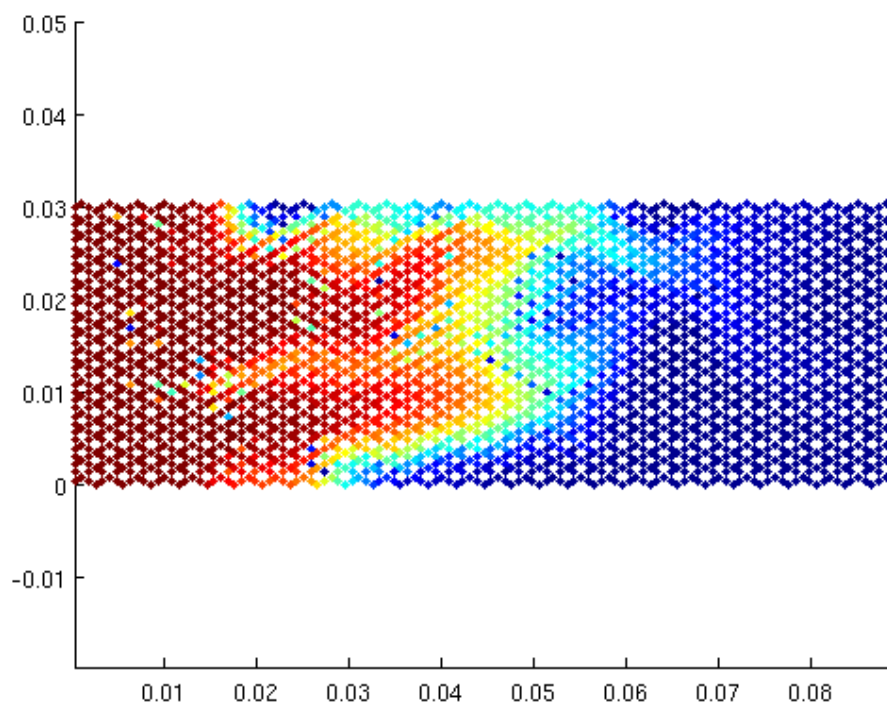


# PN – BRNS<sup>©</sup>

A PORE-NETWORK MODEL  
SIMULATING FLOW AND ADVECTIV-DIFFUSIVE  
TRANSPORT OF CHEMICAL SPECIES  
IN HETEROGENEOUS PORE NETWORKS



## LEGAL CONSIDERATIONS

Copyright © 2013, Mehdi Gharasoo, Falk Heße. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the Helmholtzcentre for Environmental Research (UFZ) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

PN-BRNS IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL Mehdi Gharasoo OR Falk Heße BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Martin Thullner  
UFZ - Helmholtz Center for Environmental Research  
Department of Computational Hydrosystems  
Permoserstr. 15  
D-04318 Leipzig  
GERMANY

e-mail: [martin.thullner@ufz.de](mailto:martin.thullner@ufz.de)

# 1 Introduction and concepts

PN-BRNS was created for generating 2-dimensional pore networks and simulating flow and transport.

The main features of PN-BRNS are:

- Generation of a two-dimensional hexagonal pore-scale network structure.
- Using randomly assigned radii for the pores. The one-point distribution of the pore radii can either follow a normal or log-normal distribution, whereas the two-point distribution of the radii (variogram, correlation or covariance function) may follow a Gaussian, exponential, power-law or truncated power-law model function.
- Flow is generated according to a prescribed pressure drop along the flow direction.
- Reactive transport is computed for a (theoretically) arbitrary number of chemical species.

Additional information on the code and some benchmark calculations are published by Gharasoo et al. [2012]. For more information on the used random field generator we refer to [Kramer et al., 2007] as well as the Biogeochemical Reaction Network Simulator, which is used for the solving of the reactive part we refer to Regnier et al. [2002] or Centler et al. [2010].

## 2 Installation and Use in a Matlab Environment

### 2.1 Prerequisites

To use the functionality of PN-BRNS you need a working Matlab environment. For the calculation of simple reactive transport simulations you can use the pre-defined Matlab routines. In case of more complex reaction networks we provide binaries of the Biogeochemical Reaction Network Simulator (BRNS) library. In order to create libraries of BRNS yourself you have to contact Pierre Regnier as the copyright holder. You can however, also use other software and couple it with PN-BRNS on your own.

### 2.2 Specifying the platform

Matlab itself is a cross-platform application available both for Windows-like (Windows XP, Windows Vista, Windows 7) systems as well as Unix-like (Linux, Mac OSX, ) systems. However, both types of operating systems have different implementation of shared libraries, which are used by PN-BRNS in order to solve the reaction of the transported species. As a result the type of architecture has to be specified. To that end a file called `conf.txt` is used, which is located in the `arch` subdirectory (located in the main directory). In this file the type of the operating system has to be given (windows or unix) as well as the names of the reactive library and its header file.

## 3 Reference Manual

### 3.1 Basic structure

PN-BRNS is made up of several Matlab files, which in conjunction provide the functionality described above. In the following these files are listed and explained.

