches, is based on a FVM toolbox for Julia (A. A. Eftakhari [2]).

A one dimensional problem is set up and run

Table 1: Parameters of 1D problem

Variable	Symbol	General			Unit
Grid length	L_x	1			m
Porosity	ϕ	0.4			-
Darcy velocity	u	$1.1574 \cdot 10^{-5}$			m/s
Equilibrium constant	K_{eq}	10			-
Forward reaction rate	k_f	100			-
Backward reaction rate	k_b	10			-
Tolerance	-	10^{-10}			-
Variable	Symbol	Component A	Component B	Component C	Unit
Stochiometric constant	ν	1	1	1	-
Reaction order	a, b, c	1	1	1	-
Diffusion constant	$\mathcal{D}$	10^{-9}	10^{-9}	10^{-9}	m^2/s
Boundary condition	-	1	0.5	0	mol/m^3
Initial condition	-	0	0	0	mol/m^3

Three tests of time and error are carried out

1. Computational time with increasing grid size
2. Relative error as a function of increasing time-step size
3. Relative error as a function of increased grid refinement

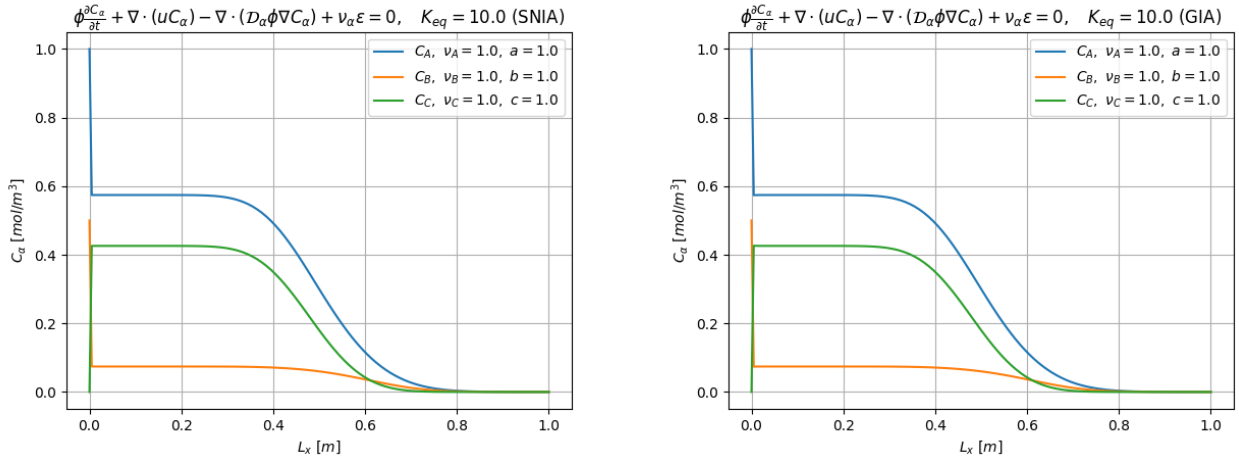
Initially the solutions of two different methods are shown (SNIA and GIA (ϵ)), in order to verify that the implementation gives the expected results. The time-step size is

$$\Delta t = \frac{L_x/(u/\phi)}{N_x} \tag{57}$$

and the simulation time

$$T = \Delta t \frac{N_x}{2} \tag{58}$$

ensuring that the flow moves at most one grid cell per time-step, and that the simulation is terminated when the flow has passed halfway through the porous medium.



