

Report

P-15-08

August 2018



User's manual of iDP version 3.2

Hedieh Ebrahimi

Paolo Trincherò

Jorge Molinero

Luis Manuel de Vries

SVENSK KÄRNBRÄNSLEHANTERING AB

SWEDISH NUCLEAR FUEL
AND WASTE MANAGEMENT CO

Box 3091, SE-169 03 Solna
Phone +46 8 459 84 00
skb.se

SVENSK KÄRNBRÄNSLEHANTERING

ISSN 1651-4416

SKB P-15-08

ID 1512655

August 2018

User's manual of iDP version 3.2

Hedieh Ebrahimi, Paolo Trincherio,
Jorge Molinero, Luis Manuel de Vries
Amphos 21

This report concerns a study which was conducted for Svensk Kärnbränslehantering AB (SKB). The conclusions and viewpoints presented in the report are those of the authors. SKB may draw modified conclusions, based on additional literature sources and/or expert opinions.

Data in SKB's database can be changed for different reasons. Minor changes in SKB's database will not necessarily result in a revised report. Data revisions may also be presented as supplements, available at www.skb.se.

A pdf version of this document can be downloaded from www.skb.se.

© 2018 Svensk Kärnbränslehantering AB

Abstract

PFLOTRAN is an open source, reactive flow and transport simulator for modelling subsurface processes that has been developed with high performance computing and efficiency in mind. To this end it has been built on top of well-known frameworks like MPI, PETSc and HDF. It implements the finite volume method and has been written in Fortran 90, in a modular way. It has demonstrated peta-scale performance (i.e. more than 10^{15} floating point operations per second) in simulations of uranium migration in the superfund Hanford 300 Area as part of the SciDAC-2 programme (Hammond and Lichtner 2011).

A powerful software for the simulation of groundwater flow in crystalline, fractured media is DarcyTools (Svensson and Ferry 2014). This software, owned by SKB, has been developed over the last 15 years and has been used in many different site investigation projects. Just like PFLOTRAN it is based on the finite volume method and uses MPI for its parallelization.

These two simulation codes are combined into a framework for reactive transport modelling in crystalline fractured media. This framework is denoted as DarcyTools-PFLOTRAN Interface (iDP; Molinero et al. 2016). The interface is written in a modular, cross-platform and extensible way using Fortran and Python.

This document is the User's Manual of iDP. Details of the code architecture are first provided. The different configuration files and running operations are then exhaustively described. Benchmark tests are also presented and discussed.

Sammanfattning

PFLOTRAN är en reaktiv flödes- och transportsimulator, med öppen källkod, för modellering av underjordiska processer och har utvecklats med högpresterande beräkningar och effektivitet i åtanke. För detta ändamål har den byggts på välkända grunder såsom MPI, PETSc och HDF. Den implementerar den finita volymmetoden och har skrivits modulärt i Fortran 90. Den har uppvisat prestanda i peta-skalan (det vill säga mer än 10^{15} flyttalsoperationer per sekund) för simuleringar av uranmigration i projektet ”superfund Hanford 300-Area” som är en del av SciDAC-2-programmet (Hammond och Lichtner 2011).

En kraftfull programvara för simulering av grundvattenflöde i kristallina, sprickiga medier är DarcyTools (Svensson och Ferry 2014). Denna programvara, som ägs av SKB, har utvecklats under de senaste 15 åren och har använts i många olika platsundersökningar. Precis som PFLOTRAN är den baserad på den finita volymmetoden och använder MPI för dess parallellisering.

Dessa två simuleringskoder kombineras i ett ramverk för reaktiv transportmodellering i kristallina sprickmedier. Detta verktyg betecknas DarcyTools-PFLOTRAN Interface (iDP; Molinero et al. 2016). Gränssnittet är skrivet på ett modulärt, plattformsoberoende och utökningsbart sätt med användande av Fortran och Python.

Detta dokument är användarmanualen för iDP. Detaljer om kodarkitekturen tillhandahålls först. De olika konfigurationsfilerna och instruktioner hur man använder verktyget beskrivs sedan fullständigt. Olika testfall presenteras och diskuteras även.

Contents

1	Introduction	7
1.1	Background	7
1.2	Overview of iDP	7
1.3	Objective and scope	8
2	History of changes	9
2.1	What has changed in iDP V3.0	9
2.2	What has changed in V3.2	10
3	Introductory implementation details	11
3.1	Application architecture	11
4	Installation (Windows systems)	13
4.1	System requirements	13
4.1.1	Where to get iDP	13
4.1.2	Installing iDP-DT	13
4.1.3	Installing iDP-PF	15
4.2	Uninstall	17
5	iDP-DT functionality specification	19
5.1	Overview	19
5.2	Process configuration file	19
5.3	Generate PFLOTRAN subdomain	20
5.4	Process DarcyTools binary files	21
5.5	Generate absolute cell balance errors	21
5.6	Process DarcyTools markers	21
5.7	Process grouting cells	21
6	iDP-PF functionality specifications	23
6.1	Overview	23
6.2	Process configuration file	23
6.3	PFLOTRAN settings	24
6.4	File name settings	25
6.5	Handling of regions	25
6.6	Marker functionality	26
6.7	Process velocities	26
6.8	Process porosity and permeability	27
6.9	Process mass balance errors	27
6.10	Generate PFLOTRAN input file	27
7	Running iDP	29
7.1	Introduction	29
7.2	iDP-DT	29
7.2.1	iDP-DT User interface components	30
7.2.2	Running iDP-DT from command line	35
7.2.3	iDP-DT configuration file structure	35
7.3	iDP-PF	37
7.3.1	iDP-PF User interface components	38
7.3.2	Running iDP-PF from command line:	40
7.4	iDP Use cases/benchmarks	40
7.4.1	iDP-DT Use case examples	40
7.4.2	iDP-PF Use case examples	41
	References	55

1 Introduction

1.1 Background

In this chapter, we will start with a brief introduction to iDP. In the chapters following we will get in to all the details regarding installation, running and post processing of iDP results.

PFLOTRAN is a next-generation, open source, reactive flow and transport simulator for modelling subsurface processes that has been developed with high performance computing and efficiency in mind. To this end it has been built on top of well-known frameworks like MPI, PETSc and HDF. It implements the finite volume method and has been written in an object oriented modular way, in FORTRAN 2003. It has demonstrated peta-scale performance in simulations of uranium migration in the superfund Hanford 300 Area as part of the SciDAC-2 programme (Hammond et al. 2014, 2011).

On the other hand, one of the best codes available to simulate groundwater flow in crystalline, fractured media is DarcyTools (Svensson and Ferry 2014). This software, owned by SKB, has been developed over the last 10 years and has been used in many different site investigation projects. Just like PFLOTRAN it is based on the finite volume method and uses MPI for its parallelization.

These two simulation codes are combined into a framework for reactive transport modelling in crystalline fractured media. This framework is denoted as interface between DarcyTools and PFLOTRAN (iDP). The interface is written in a modular, cross-platform and extensible way using FORTRAN and Python.

The combined reactive transport framework has been verified, by means of five benchmark problems, and subsequently tested in 2 relevant case studies for SKB: (1) the degradation of cement grout applied in the Forsmark spent fuel repository and, (2) the penetration of glacial water and oxygen consumption in fractured crystalline rocks (Molinero et al. 2016). These test cases have been run successfully on three supercomputers in Europe: the MareNostrum supercomputer at the Barcelona Supercomputing Center (BSC), and the Juqueen and JURECA supercomputer at the Jülich Supercomputer Centre (JSC). The performance and scalability of iDP has been demonstrated for up to 8 000 cores in both BSC and JSC supercomputers. In addition, during the extreme scalability workshop further scalability tests up to 131 072 cores were performed on JUQUEEN and great scalability for up to one rack of JUQUEEN (16 384 cores) was demonstrated.

1.2 Overview of iDP

iDP can read multiple pseudo-transient flow fields from DarcyTools and generate the input files for pseudo-transient reactive transport simulations in PFLOTRAN.

The output of DarcyTools is converted to binary files and these are then converted to PFLOTRAN input files. Both steps can be run on the same computer, or on separate computers. This allows different “types” of users (the DarcyTools users are typically hydrogeologists whereas PFLOTRAN is mostly used by geochemists) to work independently, if required.

iDP consists of two parts: iDP-DT (iDP-DarcyTools) and iDP-PF (iDP-PFLOTRAN). Both parts use configuration files that provide the required run specifications (e.g. location of input and output files, domain size, geochemical input, etc). The resulting input file for the PFLOTRAN simulation, which includes links to generated HDF5 files containing the information about velocities and hydraulic and transport properties, can be modified manually as well.

iDP-Post is used to post-process the PFLOTRAN HDF5 output files so that they can be used directly in the ParaView visualization software. The post-processing step includes calculating statistics, merging output of different time steps and generating smaller output files for specific species that can be directly opened in ParaView.

1.3 Objective and scope

This report is intended to be a User's Manual for iDP V3.2 and its primary objective is to provide users with an exhaustive overview of the numerical code, including the input steps required to run iDP. In Chapter 2 to 4 the history of changes of the interface is presented, introductory implementation details are discussed and system requirements are specified. Chapter 5 and 6 provide specific implementation details of the two main parts of the interface (iDP-DT and iDP-PF). Chapter 7 is focused on the steps needed to run iDP. In this chapter, the different parts of the Graphical User Interface (GUI) are presented. Chapter 7 also contains details of the different validation tests. These tests are also exhaustively discussed in a companion SKB report (Molinero et al. 2016).

2 History of changes

iDP V3.0 was the first stable version of the interface that included a major improvement: the capability to handle structured irregular grids, while version 3.2 is currently the last released stable version of iDP. Thus, in the following two sub-sections brief details about the changes and new code capabilities introduced in these two versions are presented.

2.1 What has changed in iDP V3.0

The previous version of iDP (i.e. iDP V2.0) only supported regular and structured grids. In iDP version 3.0, the capability to work with structured but irregular grids has been added. The iDP User Interface for iDP V3.0 is the same as in the previous version, while the functionality of the code has been upgraded to be able to support models with structured irregular grids.

From a user point of view changes are:

- PFLOTRAN input file: has been updated by using, when needed, specific cards and configurations related to the topology of the irregular grid.
- HDF5 files: absolute groundwater mass balance errors are now provided (in version 2 relative errors were provided).
- Log files: the log files contain more details about the grid topology.

A reference example regarding structured irregular grid (i.e. structured grid having variable refinement) has been added to the installation package (Figure 2-1).

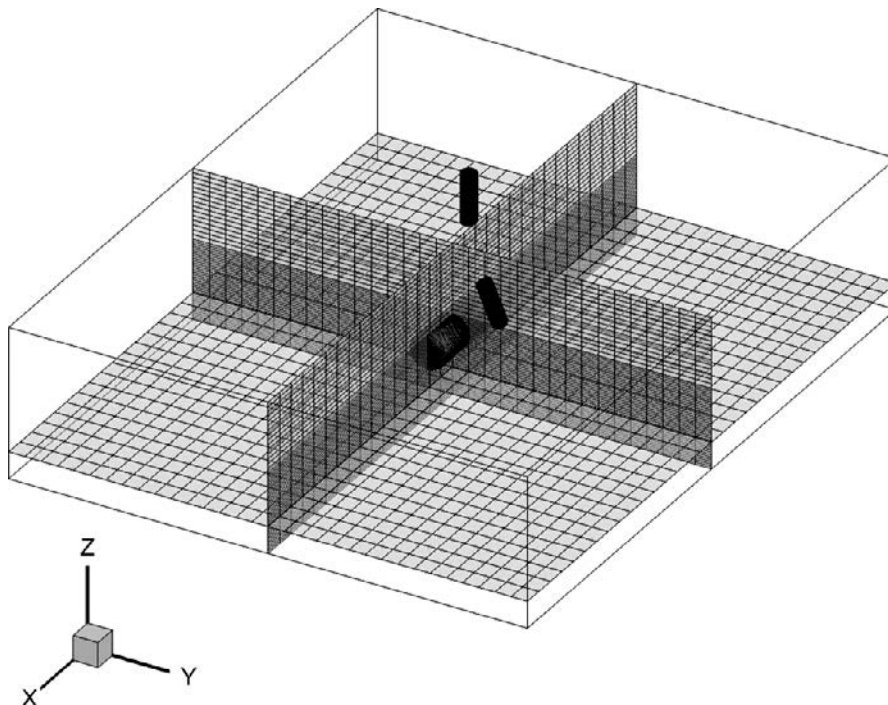


Figure 2-1. Structured Irregular Grid Example. Variable refinement is used in Z direction. This reference example has been included in the installation package of iDP V3.2.

2.2 What has changed in V3.2

In version 3.2 the following enhancements have been applied.

- Addition of automated regression tests for iDP.
- iDP-GUI version for Linux Systems have been added.
- Enhancements related to iDP-DT-GUI graphical window to match display size automatically.

3 Introductory implementation details

3.1 Application architecture

The full reactive transport framework consists of the following components (see Figure 3-1 and Figure 3-2):

- DarcyTools:
Groundwater flow simulator (Svensson et al. 2010, Svensson and Ferry 2014).
- iDP (interface between DarcyTools and PFLOTRAN):
It extracts the required flow and transport information (i.e. Darcy's fluxes and kinetic porosities) from a sub-domain of a DarcyTools model and generates the PFLOTRAN inputs (i.e. pflotran.in and related hdf5 files).
- PFLOTRAN:
High Performance Computing (HPC) reactive transport simulator.
- IDP-Post:
It splits and reduces the size of the outputs of PFLOTRAN for visualization purposes.
- ParaView:
Post-processing software (www.paraview.org/).



Figure 3-1. Overall iDP framework.



Figure 3-2. Individual iDP components.