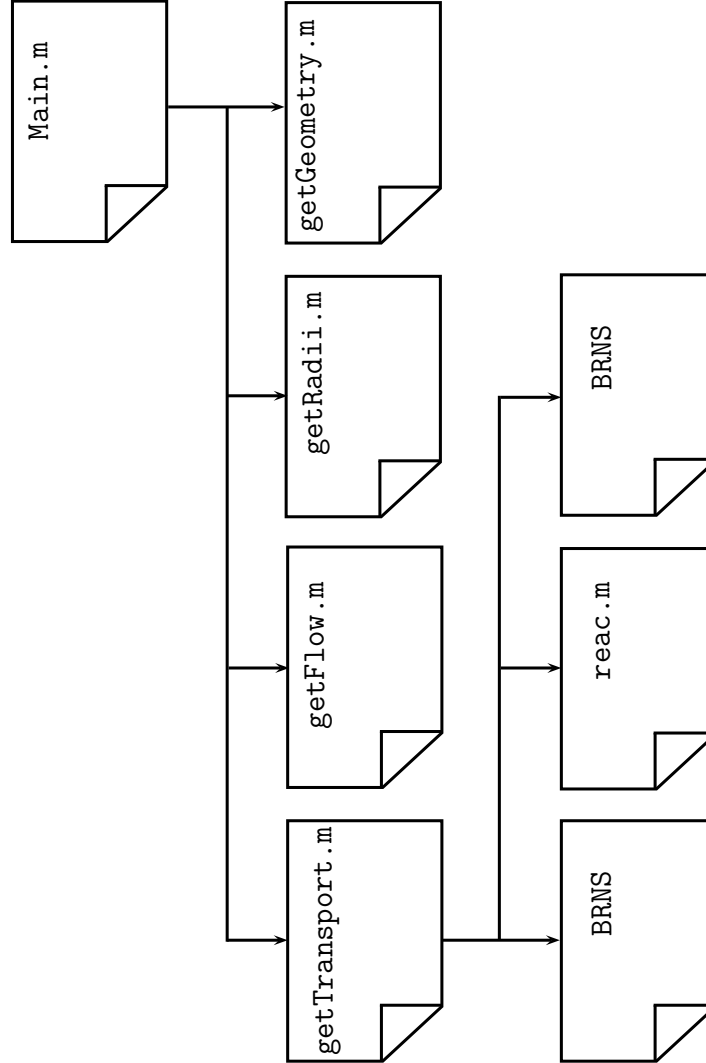


Figure 3.1: Schematic of the different files.

The first file is `getGeometry.m`, where the skeleton of the network is build (see schematic in Figure 3.1). In the second step random values are assigned to each pore in the `getRadii.m` file. The flow field is solved for the network in the `getFlow.m` file. In the last step the (reactive) transport for several species is computed in the `getTransport.m` file.

### 3.2 Creating the geometry

In the first step the raw geometry of the pore space without any additional information is generated. The relevant parameters are

- **n**, which should be a factor of 4 and represents the X axis,
- **m**, which should be even and represents Y axis and
- **l\_pore**, which represents the pore length  $l_{pore}$  in *m*.

These parameters are grouped together and passed to the `getGeometry.m` function. The return value of this function is the **struct** variable `GeometryData` containing the relevant grid data.

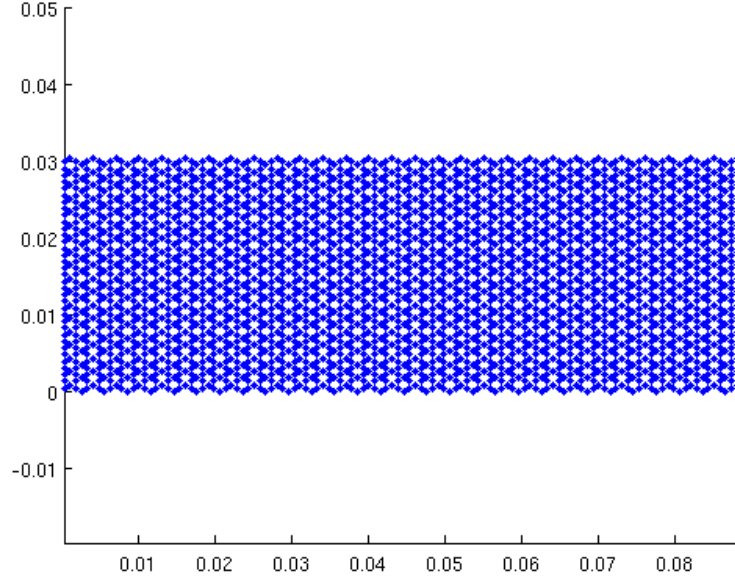


Figure 3.2: Example geometry of the pore-space network. The used parameters were:  $n = 120$ ,  $m = 36$  and  $l_{pore} = 0.001$  (given in meter). This resulted in network with  $n_{pore} = 3162$  number of pores and  $n_{node} = 2160$  number of nodes. The dimensions were  $L_x = 0.09$  *m* and  $L_y = 0.03$  *m* in *x* and *y* directions respectively.

The result is the raw geometry of the pore network(see Figure 3.2), which is used in the following for the calculation of flow and transport.

The parameters **n** and **m** need some further explanation since their values do not straight forwardly apply to the generated pore structure. The layout of the structure is a hexagonal honey-comb network of interconnected tubes having an angle of  $\varphi = 120^\circ$ . Here these tubes represent the pores and effects appearing on the nodes are neglected. The hexagonal grid has been chosen over a simpler square grid since the latter produces unphysical effects due to an over-regular structure [Dupin et al., 2001].

A schematic representation of the hexagonal layout of the pore network can be seen in Figure 3.3, where the dotted lines represent the hypothetical grid, along

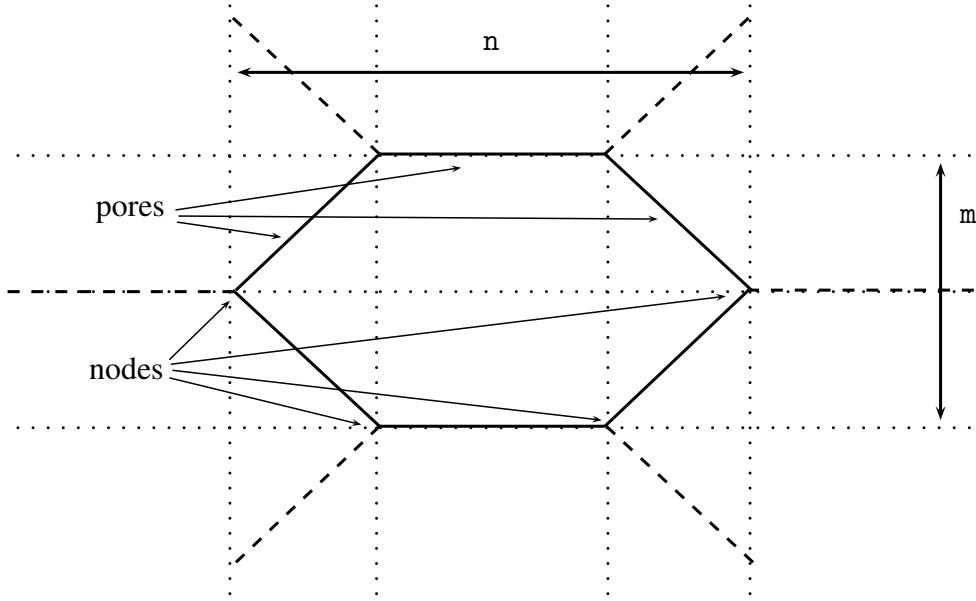


Figure 3.3: Schematic of the hexagonal pore structure.

which the pores are arranged. The values  $n$  and  $m$  are therefore the number of these lines of this hypothetical square grid. A single hexagonal pore therefore needs  $n = 4$  and  $m = 3$ . Due to reasons regarding symmetry the parameter  $n$  should always be a multiple of 4 and  $m$  being a multiple of 2.