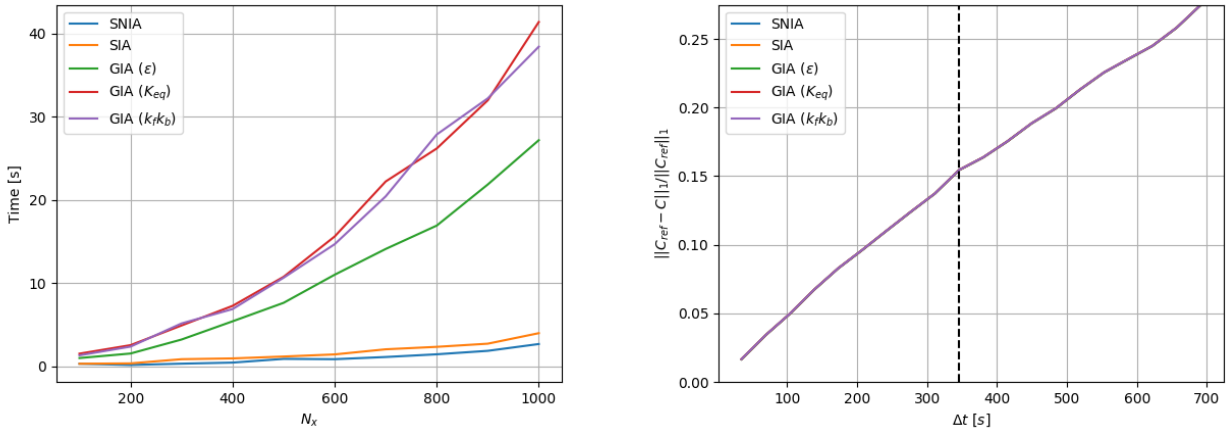
(a) (b)

Figure 3: **(a)**: Solution to (2), using the SNIA formulation. **(b)**: Solution to (2), using the GIA (ϵ) formulation.

The initial jumps seen in the concentrations for both Fig. 3a & 3b is due to the equilibrium being reaches almost instantly after the reactants are injected into the porous medium.

In order to test the five approaches' error without an analytical solution, a reference solution is produced instead. In all cases the GIA (ϵ formulation) is used as reference. For the test of error as a function of time-step, this solution is found using a very low time-step size. For the test of error as a function of grid refinement, the reference case is solved with a very high number of grid cells. This approach is used to approximate an analytical solution.

The time related behaviour of the solutions are seen on the next plots. Where the reference time-step is chosen to be a tenth of the minimum tested time-step.

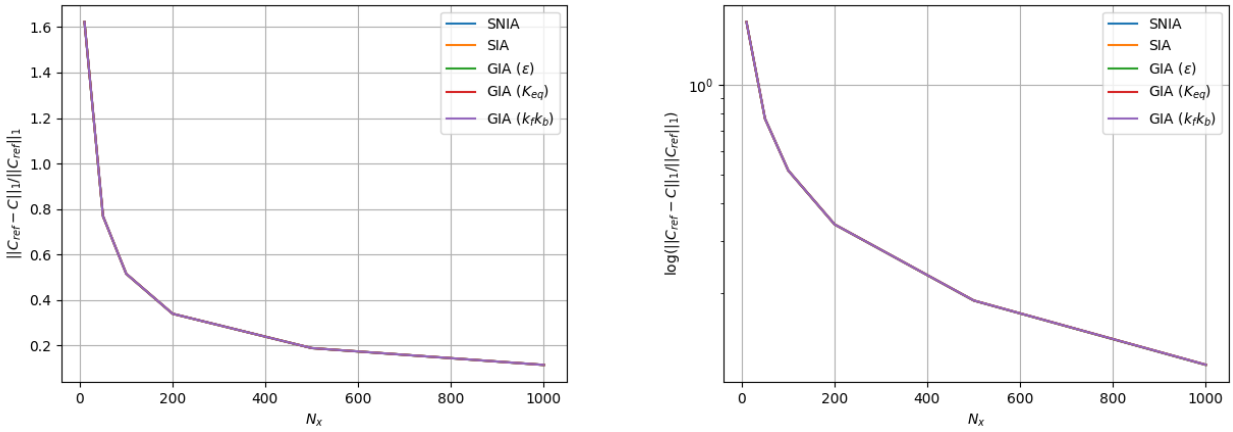


(a) (b)

Figure 4: **(a)**: Computational effort of the five approaches with increasing grid size. **(b)**: Error of solutions with increasing time-step size (relative to reference solution)

Fig. 4a shows the expected behaviour. Namely that the sequential methods scale substantially better than the global implicit approaches, and that the K_{eq} and $k_f k_b$ formulation are practically the same. This is to be expected as they are based on the same methodology, and only a minor mathematical rearranging in the formulation. The key take away here, is that the GIA (ϵ) approach scale better than the two other GIA approaches, which is not an obvious result. Fig. 4b also exhibits the expected behaviour, namely that the relative error increases when Δt increases. The dotted line shows the time-step at which the flow may move more than one grid cell per time-step.