The pre-processing done by iDP consists of 2 components (Figure 3-2):

- iDP-DT: This application is written in FORTRAN and links to the DarcyTools libraries to process the binary output format of DarcyTools. DarcyTools stores its output in so called ROF files and information about the mesh and markers in xyz files. The interface converts these files into a new set of binary files, containing information of a subdomain that can be read by iDP-DT. iDP-DT (Figure 3-3) includes a graphical user interface that is used to facilitate the user interaction. The configuration file “iDP_Options.txt” specifies the subdomain to be obtained from the DarcyTools output files. It also determines which markers should be used to extract regions. These regions can be assigned different materials, initial conditions or boundary conditions in the PFLOTRAN simulation. All output is written to binary files that can be read directly by PFLOTRAN.
- iDP-PF: This application is written in Python 3 and reads the binary files produced by iDP-DT. It then generates HDF5 files that can be read in parallel by PFLOTRAN. In addition, it generates the input file for the PFLOTRAN simulation. The user selects which geochemistry, initial conditions and boundary conditions to apply to the respective model.

The complete reactive transport framework is shown in Figure 3-4.

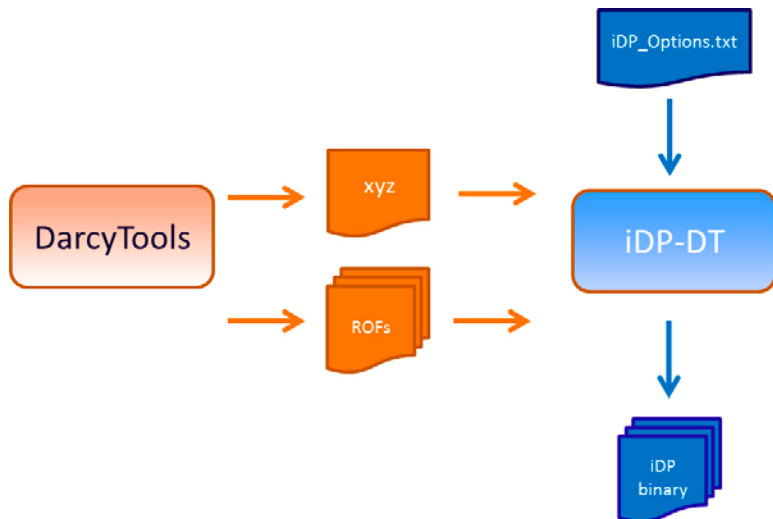


Figure 3-3. *iDP-DT overview.*

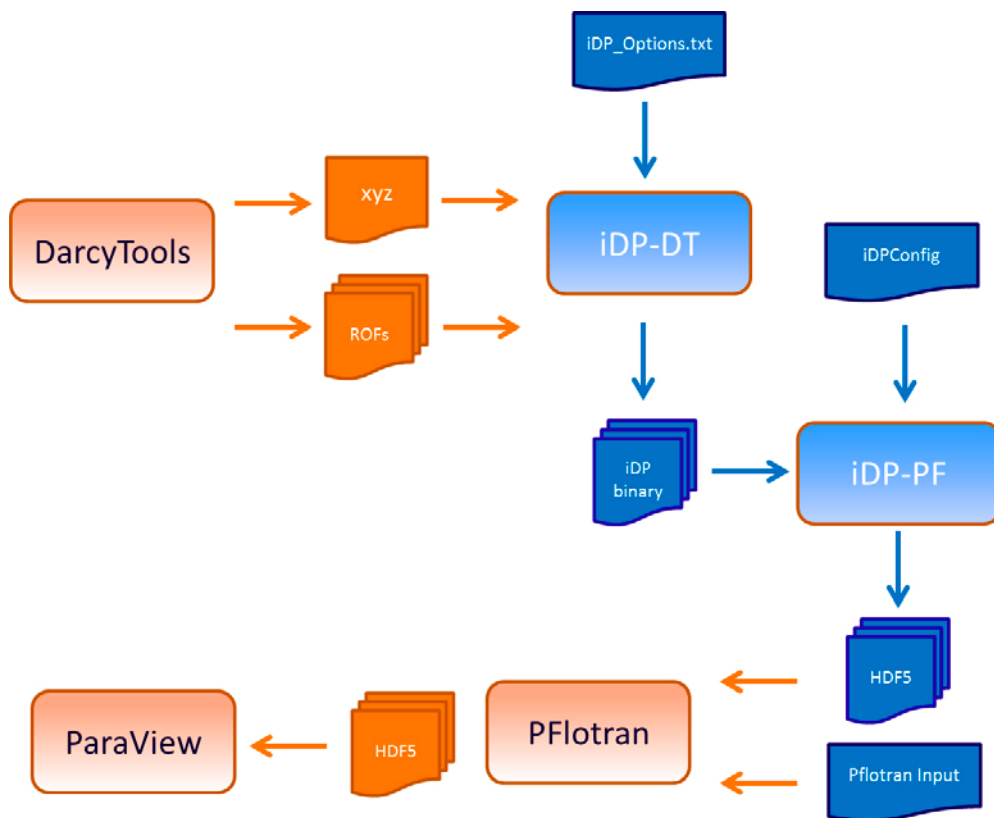


Figure 3-4. *Detailed overall iDP architecture.*

The configuration file “iDP_Options.txt” specifies the subdomain to be obtained from the DarcyTools output files. It also determines which markers should be used to extract regions. These regions can be assigned different materials, initial conditions or boundary conditions in the PFLOTRAN simulation. All output is written to binary files that can be easily read by the iDP-PF module.

The configuration file “iDPConfig.py” specifies the location of the input files, produced by iDP-DT. It also specifies all the details of the reactive transport simulation setup, like the exact domain, which geochemistry to apply and the simulation times. The complete reactive transport framework is shown in Figure 3-4.

4 Installation (Windows systems)

4.1 System requirements

Below the installation procedure for iDP installer (Windows Version) is explained. iDP does not have any specific requirements for installation. It can be installed on any 64-bit Windows by double clicking the provided setup package. During the setup procedures the location where to install iDP can be selected. The installation consists of 2 parts:

- iDP-DT
- iDP-PF

Each of these parts includes a setup package and needs to be installed separately according to the instructions. The installation procedure for both iDP-DT and iDP-PF are quite the same with only one additional installation requirement for iDP-DT.

4.1.1 Where to get iDP

The source code for iDP is kept on Bitbucket cloud repositories and is version controlled using Mercurial version control tool. Access to the repository is given to users by request. In addition, a web-based version of the iDP user manual is kept on bitbucket. In Table 4-1, the links to the three main iDP repositories is given.

Table 4-1. iDP repositories.

Repository name	Address
iDP-DT	https://bitbucket.org/amphos21_src2/idp_dt/src/default/
iDP-PF	https://bitbucket.org/amphos21_src2/idp_pf/src/default/
idp_doc	https://amphos21_src2@bitbucket.org/amphos21_src2/idp_doc

4.1.2 Installing iDP-DT

1. Login as Administrator or as a user with administrative permissions.
2. Double the setup.exe file. The installation will now start (Figure 4-1).

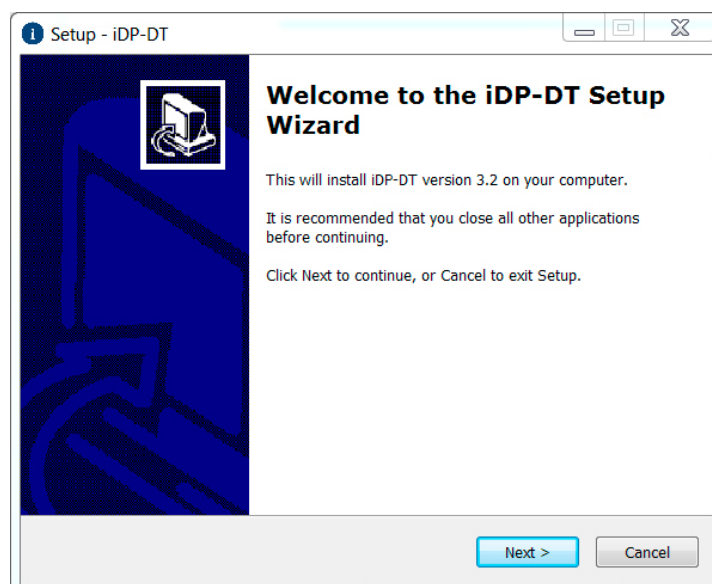


Figure 4-1. Setup wizard screenshot (1).

3. Select the Destination folder where you want iDP-DT to be installed and click *Next* (Figure 4-2).
The default installation folder is *C:\Program Files\iMaGe* where both iDP-DT and iDP-PF will be installed.

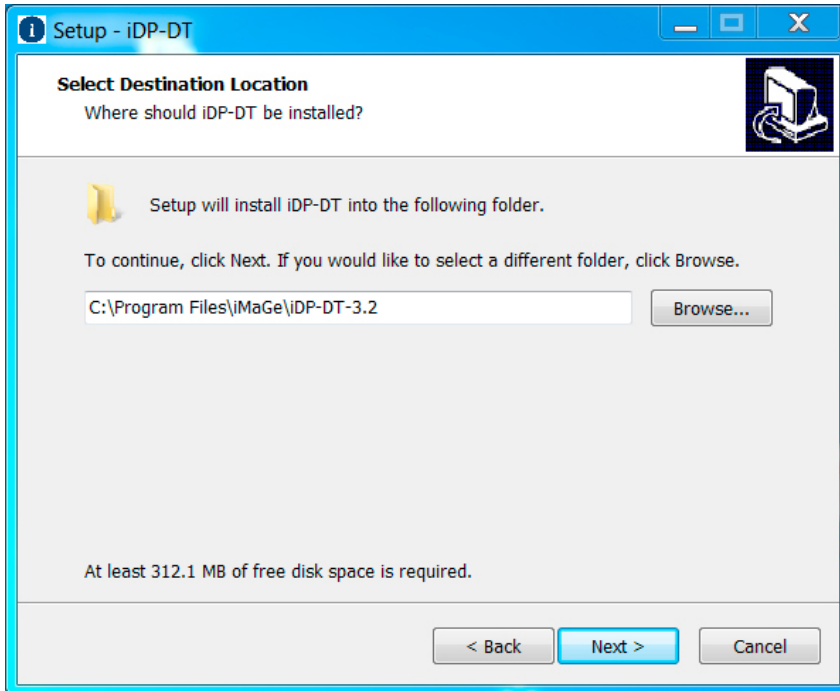


Figure 4-2. Setup wizard screenshot (2).

4. Select any additional settings such as if a Desktop Icon is desired (Figure 4-3).

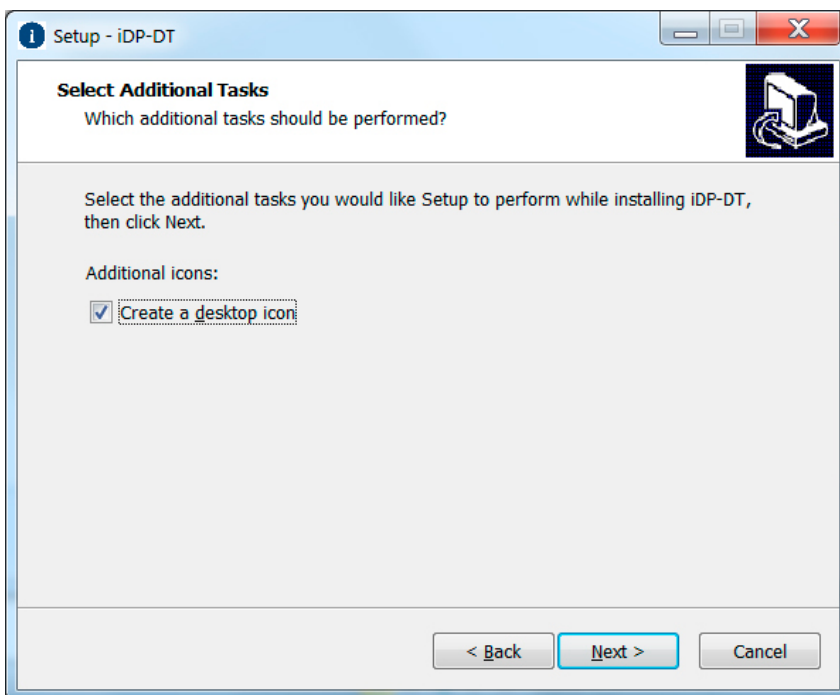


Figure 4-3. Setup wizard screenshot (3).

5. Accept your previously set configurations and click Install. This will install iDP-DT on your Machine (Figure 4-4).

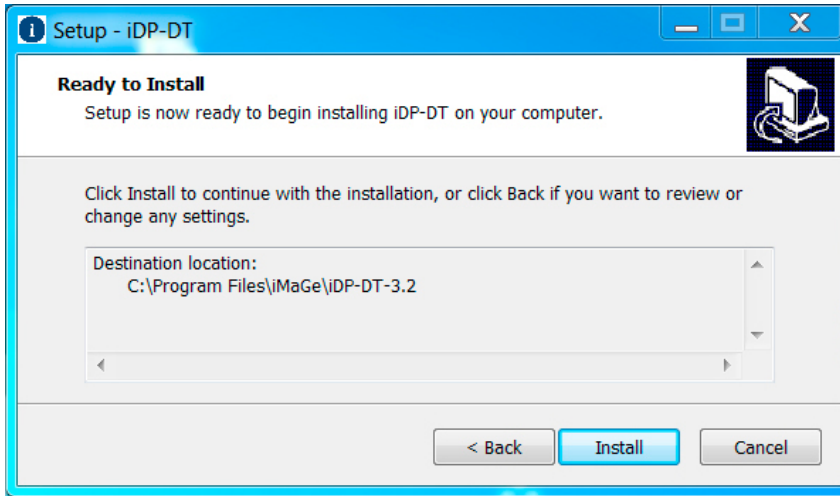


Figure 4-4. Setup wizard screenshot (4).