As a result of this design decision, it is also not straight forward to compute the number of pores or the physical dimensions of the network, which are generated for a given  $n$  and  $m$ . Whereas the number of nodes is simply  $n_{node} = n \cdot m / 2$ , the number of pores can be calculated according to  $n_{pore} = n/2(m - 1) + (n/2 - 1)\text{ceil}(m/2)$ , where  $\text{ceil}$  rounds towards minus infinity. The length  $L_x$  and height  $L_y$  of the generated pore network can be calculated according to  $L_x = l_{pore}((n/2 - 1)1.5) + 0.5$  and  $L_y = (m - 1)\sin(\varphi)$ . If the physical dimensions are given, the necessary number of elements can be computed according to  $n = 4/3(L_x/l_{pore} - 0.5) + 2$  and  $m = L_y/\sin(\varphi) + 1$ . Both values should be rounded such that above conditions are met.

### 3.3 Assigning random pore radii to the geometry

In the second step random pore radii are generated by the function `getRadii.m` in order to get a heterogeneous pore space. The relevant parameters are

- `lambda_x`, i.e. the correlation length  $\lambda_x$  in the x-direction,
- `lambda_y`, i.e. the correlation length  $\lambda_y$  in the y-direction,
- `mu`, i.e. the expectation value  $\mu$  of the random field,
- `sigma2`, i.e. the variance  $\sigma^2$  of the random field and

- `func`, which defines the variogram function of the random field.
- The type of the one-point distribution.

These parameters are again grouped in a `struct` variable called `RadiiCoeffs`. The return value of the `getRadii.m` function is a list containing the pore radii of every pore.

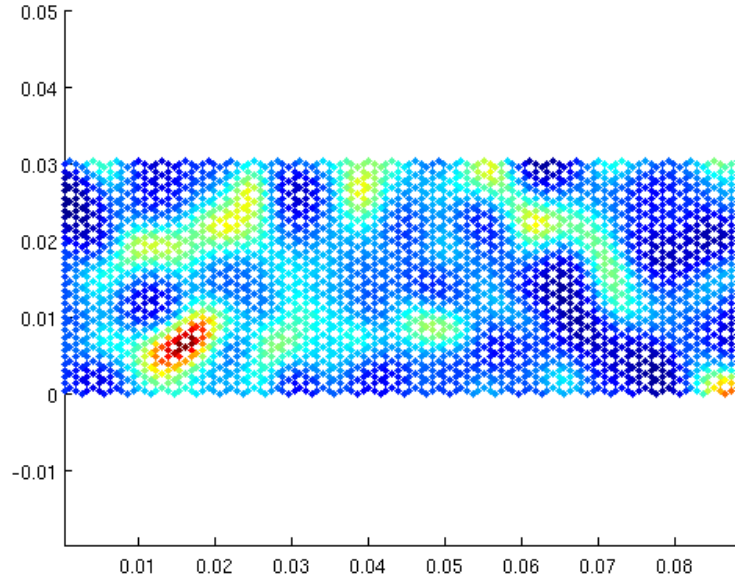


Figure 3.4: Example geometry of the pore-space network with heterogeneous distributed pore radii. The used parameters were:  $\lambda_x = \lambda_y = 0.005$ ,  $\mu = 0.16 \cdot 10^{-3}$  and  $\sigma^2 = 5 \cdot 10^{-9}$ . The field was log-normally distributed and had a Gaussian variogram.

The function uses a random field generator, which can provide random values for the pores. The position is taken to be at the center of each pore. The field generator can provide both a normal as well as a log-normal distribution for the point values. For the variogram several model functions can be generated: a Gaussian, an exponential, a power law as well as a truncated power law. The generator can also provide isotropic and anisotropic fields.

### 3.4 Computing the flow field

In the third step the velocity field is generated and assigned to every pore  $j$  in the network by the function `getFlow.m`. The relevant parameters are

- `delta_p`, which is the pressure drop  $\Delta P$  along the whole geometry.



Since the flow direction is assumed to be from the left to the right,  $\Delta p$  is defined accordingly. The flow parameters are again grouped in a **struct** variable called **FlowCoeffs**. The return value of the **getFlow.m** function is a list containing the flow velocities in every pore.

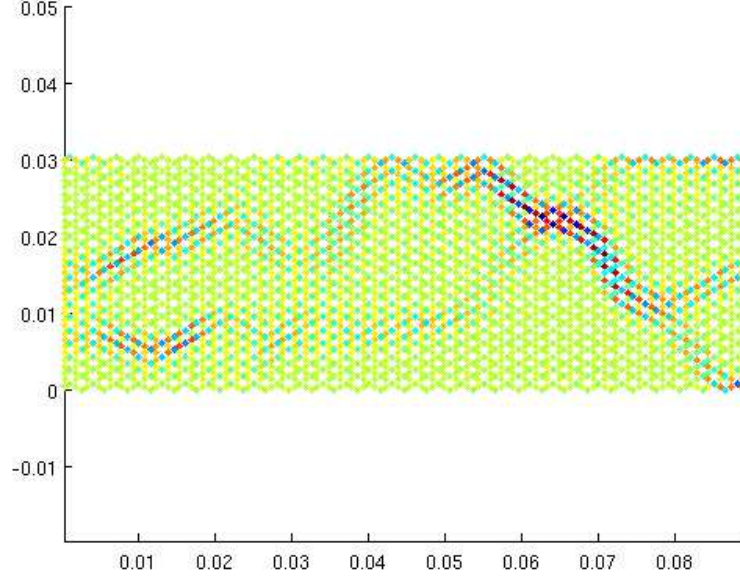


Figure 3.5: Example geometry of the pore-space network with heterogeneous distributed pore radii. The used parameters were:  $\lambda_x = \lambda_y = 0.005$ ,  $\mu = 0.16 \cdot 10^{-3}$  and  $\sigma^2 = 5 \cdot 10^{-9}$ . The field was log-normally distributed and had a Gaussian variogram.

For the solution the Hagen Poiseuille equation is used in order to compute the volumetric flow in each pore:

$$q_j = \frac{\pi r_j^4}{8\nu} \frac{\Delta p_j}{l_{pore}}.$$

Here  $\nu$  is the dynamic viscosity of water and  $\Delta p_j$  is the pressure drop along pore  $j$ . Interpreting above equation by using Darcy's law we get  $K = (\pi r_j^4 / 8\nu l_{pore})$  as the conductivity of each pore. Mass conservation is achieved by using Kirchhoff's law, i.e.  $q_{j,1} + q_{j,2} + q_{j,3} = 0$  in case of a hexagonal grid (see Figure 3.3). By virtue of Kirchhoff's law and using above equation one can derive a system of  $n_{pore}$  equations, which can be solved for the  $n_{pore}$  unknowns of  $q_j$ .

