Reaction-Diffusion-Convection

Contents

[1 Introduction 1](#_Toc499114263)

[2 Weak Formulation 1](#_Toc499114264)

[3 Discretization 2](#_Toc499114265)

[4 Time Integration 3](#_Toc499114266)

[4.1 Backward Euler 3](#_Toc499114267)

[4.2 Generalized trapezoidal rule 3](#_Toc499114268)

[4.3 Chemical reactions 4](#_Toc499114269)

[5 Mixtures 5](#_Toc499114270)

[5.1 Solid-bound molecules 6](#_Toc499114271)

[6 Analytical Solution 7](#_Toc499114272)

[7 Prescribed concentrations in NL analyses 8](#_Toc499114273)

[8 Analytical solutions for reaction equations 9](#_Toc499114274)

[8.1 Example 1 9](#_Toc499114275)

[8.2 Example 2 9](#_Toc499114276)

# Introduction

Consider *N* chemical species that undergo diffusion and can react with each other, and are inside a flow that moves with velocity **v**. The governing equation takes on the following form.



Here,  represents the concentration of chemical species *i,*  the diffusion constant of species *i*, and  accounts for all local reactions that contribute to the concentration of species *i*. The vector  represents all the concentrations.

# Weak Formulation

First, an integral version of this equation must be established. Let’s reorganize the terms as follows,

.

Then, consider *w* a trial function and integrate over the domain.



This can be rewritten as,



Here, , is the flux vector and the second term is integrated over the surface for which **q** is prescribed.

# Discretization

Next, the finite element approximations are introduced.



This results in the following,



This can be written as a matrix equation,



Where,



Here, degrees of freedom are denoted by *p*, where *p* is the degree of freedom *i* of node *a*. Similarly, *q* refers to the degree of freedom *j* of node *b*.

Equation is the semi-discrete equation. Next, a time integration method needs to be introduced. Using the generalized trapezoidal rule this can be written as



which is a nonlinear system of equations.

# Time Integration

## Backward Euler

For , we recover the backward Euler method.



We solve it using Newton’s method. First, we define the residual.



where . A Taylor expansion around the current guess gives,



Here,



This can be solved for .



Then, the solution can be updated,

.

## Generalized trapezoidal rule

Considering the general case,



The residual in this case can be written as,



It follows that the stiffness matrix is given by,



## Chemical reactions

Thus far, **F** remained unspecified, or better *R*. We look now at the specific form this function takes when considering chemical reactions.

For a reaction of the type



The reaction rate is given by



The rate is proportional to the concentrations of the reactants.



The factor *k* is the rate constant and only depends on temperature (Arrhenius rule). In single-step reactions, the exponents are taken to be the stoichiometric coefficient.

A general chemical reaction is specified as,



Here,  denotes the chemical species.

For chemical reactions, **F** takes on the following form.



Here, the sum is over *J* reactions and  and,



the reaction rate of reaction *j*. Usually, the exponent is taken to be , as according to the law of mass action. (see “rate equation” on wiki)

Let’s evaluate the stiffness matrix for **F**.



Evaluating the derivative of *R* with respect to degree of freedom *q*, we find,

,

where



Thus, the element stiffness matrix is given by,



# Mixtures

Thus far, we considered the case where the solute is free, i.e. not inside a porous material. In the case if the solutes are contained inside a porous material, then the equations above remain valid, except now that the concentrations are with respect to the mixture volume. The ‘true’ concentration  , i.e. the concentration w.r.t. to the solvent is related to the ‘bulk’ concentration via,



where  is the *fluid volume fraction*, i.e. the fraction of the volume that is accessible to the fluid (and therefore to the solvents).

In the absence of chemical reactions and convection, equation becomes,



Note that can be dependent on the position, so in general does not drop out of the equation.

Then how can I bring it in front of the first gradient of c? I think this is the correct formula in analogy with the multiphasic code, but something’s fishy here.

Notice that now **D** is the diffusivity in the porous mixture and can be different (i.e. lower) than the *free* diffusivity.

Performing the usual steps,





where . Introducing discretization,



## Solid-bound molecules

In the above we assumed that the domain was composed of a mixture, defined by two phases, a solid and a fluid phase. The fluid was assumed to be a solvent in which several species were dissolved. We emphasize again that the volume fractions of the dissolved solutions are negligible and thus that the solid and fluid volume fractions add up to unity.



We now consider the case that a chemical species exists as part of the solid phase. Such a species is called a *solid-bound molecule*. Since it is bound to the solid (immovable) phase, it is assumed that a solid-bound molecule does not undergo diffusion. Let us denote the apparent density of the sbm by . It then follows that the governing equation for a solid-bound molecule is given by,



where  is the mass supply for the sbm. In this context, the mass supply will be calculated from the chemical reactions.

Integrating equation is done via the backward Euler method.



Alternatively, the trapezoidal rule could be used,



A change in the apparent density of an SBM has an effect of the solid volume fraction, which is now defined via,



where the volume fraction of the sbm is defined via,



That is, it is the ratio of the apparent density over the true density of the sbm. The fluid volume fraction, or porosity, is then defined via .

Equation is implicit, but in the code it is not implemented implicitly. In fact, it appears that the solid-bound molecules do not contribute to the residual, making the code converge even when sbms concentrations are not converged yet. For first order reactions A 🡪 B, where B is SBM it appears to work, but not sure about B🡪A or more complex equations. Perhaps I need to implement a predictor-corrector approach.

# Analytical Solution

We try to find the analytical solution for the problem



with prescribed solution on one end.

The **M** matrix is given by



Let’s consider a 1D element with 2 nodes.



Thus,





The prescribed solution is given by  and . Thus, . From this,



Or in general,



Assuming the increment in *u1* is *d*, then



In 3D



Thus,



# Prescribed concentrations in NL analyses





# Analytical solutions for diffusion equations

## Example 1

Consider the following 1D - IC/BC problem,



The solution is given by,



## Example 2

Consider the following 1D - IC/BC problem,



The solution is given by,



# Analytical solutions for reaction equations

## Example 1

The reaction



has the following solution



## Example 2

The reactions,



has the following solution (for )



If , then the solution is