Installing redistributable libraries for intel visual FORTRAN compiler

The only installation requirement for iDP-DT is the redistributable libraries for Intel visual FORTRAN compiler. The user will need to install these additional libraries if they are not already installed on the destination machine where iDP-DT will be installed. The redistributable packages are for end users who use applications that are built with Intel compilers.

The redistributable libraries are included as a self-installable package that can be easily installed by double clicking on the installer. They are packaged along with iDP installers and can be found in the same installation package.

4.1.3 Installing iDP-PF

6. Login as Administrator or as a user with administrative permissions.
7. Double the setup.exe file. The installation will now start (Figure 4-6).



Figure 4-5. Setup wizard screenshot (1).

8. Select the Destination folder where you want iDP-PF to be installed and click *Next*.

The default installation folder is *C:\Program Files\iMaGe* where both iDP-DT and iDP-PF will be installed (Figure 4-6).

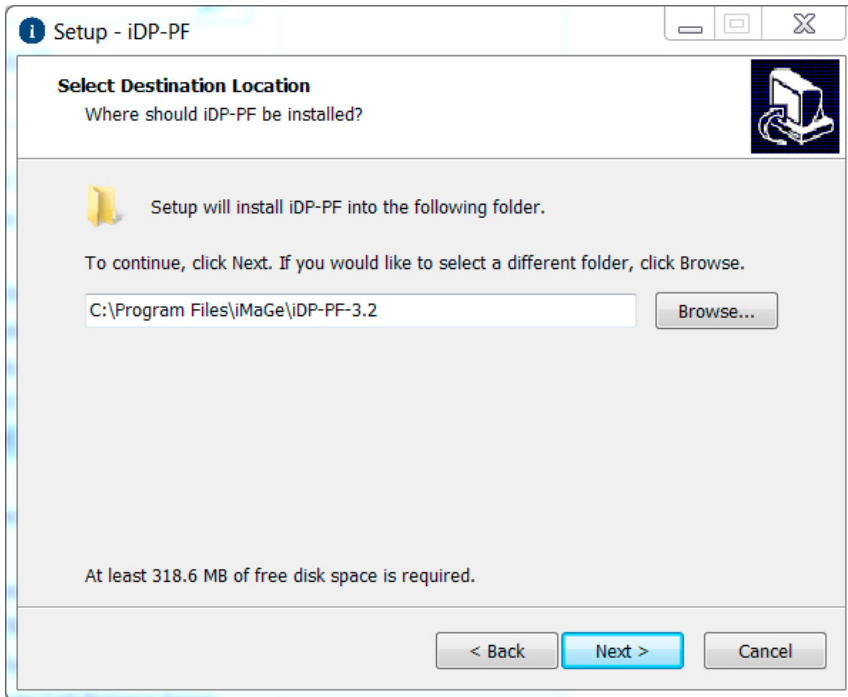


Figure 4-6. Setup wizard screenshot (2).

9. Accept your previously set configurations and click *Install*. This will install iDP-PF on your Machine (Figure 4-7).

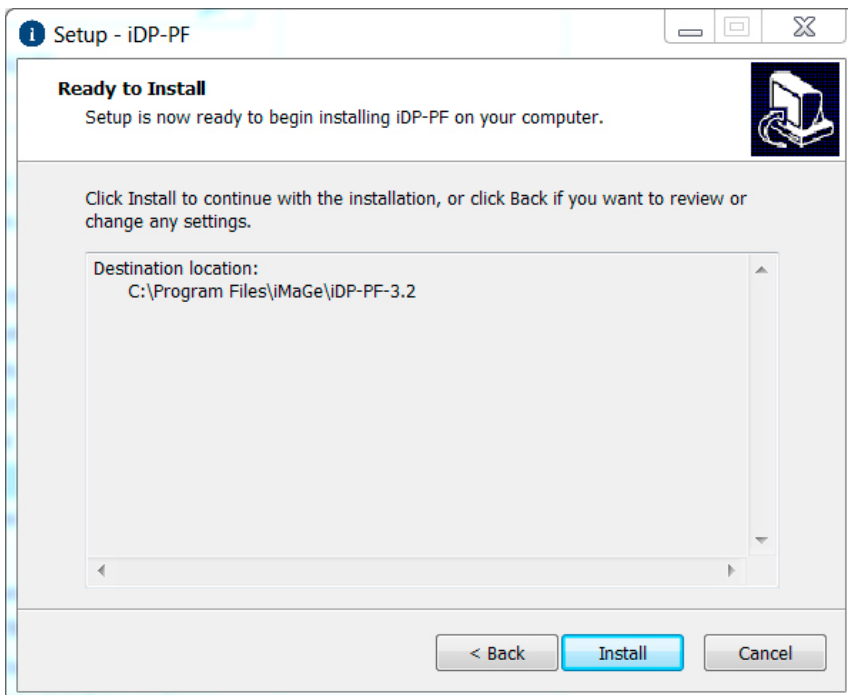


Figure 4-7. Setup wizard screenshot (4).

4.2 Uninstall

1. Go to **Add/Remove Programs** (typically under **Settings/Control Panel**).
2. Scroll down in the list of currently installed programs and select iDP-DT or iDP-PF.
3. Click the **Uninstall** button. Click **Yes** on the message box asking if you are sure you want to remove iDP-DT/ iDP-PF from your computer.
4. The removal procedure does normally leave some folder with the same name in “C:\Program Files” which might need to be removed manually afterwards.

5 iDP-DT functionality specification

5.1 Overview

iDP-DT contains the following functionalities:

- Process configuration file
- Generate PFLOTTRAN subdomain
- Process DarcyTools binary files
- Generate Absolute Cell Balance Errors
- Process DarcyTools markers
- Process grouting cells

The notation used in iDP-DT and iDP-PF is specified and briefly described in Table 5-1. Notice that this notation is consistent with the notation used in DarcyTools and PFLOTTRAN.

Table 5-1. Notation used in iDP-DT and iDP-PF.

Name	Description
Subdomain	It is a subset of the DarcyTools domain that is extracted and used in PFLOTTRAN.
ID	It is an integer value used to tag and identify e.g. grid cells, boundary conditions, flow velocity fields, etc. It is used in both iDP-DT and iDP-PF.
Marker	It is an ID number used in DarcyTools to tag and group cells belonging to e.g. specific parts of the domain (e.g. tunnels, shafts, grouted cells, etc).
Region	It is a region (e.g. point, surface, volume) within the PFLOTTRAN problem domain.
Material	It is a group of physical parameters (permeability, porosity, tortuosity, etc) that is used in PFLOTTRAN to apply properties to given regions.

5.2 Process configuration file

The configuration of iDP Fortran is read from an input file. Note that the order of the lines is important. Lines that start with a ‘!’ symbol are considered comment lines and are ignored. Some of the input data can be read from the variables are described in Table 5-2:

Table 5-2. Format of the iDP_DT configuration text file.

Variable name	Description
Output Folder	Location where the binary output files will be written.
Number of velocity fields	Number of different velocity fields provided by DarcyTools. Each component (x, y or z) can be read from a different DT binary output file (ROF).
Full path to x velocity File	ROF file for the x component of the darcy velocities (m/s).
Full path to y velocity File	ROF file for the y component of the darcy velocities (m/s).
Full path to z velocity File	ROF file for the z component of the darcy velocities (m/s).
Velocity ROF Time Step	The time step containing the velocities to be extracted.
PFLOTTRAN Velocity ID	The ID that identifies the velocity field within PFLOTTRAN.
Permeability-x ROF File	ROF file for the x component of the permeability field (m ²).
Permeability-y ROF File	ROF file for the y component of the permeability field (m ²).
Permeability-z ROF File	ROF file for the z component of the permeability field (m ²).
Permeability ROF Time Step	The time step from which to extract the permeabilities.
xyz mesh file	DarcyTools mesh/grid file. This file contains information about the mesh, connectivity and the location of the markers.

Variable name	Description
PFLOTTRAN sub-domain	These 6 lines define the sub-domain to be extracted from the DarcyTools simulation domain.
minx	
minY	
minZ	
maxX	
maxY	
maxZ	
Number of markers	For each marker the following information is read:
Marker start number	Start of marker range to consider (inclusive).
Marker end number	End of marker range to consider (inclusive).
New Marker ID	All the cells marked in DarcyTools that are within the range are assigned this new ID.
New marker ID for grout	All the cells that are within the range that have the grout filter applied, will get this new ID.
Grout filter value	If the tunnel side has a permeability larger than this threshold value, grouting will be applied, and the cell will be assigned the new marker ID for grouting.

5.3 Generate PFLOTTRAN subdomain

The coordinate limits given in the configuration file are used to extract information about the new subdomain from the DarcyTools mesh file (normally named “xyz”). The assumption is that the volume is located in an area where DarcyTools uses a structured (regular or irregular) grid. The same grid will be constructed for use in PFLOTTRAN. In this way the velocity and permeability field values can be used directly without interpolation in PFLOTTRAN.

The information extracted about the subdomain is written as one record to a binary file called (grid-Dimensions.bin) using the format specified in Table 5-3. The user does not need to concern about the contents of this file as it is an intermediate file generated by iDP. This file will be automatically used by iDP-PF to read the grid information. The format of the data is Real*4 by default in V3.0. Note that real values of 8 bytes can be used by recompiling iDP-DT against the DarcyTools R8 libraries, instead of the R4 libraries.

Table 5-3. Binary format for grid information file.

Variable name	Description	Format
Nx	Number of cells in x direction.	Integer*4
Ny	Number of cells in y direction.	Integer*4
Nz	Number of cells in z direction.	Integer*4
minX	Lower x coordinate of the subdomain.	Real*4
minY	Lower y coordinate of the subdomain.	Real*4
minZ	Lower z coordinate of the subdomain.	Real*4
maxX	Higher x coordinate of the subdomain.	Real*4
maxY	Higher y coordinate of the subdomain.	Real*4
maxZ	Higher z coordinate of the subdomain.	Real*4
nrefX	Number of refinements in X direction.	Integer*4
nrefY	Number of refinements in Y direction.	Integer*4
nrefZ	Number of refinements in Z direction.	Integer*4
numOfCells@deltaX	Number of cells with dx length in x direction; this record will be repeated nrefx times.	Integer*4
numOfCells@deltaY	Number of cells with dy length in y direction; this record will be repeated nrefy times.	Integer*4
numOfCells@deltaZ	Number of cells with dz length in z direction; this record will be repeated nrefz times.	Integer*4

5.4 Process DarcyTools binary files

DarcyTools stores the grid in an XYZ file and the variable values during the DarcyTools Simulation in a binary ROF file. ROF stands for “Result Output Files” and is used to store variable values during the DarcyTools simulation. For a detailed description of the ROF file format please refer to DarcyTools User’s Manual (Svensson et al. 2010). iDP-DT processes the DarcyTools binary files (“ROF”) to obtain data sets (velocities, permeability and porosity) within the PFLOTRAN subdomain. All the obtained datasets are written as binary files that can be read in effortlessly from the Python NumPy library.

5.5 Generate absolute cell balance errors

To ensure that the DarcyTools flow simulation converged sufficiently to provide a velocity field that can be used to perform reactive transport simulations, the absolute balance errors (cell mass balance residual) for each cell is written to data set.

5.6 Process DarcyTools markers

In the DarcyTools input file (“cif.xml”) cells can be assigned a unique marker ID. Often a range of markers are used to group a specific type of objects (e.g. markers 3 to 21 are access tunnels). iDP provides the possibility to select such a range and assign it to one PFLOTRAN region with a specific material type.

Neighbouring cells of marker cells are also identified and properly tagged. This allows, for instance, marker cells to be deactivated and neighbouring cells to be used to apply boundary conditions.

5.7 Process grouting cells

Since the marker IDs are often used to indicate tunnel walls, iDP implements an extra filter to apply an upper permeability threshold. If the permeability of a specific marker ID cell side is higher than the given threshold, the cell is marked with a new marker ID, a given grouting ID.

6 iDP-PF functionality specifications

6.1 Overview

The iDP-PF application contains the following functionalities:

- Process configuration file
- Create layered regions
- Convert random fields to regions
- Marker functionality
- Material assignment
- Process velocities
- Process porosity and permeability
- Process mass balance errors
- Generate PFLOTRAN input file

iDP-PF generates the HDF5 files required to run PFLOTRAN. To facilitate the use of these HDF5 files, an example input file for PFLOTRAN is also created. The generated PFLOTRAN file should always be reviewed (and in most cases modified) manually before running the simulation. In addition, since PFLOTRAN is under constant development, changes might have been introduced in PFLOTRAN input file format since the last iDP release. In this case, the user might need to manually refactor the input file.

6.2 Process configuration file

The configuration of iDP-PF is read from an input file. The input file contains a python data structure, called dictionary, to structure its contents. A dictionary can contain several keys, where each key will have a value. In addition, the values can be dictionaries as well. The definition of a key is case-sensitive. The main parts in the configuration file are described in Table 6-1.

Table 6-1. Main items defined in the configuration file.

Variable name	Description
logFile	The full path to and name of the log file. Note that instead of “\”, the “/” needs to be used.
simulations	This dictionary defines a sequence of PFLOTRAN simulations to generate. For each simulation a PFLOTRAN input file will be generated. <ul style="list-style-type: none">• simulationID: the ID defined for a velocity field in the iDP-DT configuration.• simulationTime: total time to simulate.• wallclockStop: the simulation will be stopped after this time, even if it is not finished.• Restart (optional): A list with the filename of the checkpoint file and the time to restart from. The time is given to prevent PFLOTRAN from starting from time = 0.0.
pflotranSettings	This dictionary defines the setup of the simulation in PFLOTRAN. Among the most important are: <ul style="list-style-type: none">• Time steps.• Solver settings.• Chemistry Setup.• Material Properties.
RealType	Specifies if the binary files are written in Real*4 or Real*8 format.
FileNames	The output file names for the HDF5 files and the PFLOTRAN configuration file.
regions	The subdomain can be divided into regions subdomains defined by depth. In this section the different regions and their properties are specified.