The flow is then used in order to determine the water velocity in each pore  $j$  according to

$$u_j = \frac{q_j}{\pi r_j^2}.$$

These data are then used in order to determine the advective transport of each species.

### 3.5 Solving the transport

In the fourth step the time-dependent (reactive) transport is solved for a number of  $n_{spec}$  chemical species in each pore  $j$  in the function `getTransport.m`. The relevant parameters used for the solution of the transport are

- `t_end`, which is the length of the time interval,
- `t_0`, which is the beginning of the time interval,
- `dt`, which is the length of the step size  $\Delta t$  and
- `spec_no`, which is the number  $n_{spec}$  of chemical species.

These parameters are again grouped in a **struct** variable called `TransportCoeffs`. The return value of the `getTransport.m` function is a list containing the concentration values of every species  $i$  in every pore  $j$ .

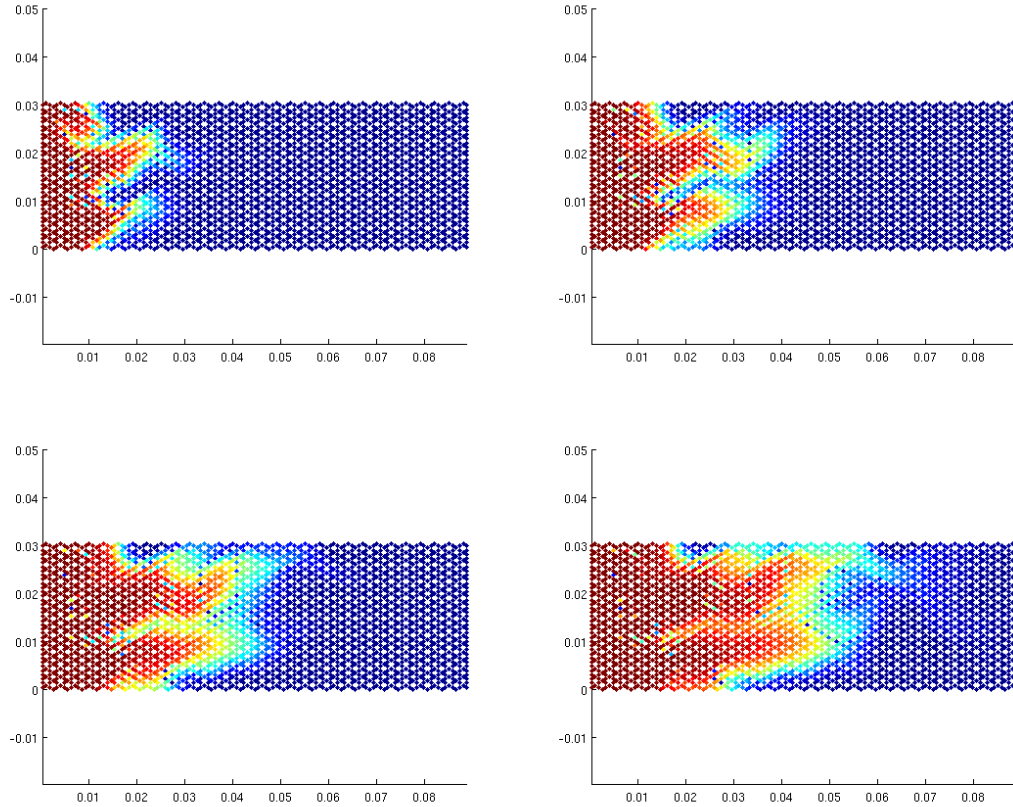


Figure 3.6: Transport of a conservative tracer at four consecutive time steps.

The transport step is solved using the advection-diffusion-reaction equation within each pore  $j$ , i.e.

$$\frac{\partial}{\partial t} c_{i,j} = -\frac{\partial}{\partial x_L} \cdot (u_j c_{i,j}) + D_i \left( \frac{\partial^2}{\partial x_L^2} + \frac{\partial^2}{\partial x_T^2} \right) c_{i,j} + r_{i,j}. \quad (3.1)$$

Here  $t$  is the time,  $c_{i,j}$  is the concentration of species  $i$  in pore  $j$ ,  $D_i$  is the diffusion coefficient of each species and  $R_{i,j}$  the respective production and consumption rate. Several methods can be employed for the reaction part. The directions  $x_L$  and  $x_T$  are the longitudinal and transversal directions within each pore. Two different scenarios are possible according to the hexagonal geometry (see Figure 3.3): (i) a horizontal pore, where  $x_L = x$  and  $x_T = y$  as well as (ii) an inclined pore, where  $x_L$  and  $x_T$  are inclined, too.

Instead of the full two-dimensional system given by Equation (3.1) we only solve a one-dimensional system averaged along the width of the pore. This system is given by the following expression

$$\frac{\partial}{\partial t} \mathbf{C}_j = -\frac{\partial}{\partial x_L} \cdot (\mathbf{U}_j \mathbf{C}_j) + \mathbf{D} \frac{\partial^2}{\partial x_L^2} \mathbf{C}_j + \mathbf{R}_j. \quad (3.2)$$

Here  $\mathbf{C}_j$  is the list of the averaged concentration of all species and  $\mathbf{U}_j$ ,  $\mathbf{D}$  and  $\mathbf{R}_j$  are lists of the appropriate velocity, diffusion and reaction components. For a derivation of this coefficient functions we refer to [Heße et al., 2010].

In order to solve Equation (3.2) for every pore a simple operator splitting procedure is applied, where the different terms like advection, longitudinal diffusion as well as reaction are solved successively. For the discretization of the time derivative a simple explicit Euler scheme is employed, i.e.

$$\frac{\partial}{\partial t} \mathbf{C}_j \approx \frac{\Delta \mathbf{C}_j}{\Delta t} = \frac{\mathbf{C}_j^n - \mathbf{C}_j^{n-1}}{\Delta t}.$$

In order to avoid unstable solutions, which may arise with this method, the Courant-Friedrichs-Levy conditions must be met.