For the test of the number of grid cells effect on the relative error, a reference solution is created, to mimick the analytical solution. The grid is tested using $N_x \in [11,\ 51,\ 101,\ 201,\ 501,\ 1001]$ and a reference grid size of $N_{x,ref} = 10001$.



(a) Relative error.

(b) Relative error (log y-axis).

Figure 5: Relative error of the five approaches relative to a high resolution reference solution.

It is slightly difficult to make assumptions about the behaviour of the error as a function of number of grid cells based on Fig. 5a & 5b, as it would require the analytical solution to observe linear, quadratic or higher order behaviour of convergence. However, it is evident, that error is reduced when the number of grid cells is increased, as it should.

7 Discussion

In general the code exhibits the expected behaviour, as previously expected. However one major result is a surprise, namely that all five approaches have almost the exact similar error, as seen on Fig. 4b, 5a & 5b (cannot see the differences on the plots). It was expected that the GIA approaches would have a lower error than the sequential ones. An explanation for this may be, that the very simple equilibrium reaction considered here, does not introduce enough non-linearity into the system, to the point where the sequential methods cannot cope.

8 Conclusion

Based on the results and the discussion, especially Fig. 4a, 4b, 5a & 5b, it is evident that the sequential methods scale much better with increasing number of grid cells, while it has no effect on the error of the solution. For this reason it can be concluded, that for simple reactions, sequential methods, and SNIA in particular is the optimal solver. It is important to emphasize though, that this may not be the case when additional complexity is added to the system, where more non-linearity may introduce complications for the sequential methods.

9 Appendix

9.1 Appendix A

With regards to C_A

$$\epsilon'_{A_0} = \frac{\partial \epsilon}{\partial C_A} = \frac{1}{2\nu_A\nu_B K_{eq}}\Bigg(\nu_B K_{eq} - \frac{C_A K_{eq}^2\nu_B^2 - C_B K_{eq}^2\nu_A\nu_B + K_{eq}\nu_B\nu_C}{\sqrt{\nu_A^2 C_B^2 K_{eq}^2 - 2\nu_A\nu_B C_A C_B K_{eq}^2 + \nu_B^2 C_A^2 K_{eq}^2 + 4\nu_A\nu_B C_C K_{eq} + 2\nu_A\nu_C C_B K_{eq} + 2\nu_B\nu_C C_A K_{eq} + \nu_C^2}}\Bigg) \tag{59}$$

With regards to C_B

$$\epsilon'_{B_0} = \frac{\partial \epsilon}{\partial C_B} = \frac{1}{2\nu_A\nu_B K_{eq}}\Bigg(\nu_A K_{eq} - \frac{C_B K_{eq}^2\nu_A^2 - C_A K_{eq}^2\nu_A\nu_B + K_{eq}\nu_A\nu_C}{\sqrt{\nu_A^2 C_B^2 K_{eq}^2 - 2\nu_A\nu_B C_A C_B K_{eq}^2 + \nu_B^2 C_A^2 K_{eq}^2 + 4\nu_A\nu_B C_C K_{eq} + 2\nu_A\nu_C C_B K_{eq} + 2\nu_B\nu_C C_A K_{eq} + \nu_C^2}}\Bigg) \tag{60}$$

With regards to C_C

$$\epsilon'_{C_0} = \frac{\partial \epsilon}{\partial C_C} = - \frac{1}{\sqrt{\nu_A^2 C_B^2 K_{eq}^2 - 2\nu_A\nu_B C_A C_B K_{eq}^2 + \nu_B^2 C_A^2 K_{eq}^2 + 4\nu_A\nu_B C_C K_{eq} + 2\nu_A\nu_C C_B K_{eq} + 2\nu_B\nu_C C_A K_{eq} + \nu_C^2}} \tag{61}$$

References

[1] F. W. Lehn (2016). *Multiscale Modelling for Efficient Simulation of Reservoirs*. B. Sc. in Mathematics and Technology, DTU

[2] A. A. Eftekhari. *Finite Volume Method Toolbox for Julia*. Found at `https://github.com/simulkade/JFVM.jl`.