6.3 PFLOTTRAN settings

The PFLOTTRAN settings section deals with the several mandatory keywords. Most values will simply write the value mentioned directly to the PFLOTTRAN config file. For more details on the keywords, check the PFLOTTRAN manual (Lichtner et al. 2015). What follows is the list of keywords:

- **simulationTitle:** the title to use for this simulation.
- **simulationInputPath:** The path where the DarcyTools output files are located. The relative path of PFLOTTRAN Configuration file (pflotran_gen.in) to the simulation hdf5 input files. Make sure that only forward slashes, '/', are used instead of backward slashes.
- **timeSteps:** Contains all time step size related commands, like initial time step size, maximum time step size.
- **outputTimeUnit:** Unit of output time. This unit will be used in all outputs of PFLOTTRAN, e.g. if selected 'y' for year, the Darcy velocity will be written as m/y.
- **outputTimes:** This is a list that contains information about one or more time segments. Each segment is defined by a list. The first item in the list specifies the type of the segment:
 - **prescribed:** To be used to explicitly set the output times.
 - The second item is a list of times.
 - The third item in the list specifies the format the numbers are written to. ('g' is the normal default. Check the python manual for advanced formats).
 - Example: ["Prescribed", [1, 2, 3, 4, 5, 7.5, 10], "g"].
 - **Linear:** To use a regular range of times.
 - The second item is a list with minimum and maximum time.
 - The third item is the number of times to output.
 - The fourth item in the list specifies the format the numbers are written to.
 - The fifth item indicates if the upper limit is included.
 - Example: ["Linear", [1, 10], 10, "g", True].
 - **Log:** To use a logarithmic time output scale.
 - The second item is a list with minimum and maximum time.
 - The third item is the base to use for the logarithm.
 - The fourth item is the number of times to output.
 - The fifth item in the list specifies the format the numbers are written to.
 - The sixth item indicates if the upper limit is included.
 - Example: ["Log", [1e-10, 1], 10, 5, "g", True].
- **Checkpoint:** A checkpoint file is written every X time steps. This option specifies the value X.
- **hdf5ReadGroup:** Default value to use is "None". Only if PFLOTTRAN is compiled with the SCORPIO library, this value needs to be set.
- **hdf5WriteGroup:** Default value to use is "None". Only if PFLOTTRAN is compiled with the SCORPIO library, this value needs to be set.
- **Materials:** This section defines all the materials to be used in PFLOTTRAN (Table 6-2).

Table 6-2. Material configuration.

Variable name	Description
ID	Unique ID for material. Note that the IDs start with 1 and should be consecutive (e.g. do not define 2 materials with ID 1 and 3, but use 1 and 2 instead).
Tortuosity	Tortuosity to be used in the simulation.
Permx	Permeability to be used in the simulation.
Permy	Permeability to be used in the simulation.
Permz	Permeability to be used in the simulation.

- **totalChemistry:** A string containing the whole PFLOTTRAN CHEMISTRY section. This section defines, among other things, the primary and secondary species and the minerals.
- **chemistryTypes:** This string defines the constraints and the transport conditions (which are also used in PFLOTTRAN to define initial conditions).
- **solverConfig:** This string enables setting special transport solver flags.

6.4 File name settings

The “fileNames” section specifies which filenames to use for the output files of iDP-PF. These files will be written to the output folder specified when running iDP-PF. The velocity file name consists of a *velocityBaseName* and a *velocityExtName*. These two parts will be joined with an integer in between, to generate an output file for each velocity field. The *generatedRegions* file is used to specify a file containing random field indicators (generated by SGeMS, in the GSLIB format). If no random field is provided, this value should be set to *None*.

6.5 Handling of regions

The configuration for each region contains the values specified in Table 6-3.

Table 6-3. Region configuration.

Variable name	Description
ID	The name of the region.
markerID	If defined, the region contains all the cells marked by this ID.
Side	If the region is defined in terms of sides, this values defines which side (1–6). If the region is defined by cells this is set to 0.
materialID	The ID of the material assigned to this region.
isExternalRegion	If the region defines the box, this value is set to “True”. If it defines an internal region (like a marker ID region), this value is set to “False”.
generatedRegion	If no random field is provided, this value should be set to “None”. Otherwise the field should be set to the indicator (an integer value) to use for this region. Note that each region can be related to a unique material.
minDepth, maxDepth	If different geochemistries are assigned to different depths for external regions, these values define the depth range. If set to “None”, the whole subdomain is treated as a single domain. In case of an internal region, all cells outside this range will be ignored.
Geochemical initial conditions	The geochemical initial conditions for the reactive transport simulation in PFLOTTRAN. This name should correspond to a TRANSPORT_CONDITION in PFLOTTRAN.
Geochemical Boundary conditions	A list of 6 TRANSPORT_CONDITION IDs, one for each side, representing the boundary condition for this region in PFLOTTRAN. The order is as follows: [WestBC, EastBC, SouthBC, NorthBC, BottomBC, TopBC]. Each of these tags should refer to a corresponding TRANSPORT_CONDITION.
writeOrder	The order in which the regions and the initial conditions are written to the PFLOTTRAN configuration file. If cells belong to two different regions (e.g. domain cells and marker cells), the one with the last write order is used.

Assigning material ID and write order to regions:

If the user intends to define more than one region in Z or intends to have additional grout and Tunnel regions, then the user will need to assign write order to these regions in the configuration file (iDPConfig.py).

The write order by default is set to 1. User will need to assign a unique writeOrder to each of these regions in the following order starting from 1 and incrementing by 1 eg. 1, 2, 3, 4, 5....

Priority Rule: Priority increases as the writeOrder number increases. A region with a higher priority (higher writeOrder) will over write other region's material IDs if there exist common cells between the regions. For example, in case some cells belong to 5 regions; the region with write order 5 will overwrite the materialID value for cells common with regions with write order 1, 2, 3 and 4.

Different regions must not have the same write order. If two regions have been set the same write order, then iDP_PF will run successfully but will contain logic error in region definition.

Two regions can have the same Material IDs. (Useful when assigning different boundary conditions). It is important for the user to understand the logic behind the Material ID and write Order assignment to prevent Logic errors. In Table 6-4 and Table 6-5 you can find some correct and wrong examples of Material ID and write Order assignment.

Table 6-4. Wrong assignment of write orders to regions.

Variable name	Write order	Material ID
Region 1	Write Order = 1	Material ID = 1
Region 2	Write Order = 1	Material ID = 2
Region 3	Write Order = 2	Material ID = 3
Region 4	Write Order = 3	Material ID = 4

Table 6-5 Correct assignment of write orders to regions.

Example 1:		
Variable name	Write order	Material ID
Region 1	Write Order = 1	Material ID = 1
Region 2	Write Order= 2	Material ID = 2
Region 3	Write Order = 3	Material ID = 3
Region 4	Write Order = 4	Material ID = 4
Example 2:		
Variable name	Write order	Material ID
Region 1	Write Order = 1	Material ID = 1
Region 2	Write Order = 2	Material ID = 1
Region 3	Write Order = 3	Material ID = 2
Region 4	Write Order = 4	Material ID = 2

6.6 Marker functionality

If a region is defined by a marker ID, the cell IDs are read from the binary file:

“markedCells_*markerID*_CellIDs.bin”. If there a boundary condition is defined for these cells, the cells and their side are read from “markedCells_*MarkerID*_BorderCellIDs.bin” and “markedCells_*MarkerID*_BorderCellSides.bin”. Each side is read in as a separate Region (region name with the extension: “_west”, “_east”, “_north”, “_south”, “_top”, “_bottom”). In this way different boundary conditions can be assigned to each side.

6.7 Process velocities

For PFLOTTRAN to be able to import velocities, the velocity HDF5 file created by iDP-PF should contain three arrays (for the three velocity components) that hold values for each cell (upwind direction). For each boundary condition another full array with velocity values is required. This means that most values will be zero, only the boundary cells will have a non-zero value.

6.8 Process porosity and permeability

The porosity hdf5 file created by iDP-OF contains a value for each cell.

The permeability file is only written for post-processing purposes, since the velocity field is already calculated.

6.9 Process mass balance errors

The field of absolute mass balance errors is written in PFLOTTRAN HDF5 output format so that it can be opened and analysed directly in ParaView.

6.10 Generate PFLOTTRAN input file

iDP-PF generates an ASCII PFLOTTRAN input file that automatically links to all the HDF5 files that were generated by iDP-PF. In addition, it will apply the correct initial and boundary conditions to the regions specified in the iDP-PF configuration file. All PFLOTTRAN settings can be modified through the iDP configuration file. Note that in most cases the generated ASCII file needs to be modified slightly before running the actual simulation. This could be due to PFLOTTRAN version being more recent than the iDP release in case changes regarding the ASCII input file format are implemented in PFLOTTRAN.

7 Running iDP

7.1 Introduction

This chapter will explain how iDP is run and will look at details such as the structure of the configuration file, the GUI components, the input directory and the output directory containing the input and output files for iDP.

7.2 iDP-DT

The idea of iDP-DT is to extract a subdomain from the DarcyTools output files, xyz and ROF files (number 1 and 2 in Figure 7-1). The extracted subdomain is stored in binary files that will be used as input for iDP-PF (3). iDP-PF will then create a configuration file for PFLOTRAN and several HDF5 data files referred to in the configuration file.

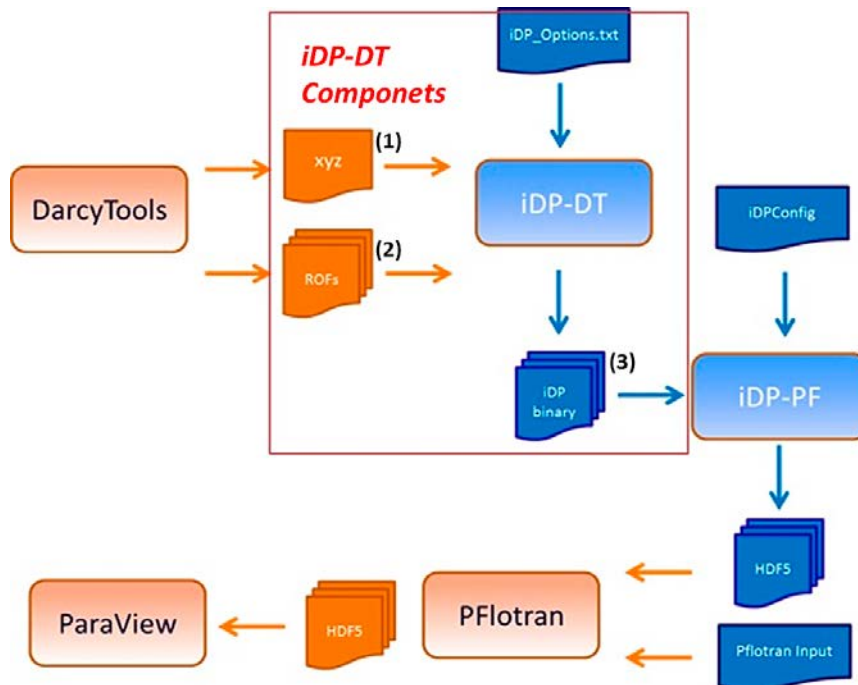


Figure 7-1. iDP-DT Components.

7.2.1 iDP-DT User interface components

After installation of iDP-DT, it can be started from the Windows Start Menu. The graphical interface (Figure 7-2) makes it easier to generate the configuration required for running iDP-DT.

In Figure 7-2 the different components of iDP-DT graphical user interface have been numbered. These components include the following:

- DarcyTools Velocity (1)
- DarcyTools Permeability (2)
- DarcyTools Porosity (3)
- PFLOTRAN Subdomain (4)
- Markers (5)
- Output Folders (6)
- Control Buttons (7, 8, 9, 10)

In the following section, the components above will be explained in numerical order.

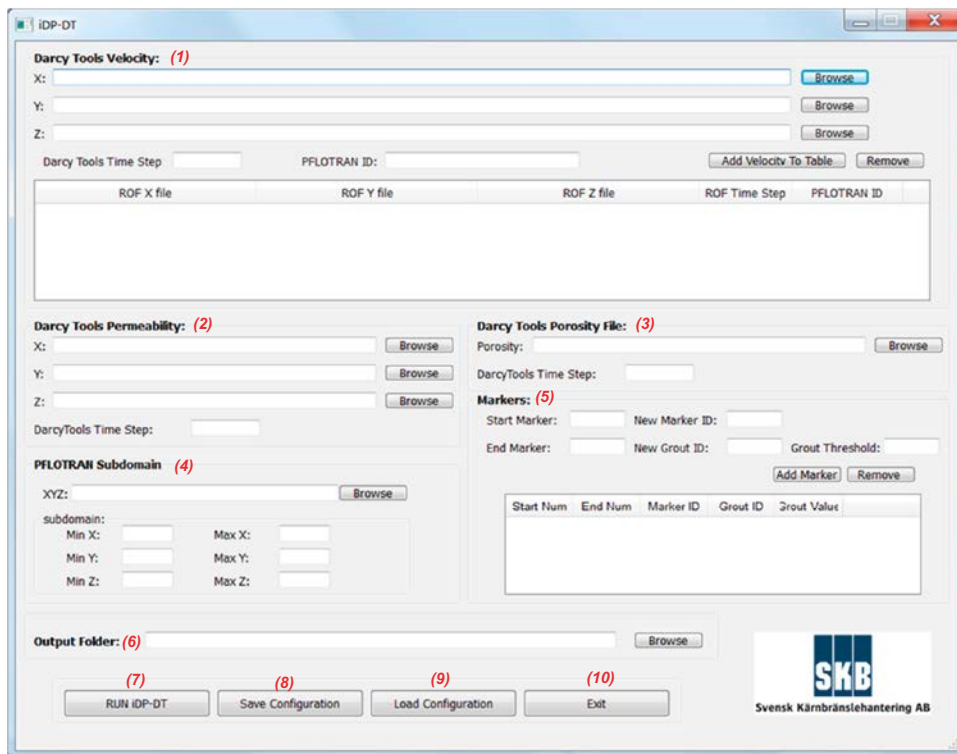


Figure 7-2. iDP-DT Graphical user interface.