**Advection** According to Equation (3.2) we get for the advection the following expression

$$\left( \frac{\partial}{\partial t} \mathbf{C}_j \right)_{\text{adv}} = -\frac{\partial}{\partial x_L} \cdot (\mathbf{U}_j \mathbf{C}_j).$$

Like the time derivative, the spatial derivative in this expression is also approximated by using a difference method, i.e. the differential is replaced with a difference, leading to the following expression

$$(\Delta \mathbf{C}_j)_{\text{adv}} = -\frac{\Delta t}{\Delta x} \mathbf{U}_j (\mathbf{C}_{\text{in},j}^{n-1} - \mathbf{C}_j^{n-1}) \quad (3.3a)$$

$$\mathbf{C}_{\text{in}} = \frac{\sum q_k \mathbf{C}_k}{\sum q_k} \quad (3.3b)$$

Here the  $\mathbf{C}_{\text{in},j}$  is the list of inlet concentrations for each pore  $j$ . Due to the hexagonal geometry the calculation depends on the type of node.

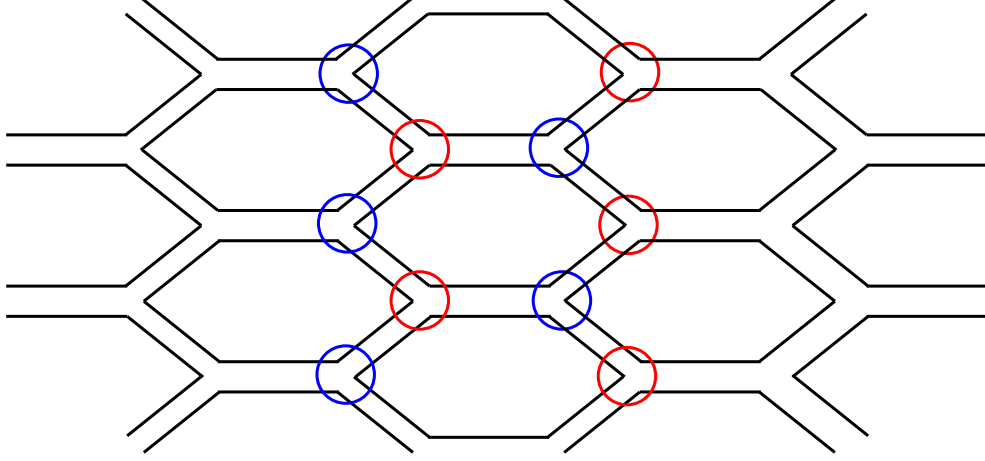


Figure 3.7: The two different node types of the hexagonal geometry structure.

Two different types of nodes can be distinguished (see Figure 3.7). The first type is where a horizontal pore is branching into two inclined pores. Every inclined pore therefore has only one inlet and the concentration is divided according to the fluxes. The second type of node is where two inclined pores merge into one horizontal pore. Here the inlet concentration is derived by summing up both predecesing concentration according to the fluxes.

**Longitudinal Diffusion** According to Equation (3.2) we get for the longitudinal diffusion the following expression

$$\left( \frac{\partial}{\partial t} \mathbf{C}_j \right)_{\text{dif}_L} = \mathbf{D} \frac{\partial^2}{\partial x_L^2} \mathbf{C}_j.$$

Like in case of the advective flux we chose again a difference approximation leading to the following expression

$$(\Delta \mathbf{C}_j)_{\text{dif}_L} = \Delta t \mathbf{D} \frac{\mathbf{C}_{j-1}^{n-1} - 2\mathbf{C}_j^{n-1} + \mathbf{C}_{j+1}^{n-1}}{\Delta x^2/2},$$

where  $\mathbf{C}_{j-1}$  and  $\mathbf{C}_{j+1}$  are the concentrations at the upstream and downstream nodes respectively.

**Transversal Diffusion** No transversal diffusion is featured in Equation (3.2) due to the averaging across the width of the pore.

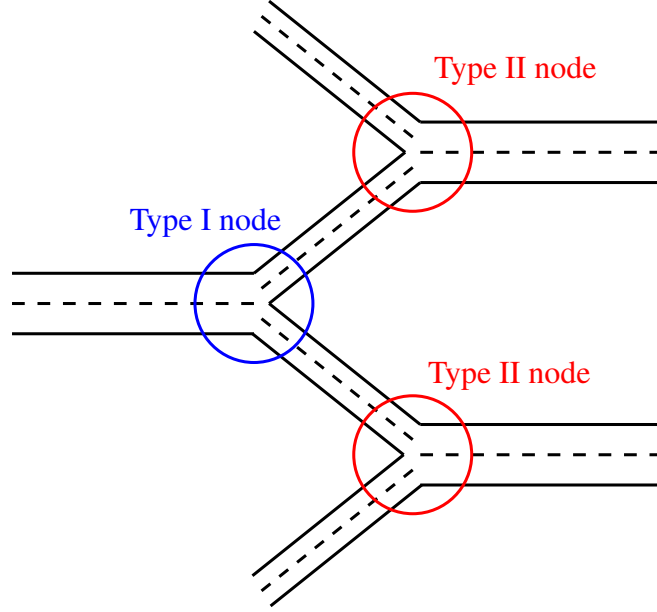


Figure 3.8: Discrimination of upper and lower pore regions in order to emulate transversal diffusion.

We can however, define an upper and a lower concentration for each pore  $j$  (see Figure 3.8) leading to the following approximation of the transversal diffusion

$$\left( \frac{\partial}{\partial t} \mathbf{C}_j \right)_{\text{dif}_T} = \mathbf{D} \frac{\partial^2}{\partial x_T^2} \mathbf{C}_j.$$

Like in case of the longitudinal diffusional flux we chose again a difference approximation leading to the following expression

$$(\Delta \mathbf{C}_j)_{\text{dif}_T} = \Delta t \mathbf{D} \frac{\pi^2}{4} \frac{(\mathbf{C}_{\text{up},j}^{n-1} - \mathbf{C}_{\text{down},j}^{n-1})}{\Delta y_j^2 / 2},$$

where  $\Delta y_j$  is the width of the pore  $j$  and  $\mathbf{C}_{\text{up},j}$  and  $\mathbf{C}_{\text{down},j}$  are the concentrations at the upper and lower part of the pore  $j$  respectively.

To fully implement upper and lower pore regions the advective part had to be adapted as well. The Kirchhoff rule for the different nodes have therefore to be rewritten dependent on the node type.

Even in a simple hexagonal structure with uniform mean average flow, the situation is complex due to different flow paths that the solute can take (see Figure 3.9). For the geometry as well as the used boundary conditions several types of nodes can therefore be distinguished.