DarcyTools velocities

The ROF file used to retrieve the velocity field can be set for each component (X, Y and Z).

X: address to ROF file containing the Darcy’s velocity in the x direction (Figure 7-3 – Number 1)

Y: address to ROF file containing the Darcy’s velocity in the y direction (Figure 7-3 – Number 2)

Z: address to ROF file containing the Darcy’s velocity in the z direction (Figure 7-3 – Number 3)

Note that the same ROF file can be used to store the three components of Darcy’s velocity.

DarcyTools Time Step: DarcyTools can save different time steps to a single ROF file. The time step to extract should be defined for each velocity field (Figure 7-2 – Number 4).

PFLOTRAN ID: iDP-DT allows extracting multiple velocity fields (e.g. these multiple velocity fields are used to run simulations with pseudo-transient flow conditions). To make sure that PFLOTRAN uses the correct velocity field at a specific time, an ID has to be assigned to the velocity field. This ID is an optional string and is user-defined when being set in iDP-DT. The ID string is case sensitive though and must be compatible with the ID later set in the next processing step in iDP-PF configuration file (iDPConfig.py). (Figure 7-2 – Number 5)

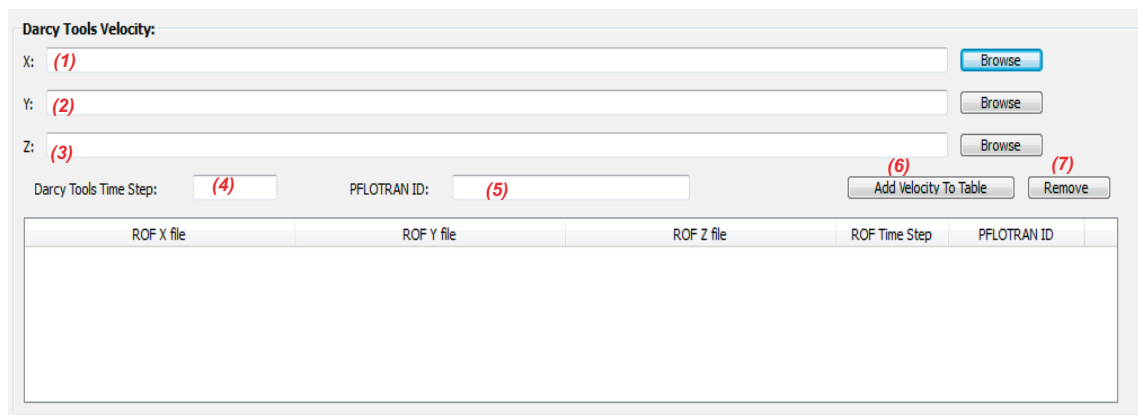


Figure 7-3. iDP-DT; setting DarcyTools velocities

If the *simulationID* set by the user in iDP-PF is not identical to the *PFLOTRAN ID* set in iDP-DT (Figure 7-4), the user will receive the following error message.

ERROR: The PFLOTRAN velocity ID “USER SET STRING” was not found.

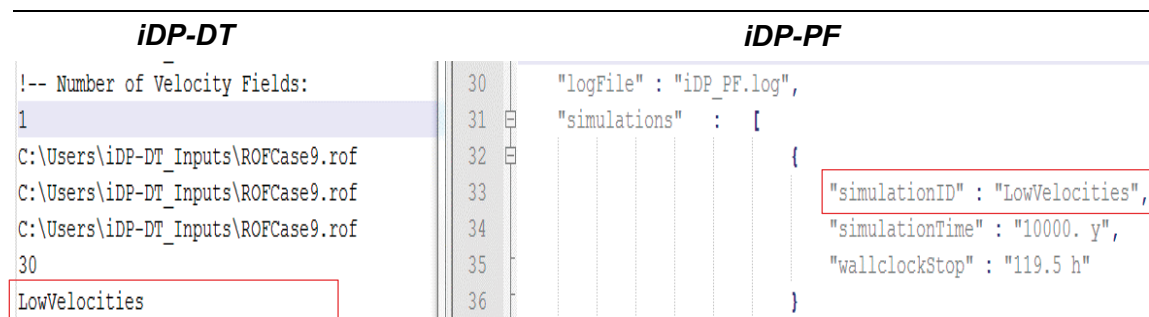


Figure 7-4. PFLOTRAN ID (iDP-DT) vs simulation ID (iDP-PF).

Add Velocity To Table Button: After all the X, Y, Z velocities, *DarcyTools Time Step* and an optional *PFLOTRAN ID* has been set; the user can add the DarcyTools velocities by clicking the *AddVelocityToTable* button. (Figure 7-3 – Number 6). This will add one row to the velocities table as can be seen from Figure 7-5.

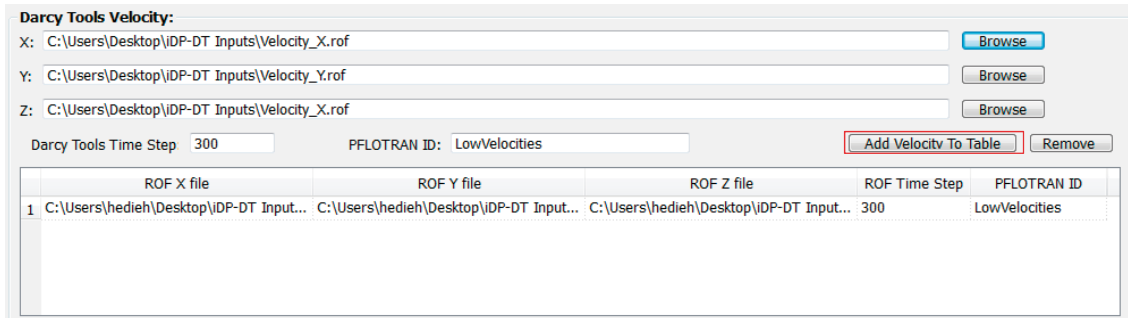


Figure 7-5. Adding velocities to table in iDP-DT.

Remove Button: The user can remove the entered velocity row by selected the desired row and removing it using the Remove button (Figure 7-6).

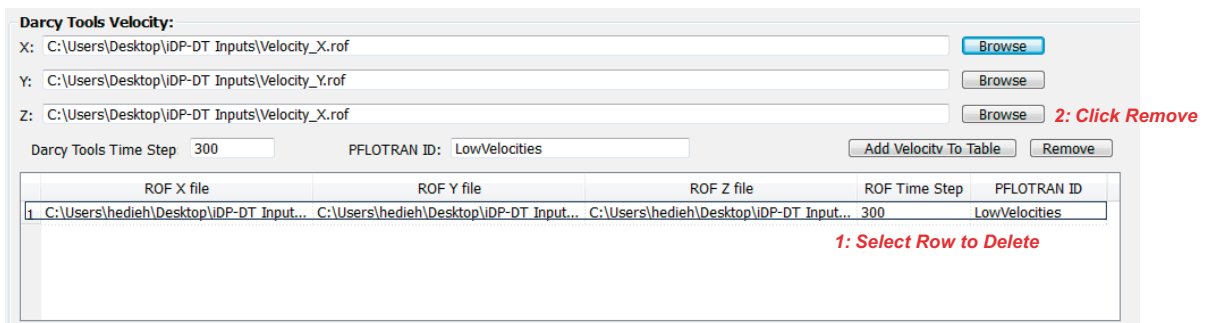


Figure 7-6. Removing velocities from the table in iDP-DT.

DarcyTools permeability

In this section, the address to the ROF file containing the permeability field output for X, Y and Z direction will be set. It is important to set the correct DarcyTools Time Step when filling in the Permeability table (Figure 7-7). Note that permeability is used for post-processing only (ground-water flow is not computed by PFLOTRAN).

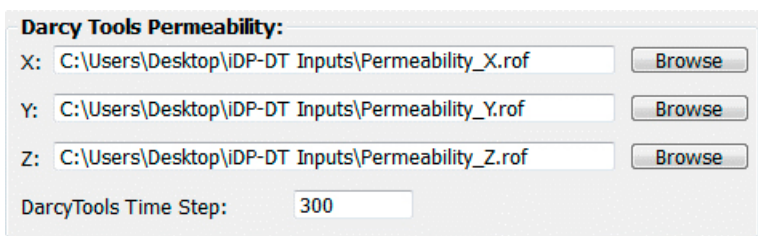
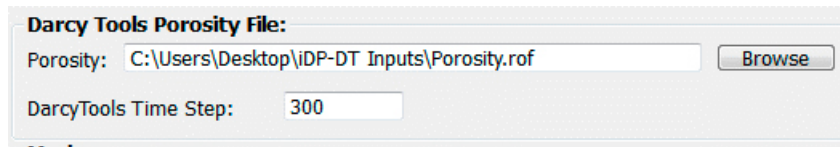


Figure 7-7. Adding permeability X, Y and Z to the table.

DarcyTools porosity

In this section the address to the ROF file containing the porosity output of DarcyTools is set (Figure 7-8).



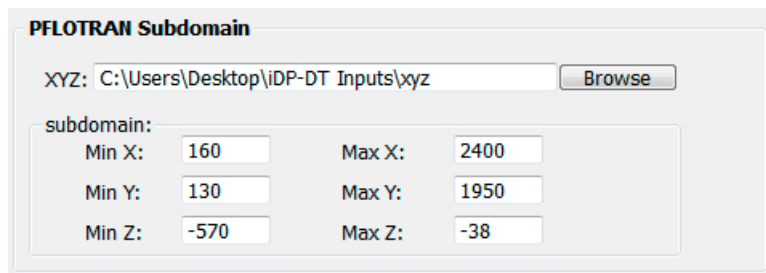
Darcy Tools Porosity File:
Porosity: C:\Users\Desktop\iDP-DT Inputs\Porosity.rof
DarcyTools Time Step: 300

Figure 7-8. Adding porosity to iDP-DT input table.

As mentioned previously it is important to set the time step to the correct time step from which the Porosity values have been retrieved from DarcyTools. This is especially important as the DarcyTools Fortran modules that are used within iDP-DT will use this time step number to read data from correspondent porosity arrays in FORTRAN.

PFLOTRAN subdomain

In this section the coordinates of the subdomain should be specified. The actual cells that are extracted are obtained from the DarcyTools xyz file (Figure 7-9).



PFLOTRAN Subdomain
XYZ: C:\Users\Desktop\iDP-DT Inputs\xyz
subdomain:
Min X: 160 Max X: 2400
Min Y: 130 Max Y: 1950
Min Z: -570 Max Z: -38

Figure 7-9. Setting the PFLOTRAN subdomain to extract from DarcyTools domain.

Markers

In DarcyTools, when defining different zones such as whole domain and/or tunnel, markers are used. A marker is an argument integer value which marks the cells of that zone with a specific id. These markers can be further processed and filtered for PFLOTRAN using iDP-DT.

A range of DarcyTools markers can be treated as a single ID in PFLOTRAN. To define the range, fill in the *DarcyTools Marker Start* and *DarcyTools Marker End* field and then set the new ID to be assigned to this range of markers in PFLOTRAN. For instance, in Figure 7-10, all the cells tagged by DarcyTools with markers from 1 to 5 are assigned a new marked ID (ID 2) and all these cells that have a permeability value higher than a threshold are assigned a second different ID (ID 3). These two ID's will be associated to related PFLOTRAN materials, whereas the other cells are kept with the originally assigned ID's.

If the cells with the selected marker have permeability side with a permeability larger than the given threshold, they are marked as grout cells, using the provided new grout ID. If no grout cells are required, the grout threshold can be set to a really large value (e.g. 1 000.0).

Markers information can be added using the *Add Marker* button. Also a markers row can be deleted by first selecting the row and then using the *Remove* button.

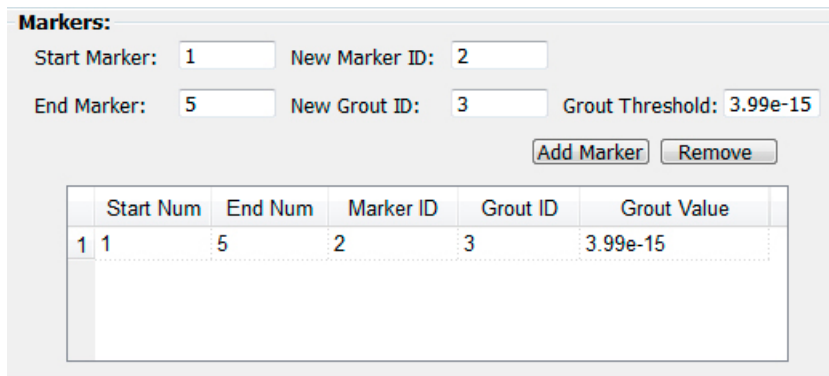


Figure 7-10. Setting new markers for PFLOTRAN from DarcyTools markers.

Moreover, if no further marker processing is needed and the user plans to set the material ID of all the cells in PFLOTRAN to default material ID = 1; user can simply do so by leaving the marker table empty or equivalently by defining the following line in the iDP-DT Configuration file (iDP_Options.txt).

!--Number Of different markers:
0

In other words, the Markers table is the only optional table in the iDP-DT GUI and the user can choose to define markers based on the requirements of the model under study.

Output directory

Before running iDP-DT, the output directory where the iDP-DT output results will be written needs to be selected (Figure 7-11).

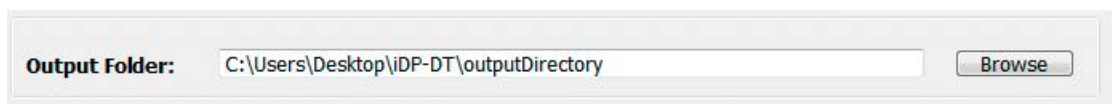


Figure 7-11. Setting the output directory in iDP-DT.

Run iDP-DT button

After all the required input tables in the GUI have been filled completely, the user can run iDP-DT using the *RUN iDP-DT* button. At this step if one of the input fields in the table contains an invalid value such as an invalid DarcyTools time step input or an incorrect address to velocity, porosity or permeability ROF input files; the iDP-DT stops with an error message (Figure 7-12). A log file containing additional information will be written in the output directory set by user. The log file address can be viewed in the error message.

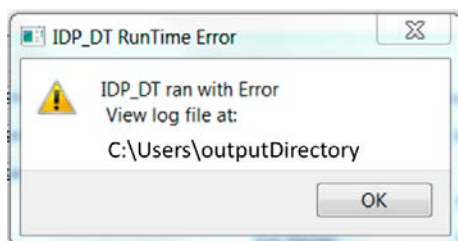


Figure 7-12. iDP-DT error message.

If all the input parameters have been set correctly, iDP-DT will run successfully (Figure 7-13) and all the output binaries and a log file will be written to the output directory set by user.

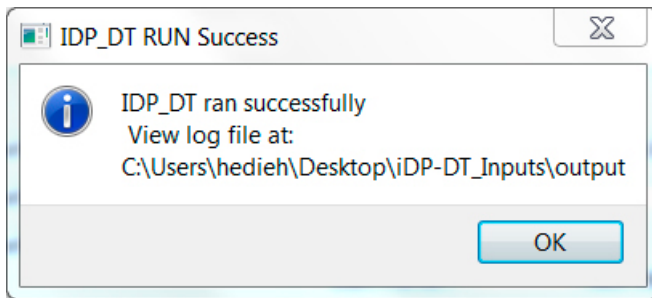


Figure 7-13. iDP-DT Success Message.

Load configuration button

The purpose of the *Load Configuration* button is to initialize the input parameters in your GUI tables from a file. To do so you can simply click on the *Load Configuration* button and browse to the correct *iDP_Options.txt* file. This will fill in the GUI input tables according to the configuration you have set in your file. We will discuss the structure of the iDP-DT configuration file later.

Exit button

Exit button is used to close the application at any time.

7.2.2 Running iDP-DT from command line

In Section 7.2.1, the iDP-DT GUI components were explained thoroughly. In Windows, iDP-DT engine is the executable *iDP-DT.exe*. This can be run either by the GUI or directly by the user from command prompt.

In order to run iDP-DT from command line you should type the following at your terminal.

```
iDP-DT.exe ....\iDP_Options.txt
```

By default, the iDP-DT engine (i.e. “*iDP-DT.exe*”) is located in the following folder.

```
C:\Program Files\iMaGe\iDP-DT-3.0\iDP_DT\bin\iDP-DT.exe
```

7.2.3 iDP-DT configuration file structure

The configuration of iDP Fortran is read from an input file. Note that the order of the lines is important. Lines that start with a ‘!’ symbol are considered comment lines and are ignored.

When running iDP-DT from GUI, a configuration file with all the input parameters you have set will be saved to the output folder. This file (*iDP_Options.txt*) has a specific structure and can be made from scratch in order to be loaded in to GUI or simply be run from the command line interface directly.

Below the structure of a sample configuration file with one velocity field and one marker definition will be explained. If the user needs to have more than one velocity field or set of markers, the sections relating to velocity field or markers need to be repeated for each additional definition of velocity or markers.

When creating the configuration file; each section must meet the constraints of the configuration format. In addition, the following order of sections must be met:

1. Output Directory:

```
1  !-- iDP-DT Output Folder:  
2  C:\Users\outputDirectory
```

2. Velocity Fields: If you have more than one velocity field the lines 5 to 9 need to be repeated for each velocity Field. Line 8 includes the DarcyTools time step at which the velocity values need to be extracted. Line 9 is the PFLOTRAN id that the user assigns to this velocity field and will be later read by PFLOTRAN.

```
3  !-- Number of Velocity Fields:  
4  1  
5  C:\Users\iDP-DT_Inputs\Velocity_X.rof  
6  C:\Users\iDP-DT_Inputs\Velocity_Y.rof  
7  C:\Users\iDP-DT_Inputs\Velocity_Z.rof  
8  30  
9  LowVelocities
```

3. Porosity: At line 11, the porosity file will be read. Line 13 includes the DarcyTools time step at which the velocity values need to be extracted.

```
10 !--ROF file containing the porosity:  
11 C:\Users\iDP-DT_Inputs\Porosity.rof  
12 !--ROF file time step  
13 30
```

4. Permeability Field:

```
14 !--ROF files with the permeability field:  
15 C:\Users\iDP-DT_Inputs\Permeability_X.rof  
16 C:\Users\iDP-DT_Inputs\Permeability_Y.rof  
17 C:\Users\iDP-DT_Inputs\Permeability_Z.rof  
18 !--ROF file time step  
19 30
```

5. DarcyTools Mesh:

```
20 !--File name of the Darcy Tools mesh:  
21 C:\Users\iDP-DT_Inputs\xyz
```

6. PFLOTRAN subdomain limits:

```
22 !--PFlotran sub-domain limits: minX, minY, minZ, maxX, maxY, maxZ:  
23 160.0  
24 130.0  
25 -38.0  
26 2400.0  
27 1950.0  
28 -570.0
```

7. Marker definitions:

```
29 !--Number Of different markers:  
30 1  
31 !--Marker start and end number:  
32 1  
33 5  
34 !--New Marker ID:  
35 2  
36 !--New Marker ID and grout threshold for grout filter:  
37 3  
38 3.99E-15
```

7.3 iDP-PF

Using iDP-DT, a subdomain from DarcyTools is extracted and stored in binary files (number 1 in Figure 7-14). This binary files (*.bin) will be used as input for iDP-PF. iDP-PF will then create an input file for PFLOTRAN (2) and several HDF5 data files referred to in the configuration file (3). In the following chapter, the iDP-PF user guide is explained in detail.

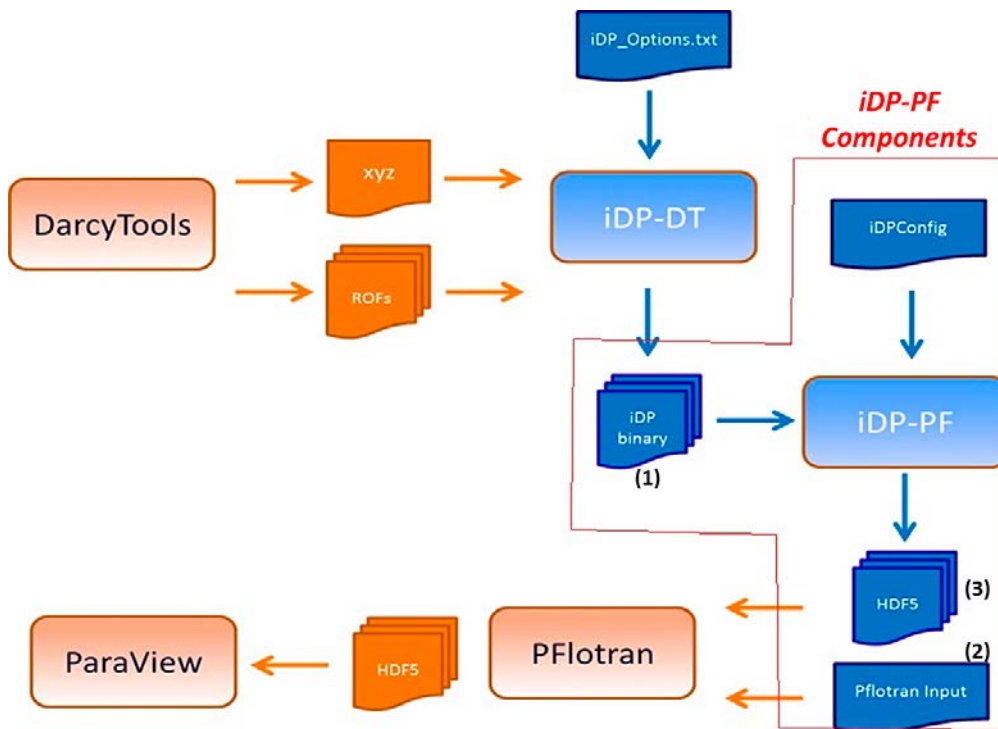


Figure 7-14. iDP-PF Components.

7.3.1 iDP-PF User interface components

After installation of iDP-PF, it can be started from the Windows Start Menu. The graphical interface (Figure 7-15) is simpler than the iDP-DT GUI. In iDP-PF the complexity is more related to setting the configuration files correctly.

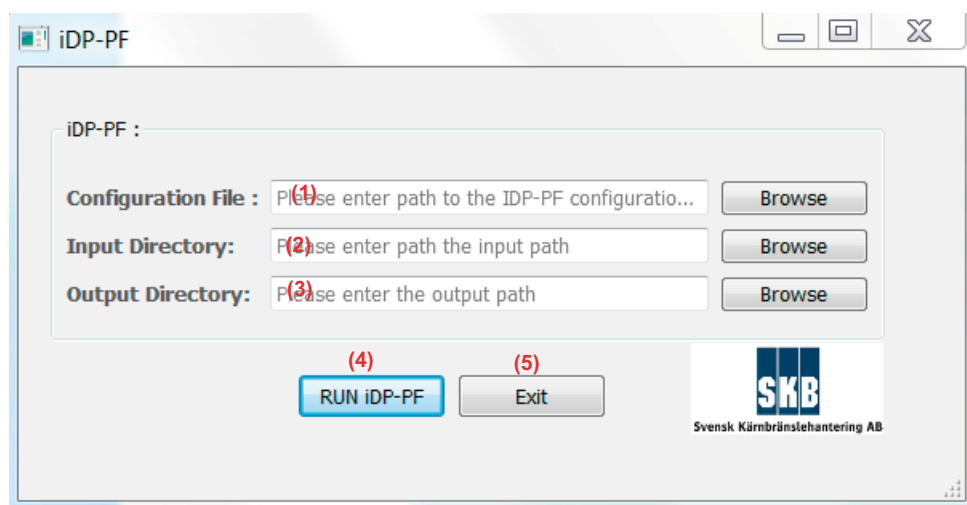


Figure 7-15. iDP-PF Graphical user interface.

Configuration file

Here the user has to specify the directory containing the iDP-PF configuration file (`iDPConfig.py`). iDP-PF Configuration file is located within a configuration package that includes both `iDPConfig.py` along with `chemistry.py`. It is important that the `iDPConfig.py` and the `chemistry.py` be located in the same folder as the `iDPConfig.py` will automatically look to import `chemistry.py` from the same folder (Figure 7-16).

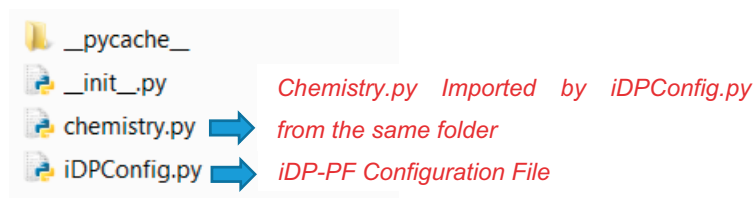


Figure 7-16. iDP-PF Configuration files package.

Input directory

As we have mentioned before, in order to prepare the PFLOTTRAN input files, the user will need to do a two-step procedure. First user needs to run iDP-DT and then iDP-PF. This is why the iDP-PF input directory must address to the output binary files which are the outputs of iDP-DT. (Figure 7-15 – Number 2 and Figure 7-17).

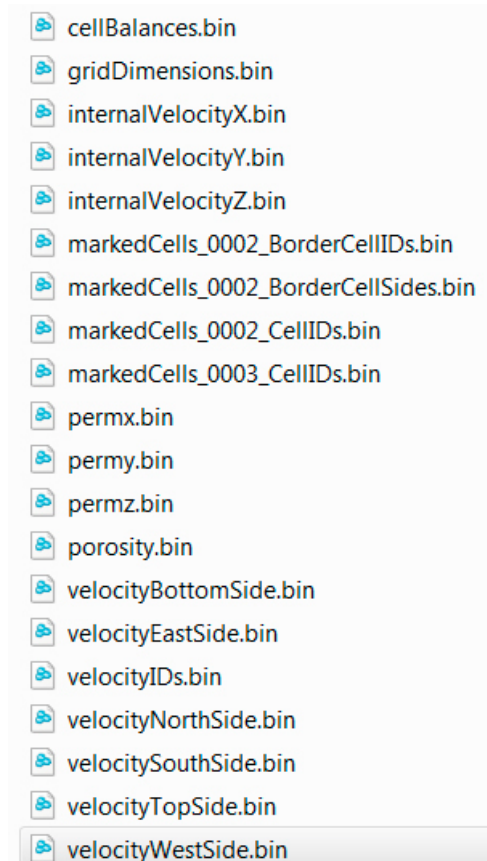


Figure 7-17. *iDP-PF Input = iDP-DT output (iDP-DT output used as input for iDP-PF).*

Output directory

The output directory where the results of iDP-PF results will be written. (Figure 7-15 – Number 3).