The most simple type is the inlet type (Type I in Figure 3.10) as well as pores at the upper and lower boundary of the medium (Type II in Figure 3.10). Here

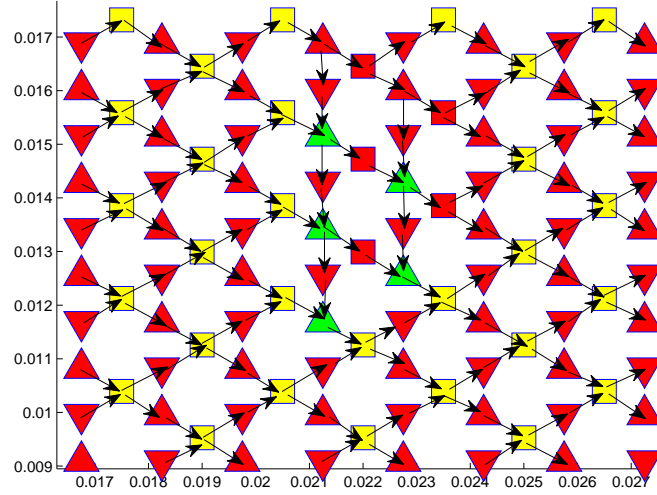


Figure 3.9: Graphical representation of the different node types. The color of the pore denotes the type of inlet node, i.e. the node upstream of the pore. Yellow pores are straight pores with two inlets. Green pores are tilted pores with two inlets. Red pores are pores with one inlet. The arrows indicate the flow direction.

no adjustments are necessary, since the inlet and outlet pores connect in a simple forward manner.

The second most simple type are nodes, where the upstream pores are all left of the pore. These can be either a tilted pore with one inlet from a straight pore (Type III in Figure 3.10) or a straight pore with two inlets from tilted pores (Type IV in Figure 3.10). Note that in a medium without complex flow paths, this would already cover all possible node types. All additional node types are therefore due to downwards or upwards flow. The resulting node types are consequently identical to the one of these two node types, but with different arrangements of pores.

**Type III node** In case of a Type III node, we can still use Kirchhoff' Theorem to determine the overall average concentration. For a pore with a single upstream pore the inlet concentration is simply the same concentration as the outlet concentration of the upstream pore so

$$C_1 = C_3$$

for the upwards pointing pore and

$$C_2 = C_3$$

for the downwards pointing pore. In case of pores, separated into upper and lower parts, this relationship must still be true for average concentrations within the

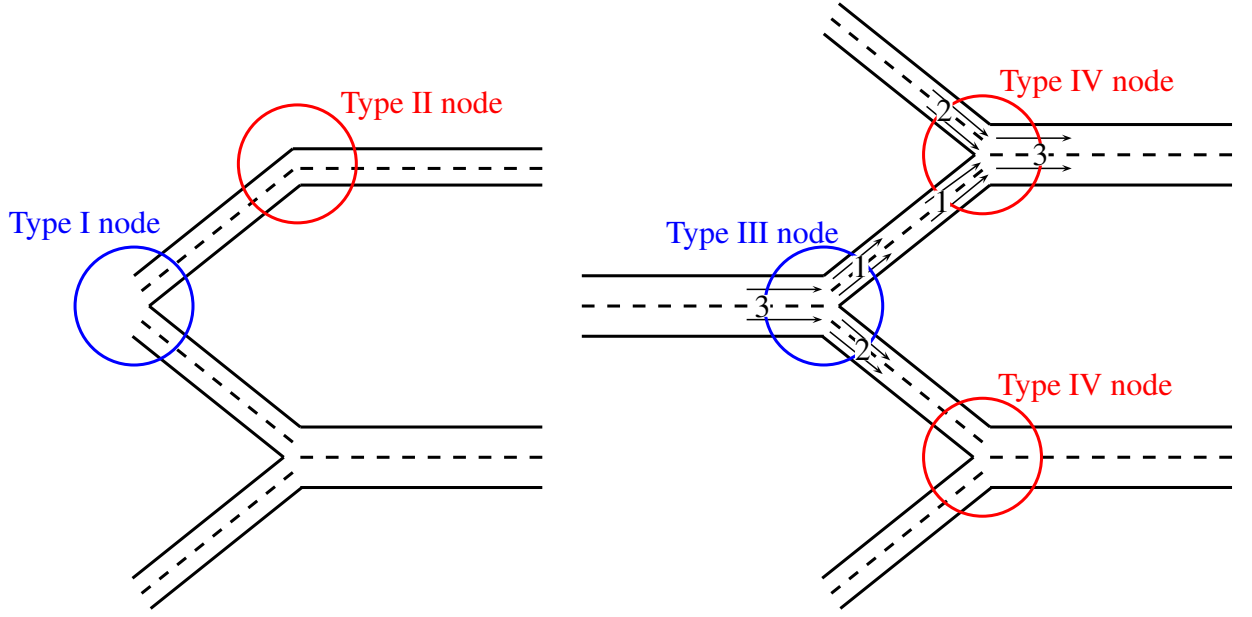


Figure 3.10: Type I, Type II, Type III and Type IV nodes.

pores. The task is therefore to use this fact and adapt it to the new situation, with it much more complex array of channels within each pore.

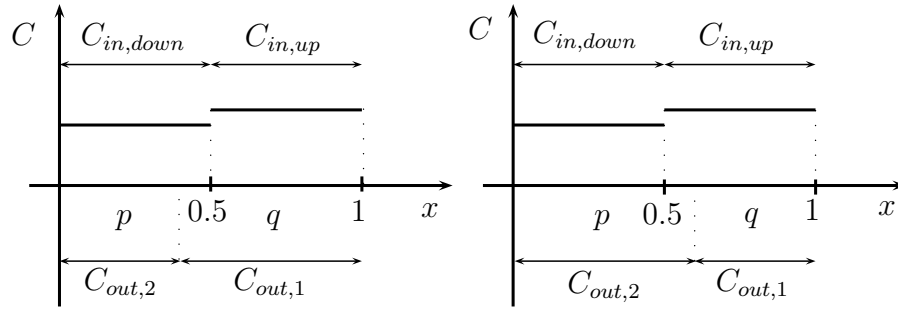


Figure 3.11: Distribution of inlet concentrations for a single inlet pore.