Depending on the configuration settings the following output is expected:

- PFLOTTRAN input configuration file (pflotran_gen.in)
- PFLOTTRAN input HDF5 files (Porosity.h5, Velocities.h5, Regions.h5, Materials.h5)
- HDF5 files containing information on Permeability and Mass Balances for further postprocessing in ParaView.
- An iDP-PF log file that contains useful information about iDP-PF.

RUN iDP-PF button

After all the input parameters to *iDP-PF* Configuration file and input directory and output directory have been set; the user can simply run *iDP-PF* by clicking on the *RUN iDP-PF* button.

7.3.2 Running iDP-PF from command line:

In Section 7.3.1 the different components of iDP-PF Graphical user Interface were explained. Although it is not recommended, the user can as well run iDP-PF from the command line interface by using the following command.

```
Python iDP.py -i inputDirectory -o outputDirectory ...iDPConfig.py
```

In order for the command above to work, all the relative paths of resources must be set correctly.

7.4 iDP Use cases/benchmarks

Within the framework of Bridge project, several benchmark examples have been prepared and iDP has been tested against at least five different benchmark problems including PHREEQC, OpenGeoSys, PFLOTRAN Standalone.

All the benchmarks are well documented in Bridge Project Report (Molinero et al. 2016) and further information regarding the benchmark examples can be consulted from the support team upon request.

7.4.1 iDP-DT Use case examples

When iDP-DT is installed, two use case examples by default are installed and located at the following address *C:\iDP_DT-Example* and they contain the following:

- Regular Grid
- Structured Irregular Grid

Regular Grid: First iDP-DT example consists of a rectangular box with regular discretization of 28 by 28 by 28 number of cells in x, y and z directions. Each cell in the domain has dimension of 80 by 65 by 19 metres. The grid for iDP-DT structured example is shown in Figure 7-18.

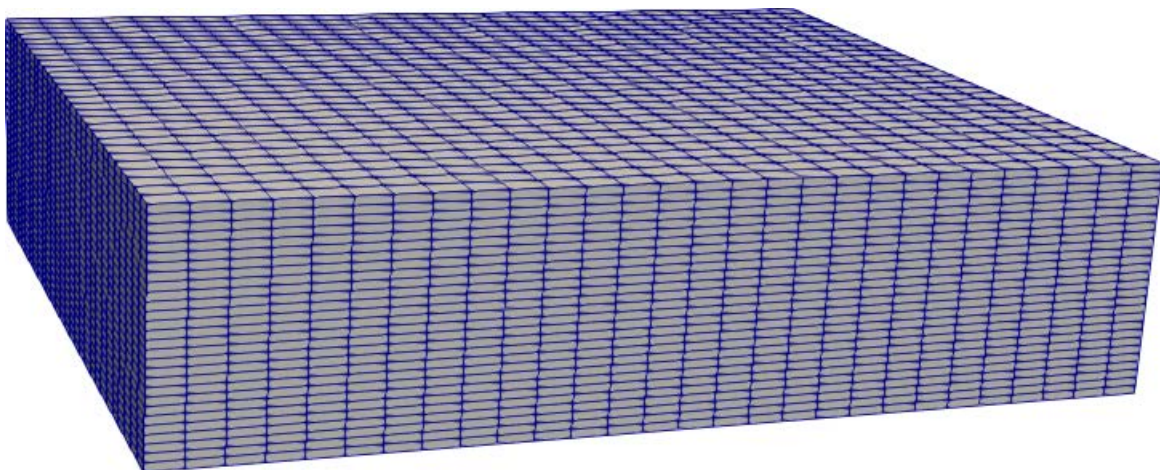


Figure 7-18. iDP-DT Structured grid.

Structured Irregular Grid: Second iDP-DT example has the same domain and it consists of the same discretization except that the grid is more refined in Z direction in the lower half of the domain and consists of 44 cells. The cell size in X direction is 80 metres and in y direction is 65 metres. The cell direction is Z in the top half of the domain is 19 and in the lower half of the domain is 9.5 metres (Figure 7-19).

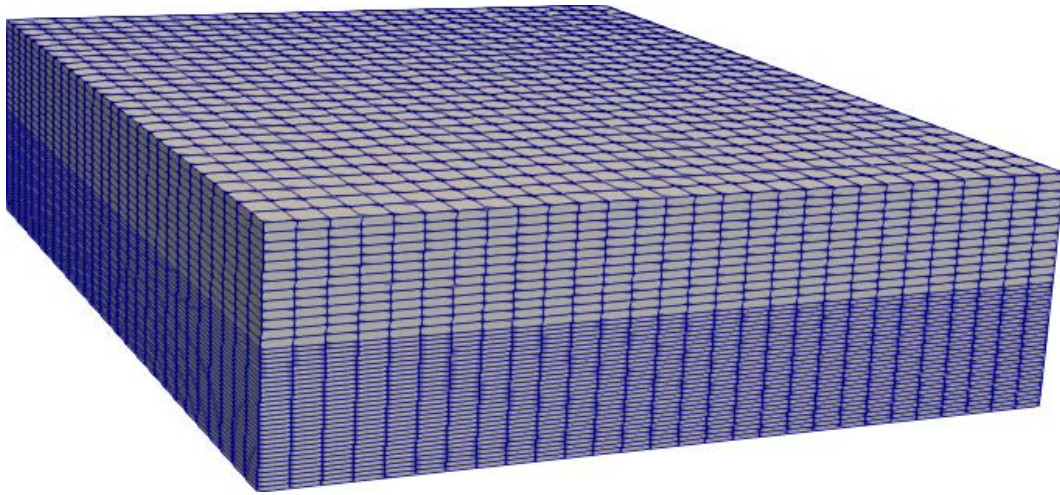


Figure 7-19. iDP-DT Structured irregular grid, the grid is more refined in the lower half of domain.

7.4.2 iDP-PF Use case examples

When iDP is installed, nine different use cases are installed by default at `C:\iDP_PF-Example`. The first four simple tests show below are built up on top of iDP-DT two test cases discussed in Section 7.4.1.

- GC Chemistry
- GC Tracer
- GC Tracer Fracture
- GC Tracer Random Field

Depending on the model under study the user can select one of these iDP-PF Configuration files to start with and further alter the configuration files to meet the users' needs.

In addition, iDP has been tested against five different benchmark problems, which are also intended to be useful iDP examples (Test Case 1 to 5 in Table 7-1). All these benchmarks are well documented in the literature. The results have been compared with other codes (e.g. PHREEQC, OpenGeoSys), with analytical solutions or with PFLOTRAN standalone. Table 7-1 contains the list of all iDP-PF validation tests.

Table 7-1. iDP-PF Use case examples.

iDP-PF USE cases	Address
GC Tracer	Based on iDP-DT regular grid and with conservative transport.
GCChemistry	Same as GC Tracer but with reactions included.
GCTracerFracture	Based on iDP-DT regular grid defining a region as fracture in iDP-PF.
GCTracerRandomField	Based on iDP-DT regular grid and with defining random regions.
TestCase1_1D_Calcite	100 × 1 × 1 regular grid with calcite dissolution (FAT and SAT validated with PFLOTRAN calcite problem).
TestCase2_Example11_phreeqc_cationExchange	200 × 1 × 1 regular grid with cation exchange (FAT and SAT validated with PHREEQC).
TestCase3_Dolomite	400 × 1 × 1 regular grid with calcite dissolution and dolomite precipitation (FAT and SAT validated with OpenGeoSys).
TestCase4_OxygenDiffusion_refined	232 × 1 × 105 structured irregular grid with three refinements in X direction. Oxygen penetration and consumption is simulated. (FAT and SAT validated with analytical solution).
TestCase5_copper_leaching	510 × 510 × 1 structured grid simulated in situ copper leaching. (FAT and SAT validated with PFLOTRAN).

FAT and SAT stand for Site Acceptance Test and Factual Acceptance Test, respectively.

In the following sub-sections, a through description of five iDP verification tests has been added. These benchmark tests are also discussed in Molinero et al. (2016).

Test Case one: 1D Calcite

This verification test is based on one of the reference examples documented in the PFLOTRAN user's manual (Lichtner et al. 2015). The results obtained with the default PFLOTRAN input file have been compared to those computed using iDP.

The test case consists of a one-dimensional domain that is initially filled with water in equilibrium with calcite. The mineral is homogeneously distributed with initial volume fraction of 10^{-5} . Acidic water (pH = 5) enters the domain from the inlet boundary. The ingress of the acidic front is temporarily buffered by the dissolution of calcite. Once the mineral is completely depleted there are no additional pH buffers and thus the acidic front penetrates deep along the domain. A conservative tracer, with unitary concentration, is also added to the boundary water.

The parameters of the model are summarized in Table 7-2 and the initial and boundary waters are listed in Table 7-3. The PFLOTRAN simulations have been carried out:

1. Using the standard PFLOTRAN input file for the 1D calcite dissolution case. In this case, a constant uniform velocity is prescribed (see Table 7-2).
2. Using iDP. In this case, first a DarcyTools model, with the same physical parameters specified in Table 7-2, has been set-up and solved. iDP has then been used to generate the PFLOTRAN input file and the external velocity file, which is read by PFLOTRAN.

Breakthrough curves for the conservative tracer, pH and calcium at the outlet of the domain are shown in Figure 7-20 to Figure 7-23. A profile showing the distribution of calcite (i.e. calcite volume fraction) after 30 years from the beginning of the simulation is shown in Figure 7-23. For all the considered species, the match between the two solutions (i.e. PFLOTRAN vs. iDP) is good and the verification exercise is considered successful.

Table 7-2. Parameters used in the problem.

Parameter	Value
Domain length (L)	100.0 m
Darcy Velocity (q_x)	1.0 m/y
Porosity (Φ)	0.25

Table 7-3. Initial and boundary waters used in the problem.

Parameter	Initial water		Boundary water	
pH	8		5	
Species	Concentration (mol/L _w)	Note	Concentration (mol/L _w)	Note
Ca ²⁺	5.0×10^{-4}	Equilibrated with calcite Equilibrium with CO ₂ gas at given partial pressure	5.0×10^{-6}	Charge balance
HCO ₃ ⁻	1.0×10^{-3} bar		1.0×10^{-3}	
Tracer	1.0×10^{-10}		1.0	

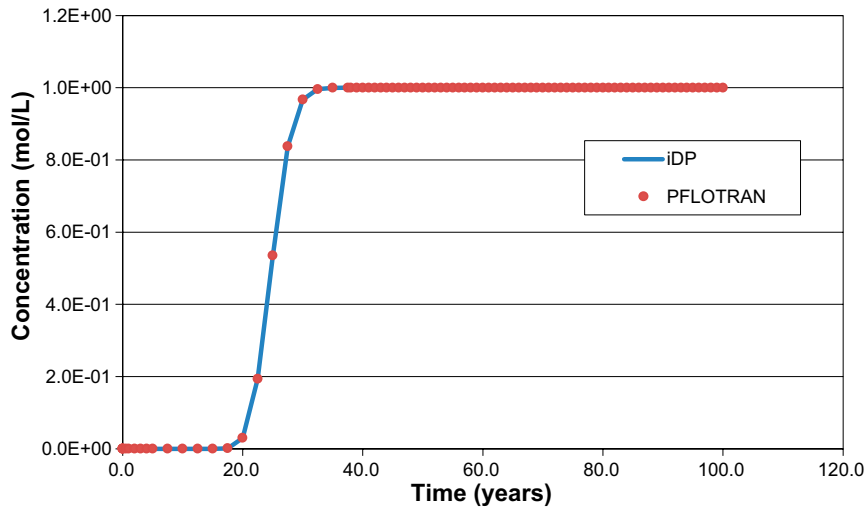


Figure 7-20. Tracer breakthrough curve at the outlet of the domain computed with iDP (blue curve) and PFLOTRAN (red dots).

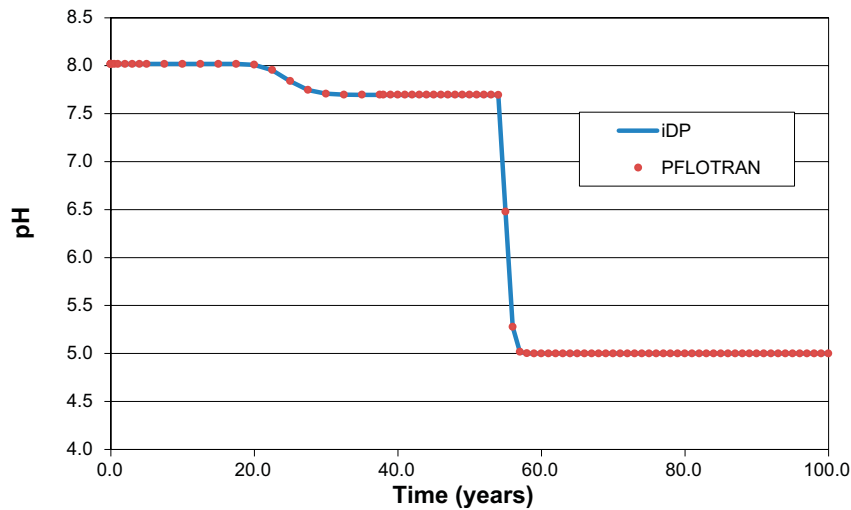


Figure 7-21. pH evolution at the outlet of the domain computed with iDP (blue curve) and PFLOTRAN (red dots).