This array of channels as well as the respective concentrations within each channel is displayed in Figure 3.11. Two cases can be distinguished, depending whether Pore 2 or Pore 3 is bigger (compare Figure 3.10 and 3.11). Since the amount of solute entering each pore is determined by the flux, we can define the ratios according to

$$p = \frac{q_2}{q_1 + q_2} = \frac{q_2}{q_3} \quad (3.4a)$$

$$q = \frac{q_1}{q_1 + q_2} = \frac{q_1}{q_3} = 1 - p \quad (3.4b)$$

The total amount of mass leaving the incoming pore is given by the average concentration. The partition of these mass into each channel in the outletting pores is then determined according to the two cases mentionned above. In the first case, i.e. for  $q > p$

$$m_{2,\text{down}} = \frac{p}{2} C_{\text{in,down}} + 0 C_{\text{in,up}} \quad (3.5a)$$

$$m_{2,\text{up}} = \frac{p}{2} C_{\text{in,down}} + 0 C_{\text{in,up}} \quad (3.5b)$$

$$m_{1,\text{down}} = \left( \frac{1}{2} - p \right) C_{\text{in,down}} + \frac{p}{2} C_{\text{in,up}} \quad (3.5c)$$

$$m_{1,\text{up}} = 0 C_{\text{in,down}} + \frac{q}{2} C_{\text{in,up}} \quad (3.5d)$$

as well as in case two, i.e. for  $p > q$

$$m_{2,\text{down}} = \frac{p}{2} C_{\text{in,down}} + 0 C_{\text{in,up}} \quad (3.6a)$$

$$m_{2,\text{up}} = \frac{q}{2} C_{\text{in,down}} + \left( \frac{1}{2} - q \right) C_{\text{in,up}} \quad (3.6b)$$

$$m_{1,\text{down}} = 0 C_{\text{in,down}} + \frac{q}{2} C_{\text{in,up}} \quad (3.6c)$$

$$m_{1,\text{up}} = 0 C_{\text{in,down}} + \frac{q}{2} C_{\text{in,up}} \quad (3.6d)$$

Using these values, the proportions of each channel are determined and can be assigned.

**Type IV node** In case of a Type IV node, we first can determine the overall solute flux into the pore according to Kirchhoff's law. Using the notation given in Figure 3.10 (b) we can write

$$\mathbf{C}_3 q_3 = \mathbf{C}_1 q_1 + \mathbf{C}_2 q_2 \quad (3.7a)$$

$$(\mathbf{C}_{\text{up},3} + \mathbf{C}_{\text{down},3}) = \frac{(\mathbf{C}_{\text{up},1} + \mathbf{C}_{\text{down},1}) q_1 + (\mathbf{C}_{\text{up},2} + \mathbf{C}_{\text{down},2}) q_2}{q_3}. \quad (3.7b)$$

This means, we need an additional equation for the determination of the individual concentrations in the downstream pore.

The motivation for the derivation of this equation is shown in Figure 3.12. In this picture, the four different bars represent the four concentration values in the two upstream pores of the node. The width of the bars is corresponding to the fluxes in the incoming pores and the median of the x-axis shows the partition of these fluxes to the outletting pore. There two different case are distinguished: First, where the flux of Pore 1 is smaller than in Pore 2 and second, where the



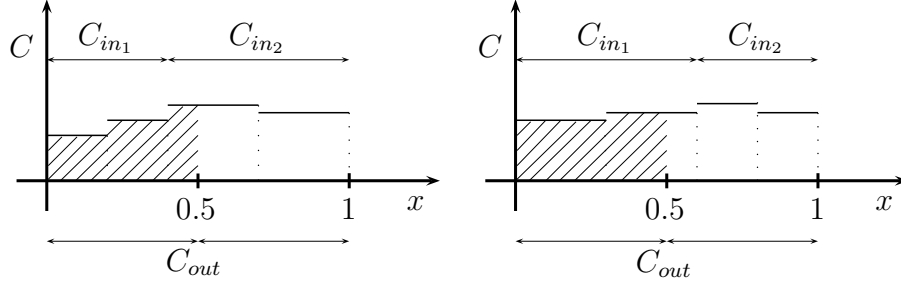


Figure 3.12: Distribution of inlet concentrations for a double inlet pore.

opposite is the case. The task is now to compute the relative contribution of the incoming concentrations to the outgoing upper and lower pore. This latter is specified in Figure by the hatched area. Using this information, it is clear that the normalized incoming concentration, i.e. the area underneath the curve is

$$m = \frac{p}{2}C_{\text{in1,down}} + \frac{p}{2}C_{\text{in1,up}} + \frac{q}{2}C_{\text{in2,down}} + \frac{q}{2}C_{\text{in2,up}},$$

with  $p = q_1/q_3$  and  $q = 1 - p = q_2/q_3$  being the ratios of the respective incoming fluxes to the total incoming flux. The area underneath the hatched is then in case one, i.e. for  $q > p$

$$m_{\text{down}} = \frac{p}{2}C_{\text{in1,down}} + \frac{p}{2}C_{\text{in1,up}} + \left(\frac{1}{2} - p\right)C_{\text{in2,down}} + 0 C_{\text{in2,up}} \quad (3.8a)$$

$$m_{\text{up}} = 0 C_{\text{in1,down}} + 0 C_{\text{in1,up}} + \left(\frac{1}{2} - \frac{q}{2}\right)C_{\text{in2,down}} + \frac{q}{2}C_{\text{in2,up}} \quad (3.8b)$$

as well as in case two, i.e. for  $p > q$

$$m_{\text{down}} = \frac{p}{2}C_{\text{in1,down}} + \left(\frac{1}{2} - \frac{p}{2}\right)C_{\text{in1,up}} + 0 C_{\text{in2,down}} + 0 C_{\text{in2,up}} \quad (3.9a)$$

$$m_{\text{up}} = 0 C_{\text{in1,down}} + \left(\frac{1}{2} - q\right)C_{\text{in1,up}} + \frac{q}{2}C_{\text{in2,down}} + \frac{q}{2}C_{\text{in2,up}}. \quad (3.9b)$$

The above equations can be interpreted as a scalar product of two vectors. Let us therefore introduce the following convention

$$C_{\text{in}} = \begin{pmatrix} C_{\text{in1,down}} \\ C_{\text{in1,up}} \\ C_{\text{in2,down}} \\ C_{\text{in2,up}} \end{pmatrix}$$

Using the above equations as well as this notation, we can derive the relative contribution of the two concentrations for the case of  $q > p$  by

$$r = \frac{m_{\text{down}}}{m} = \frac{\left(\frac{p}{2} \frac{p}{2} \left(\frac{1}{2} - p\right) 0\right) \cdot C_{\text{in}}}{\left(\frac{p}{2} \frac{p}{2} \frac{q}{2} \frac{q}{2}\right) \cdot C_{\text{in}}}$$

as well as in case  $p > q$

$$r = \frac{m_{\text{down}}}{m} = \frac{\left(\frac{p}{2} \left(\frac{1}{2} - \frac{p}{2}\right) 0 0\right) \cdot C_{\text{in}}}{\left(\frac{p}{2} \frac{p}{2} \frac{q}{2} \frac{q}{2}\right) \cdot C_{\text{in}}}$$

Note that the case of  $p = q$  is already covered, since it means that the concentrations are divided up according to the pore connected upstream to the pore.

Now that the relative contribution of the two incoming pores to the upper and lower channel of the outgoing pore has been determined, we can state the equation for the calculation of the respective channel concentrations

$$C_{\text{out,down}} = 2 C_{\text{out,down}} r \quad (3.10a)$$

$$C_{\text{out,up}} = 2 C_{\text{out,down}} (1 - r). \quad (3.10b)$$

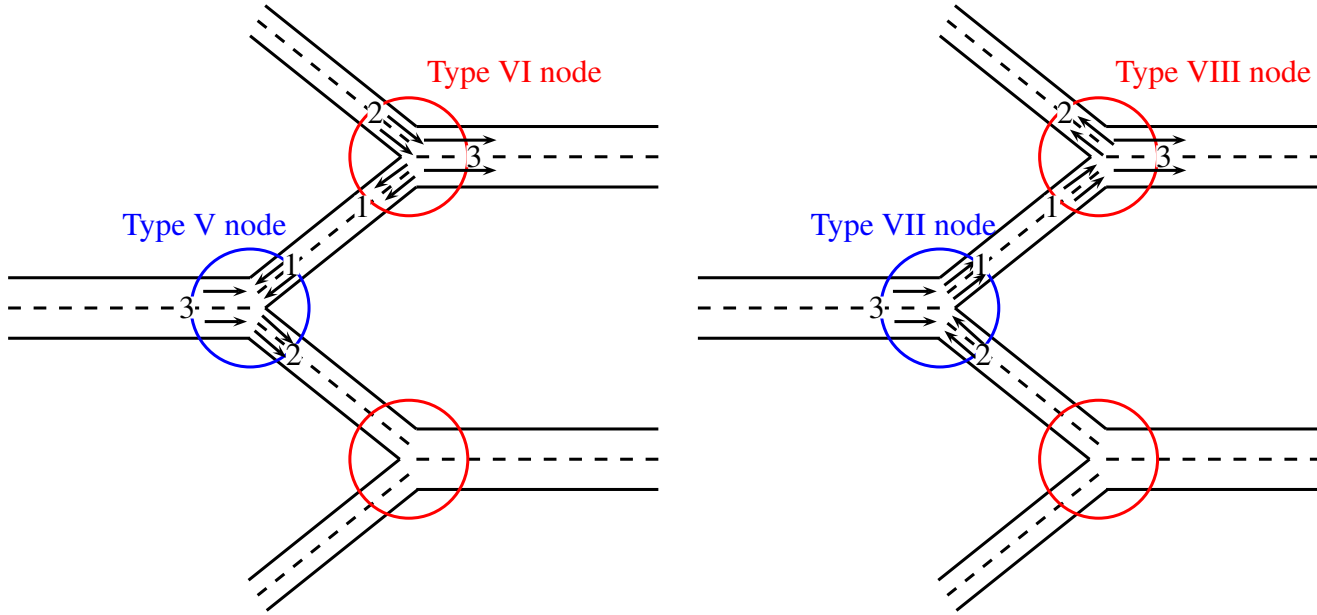


Figure 3.13: Type V, Type VI, Type VII and Type VIII nodes.

### Type V node

**Reaction** In the conservative case, i.e. without reaction, the code can already compute the movement of a tracer. For simple reactions or reaction networks matlab code can be written by the user. In case of more complex reaction networks we can provide binaries of a library, which can solve a variety of reactions,

which are described by algebraic terms or kinetic non-equilibrium reactions given by ordinary differential equations. These libraries have been generated by encapsulating the reactive module of the Biogeochemical Reaction Network Simulator (BRNS). In order to generate libraries on your own you need to contact Pierre Regnier, who holds all property rights to the software as well as Florian Centler, who generated the libraries. As an alternative you can use any other reaction network simulator and connect it to this code.

# Bibliography

- Florian Centler, Haibing Shao, Cecilia De Biase, Chan-Hee Park, Pierre Regnier, Olaf Kolditz, and Martin Thullner. GeoSysBRNS-A flexible multidimensional reactive transport model for simulating biogeochemical subsurface processes. *Computers & Geosciences*, 36(3):397–405, Mar 2010. ISSN 0098-3004. doi: {10.1016/j.cageo.2009.06.009}.
- Hubert J. Dupin, Peter Kitanidis, and Perry L. Mc Carty. Simulations of two-dimensional modeling of biomass aggregate growth in network models. *Water Resources Research*, 37:2981–2994, 2001.
- Mehdi Gharasoo, Florian Centler, Pierre Regnier, Hauke Harms, and Martin Thullner. A reactive transport modeling approach to simulate biogeochemical processes in pore structures with pore-scale heterogeneities. *Environmental Modelling & Software*, 30:102 – 114, 2012. doi: 10.1016/j.envsoft.2011.10.010.
- Falk Heße, Hauke Harms, Sabine Attinger, and Martin Thullner. Linear Exchange Model for the Description of Mass Transfer Limited Bioavailability at the Pore Scale. *Environmental Science & Technology*, 44(6):2064–2071, Mar 15 2010. ISSN 0013-936X. doi: {10.1021/es902489q}.
- Peter R. Kramer, Orazgeldi Kurbanmuradov, and Karl Sabelfeld. Comparative analysis of multiscale Gaussian random field simulation algorithms. *Journal of Computational Physics*, 226(1):897–924, Sep 10 2007. ISSN 0021-9991. doi: {10.1016/j.jcp.2007.05.002}.
- P Regnier, JP O’Kane, CI Steefel, and P Vanderborght. Modeling complex multi-component reactive-transport systems: towards a simulation environment based on the concept of a Knowledge Base. *Applied Mathematical Modelling*, 26(9):913–927, Sep 2002. ISSN 0307-904X. doi: {10.1016/S0307-904X(02)00047-1}.