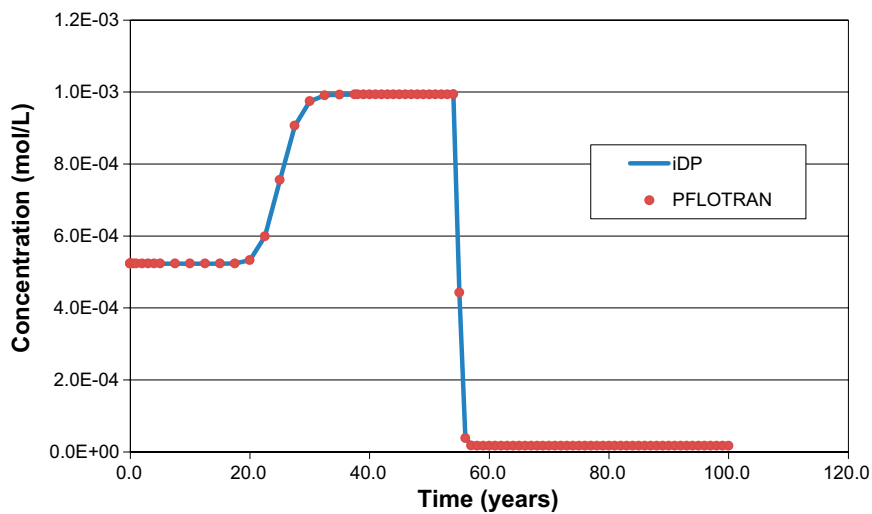


Figure 7-22. Calcium breakthrough curve at the outlet of the domain computed with iDP (blue curve) and PFLOTRAN (red dots).

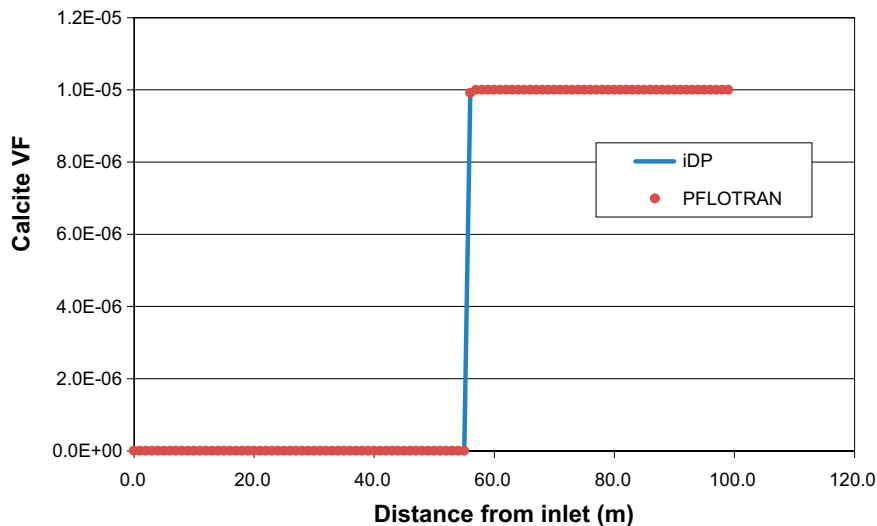


Figure 7-23. Calcite volume fraction along the considered domain after 30 years from the beginning of the simulation.

Test Case 2: Cation exchange

Here, we use the example 11 of the PHREEQC User's Manual as benchmark for iDP. In the example 11 of PHREEQC (Parkhurst and Appelo 2013), the chemical composition of the effluent from a column containing a cation exchanger is simulated. Initially, the column contains a sodium-potassium-nitrate solution in equilibrium with the cation exchanger. The column is then flushed with three pore volumes of calcium chloride solution. Calcium, potassium, and sodium react to equilibrium with the exchanger at all times.

The sodium initially present in the water exchanges with the calcium of the boundary water and is released as long as the exchanger contains still sodium. Because potassium has a higher affinity with the exchanger, is released after sodium. Finally, when all of the potassium has been released, the concentration of calcium increases to a steady-state value equal to the concentration in the boundary water.

The parameters of the problem are listed in Table 7-4.

Table 7-4. Parameters used in the problem.

Parameter	Value
Domain length (L)	0.08 m
Darcy velocity (q_x)	2.78×10^{-7} m/s
Porosity (Φ)	0.1

Table 7-5. Initial and boundary waters used in the problem.

Parameter	Initial water	Boundary water
pH	7	7
Species	Concentration (mol/L _w)	Concentration (mol/L _w)
Ca ²⁺	1.0×10^{-12}	6.0×10^{-4}
K ⁺	2.0×10^{-4}	1.0×10^{-12}
Na ⁺	1.0×10^{-3}	1.0×10^{-12}
Cl ⁻	1.0×10^{-12}	1.2×10^{-3}
Tracer	1.0×10^{-12}	1.2×10^{-3}

In this modelling case, first a DarcyTools model, with the same physical parameters specified in Table 7-4, has been set-up and solved. iDP has then been used to generate the PFLOTRAN input files, containing porosity and velocity, which are input to PFLOTRAN.

In Figure 7-24, the results computed using iDP are compared to the solution given by PHREEQC. The agreement between the two solutions is good. Slight discrepancies are most probably due to the different numerical solvers and related tolerances used in PHREEQC and PFLOTRAN.

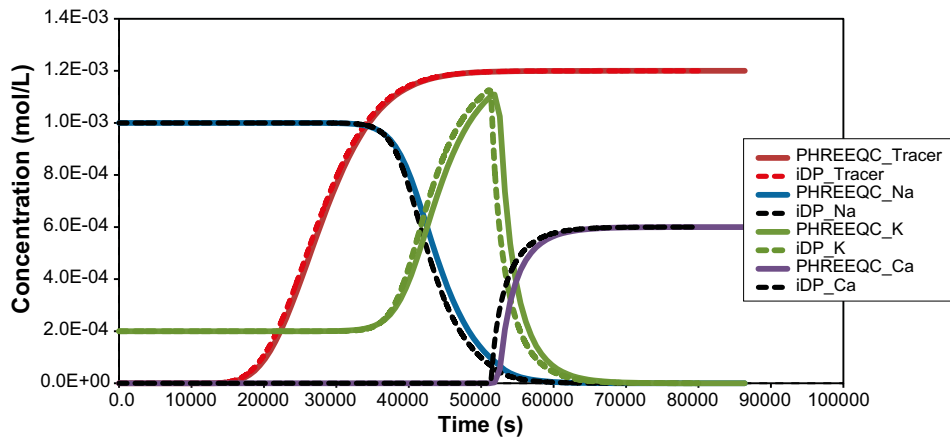


Figure 7-24. Comparison of breakthrough curves computed using PHREEQC (solid line) and iDP (dots).

Test Case 3: 1D Calcite dissolution and dolomite precipitation

This benchmark is presented in Section 15.1 of Kolditz et al. (2012) and is denoted as “1D Reactive Transport: Calcite dissolution and precipitation”. A one-dimensional column that initially contains calcite is continuously flushed with water that contains magnesium chloride. With the movement of the water front, calcite starts to dissolve and dolomite is temporarily formed.

The parameters of the benchmark are summarized in Table 7-6.

Table 7-6. Parameters of the benchmark problem.

Parameter	Value
Domain length (L)	0.5 m
Darcy velocity (q_x)	3.0×10^{-6} m/s
Porosity (Φ)	0.32
Bulk density	1830 kg/m ³
Longitudinal dispersivity	0.0067 m
Calcite abundance	2×10^{-4} mol/L

The initial and boundary waters used in this model are shown in Table 7-7.

Table 7-7. Initial and boundary waters.

Parameter	Initial water	Boundary water
pH	9.91	7
Species	Concentration (mol/L _w)	Concentration (mol/L _w)
Ca ²⁺	1.227×10^{-4}	1.0×10^{-40}
HCO ₃ ⁻	1.227×10^{-4}	1.0×10^{-40}
Mg ⁺⁺	1.0×10^{-40}	1.0×10^{-3}
Cl ⁻	1.0×10^{-40}	2.0×10^{-3}
Tracer	1.0×10^{-40}	2.0×10^{-3}

To solve the benchmark in iDP, a DarcyTools model has been previously set-up and solved using the parameters listed in Table 7-6. iDP has then been used to generate the PFLOTRAN input files, containing porosity and velocity, which are input to PFLOTRAN.

In Figure 7-25, the results of iDP are compared to those reported in Kolditz et al. (2012). The agreement between the two solutions is good.

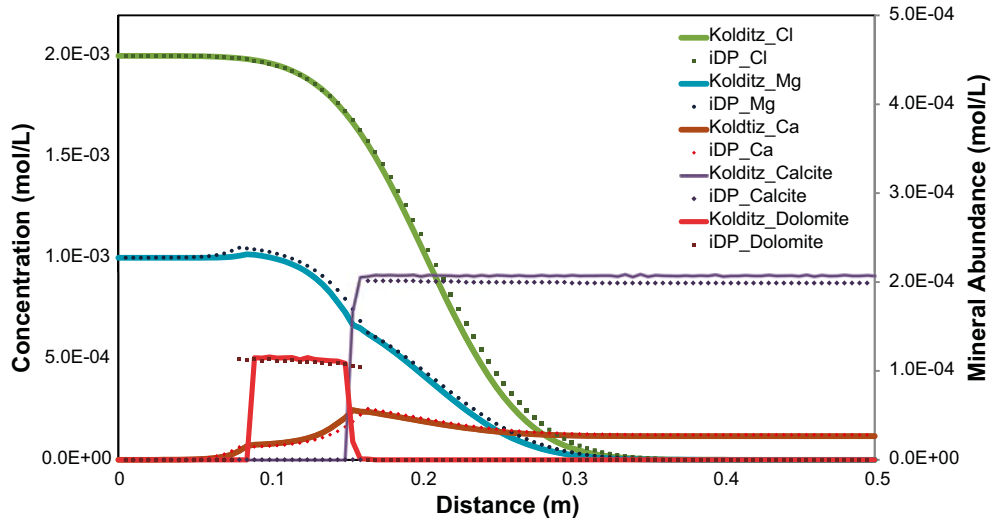


Figure 7-25. Benchmark results from iDP compared vs. Kolditz (GEOSys-GEM).

Test Case 4: Copper leaching

The copper leaching problem simulates in situ mining of copper by injection of an acidic solution ($pH = 1$) into the subsurface. The acidic solution dissolves chrysocolla ($CuSiO_3 \cdot 2H_2O$) releasing copper into the solution. Copper is extracted above ground and secondary minerals precipitate in the Chrysocolla front's path (Lichtner, 1996). A 5-spot well distribution (i.e. one injection well in the center and four extraction wells in the corner) is used in the simulation. Due to symmetry, only one quarter of the domain is modelled (Figure 7-26).

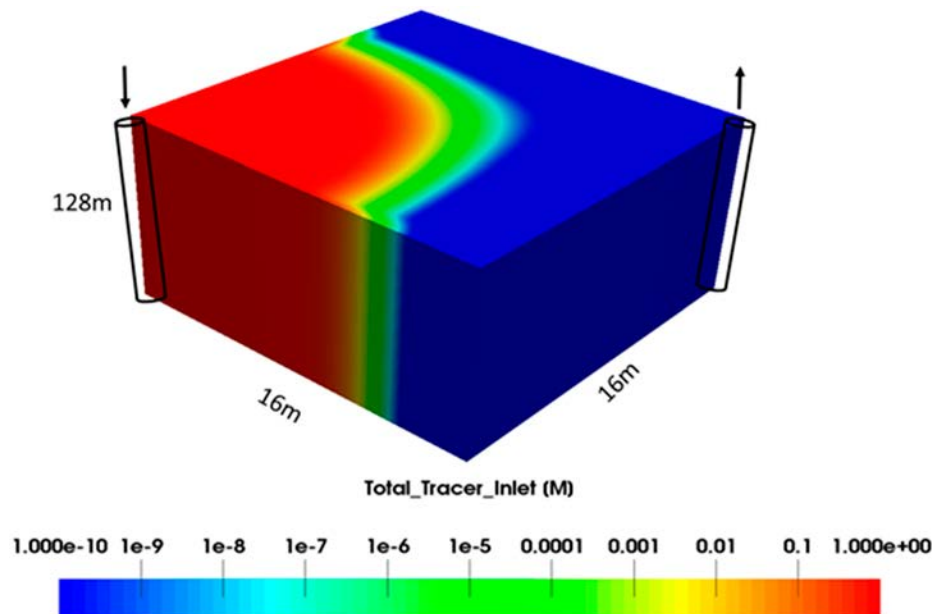


Figure 7-26. Sketch of the copper leaching.

The parameters of the problem are summarized in Table 7-8.

Table 7-8. Copper leaching model parameters.

Parameter	Value
Domain length_X	16.0 m
Domain length_Y	16.0 m
Domain length_Z	128.0 m
Input rate	0.68 kg/s
Output rate	-0.68 kg/s
Porosity (Φ)	0.05
Permeability	$1.5 \times 10^{-13} \text{ m}^2$

The initial and inlet boundary conditions (i.e. initial and boundary waters) are described in Table 7-9. The minerals and related volume fractions are specified in Table 7-10. Minerals with volume fraction equal to zero are not initially present but can precipitate if over-saturation is reached.

Table 7-10. Copper leaching: initial and boundary waters.

Parameter	Initial water		Boundary water	
pH	8.0		1.0	
Species	Concentration (mol/L _w)	Note	Concentration (mol/L _w)	Note
Ca ²⁺	6.5×10^{-4}	Equilibrium with calcite	1.1×10^{-2}	Equilibrium with gypsum
Na ⁺	5.0×10^{-3}	Total aqueous component concentration	5.0×10^{-3}	Total aqueous component concentration
K ⁺	2.5×10^{-5}	Equilibrium with muscovite	1.3×10^{-4}	Equilibrium with jarosite
Al ⁺⁺⁺	2.8×10^{-17}	Total aqueous component concentration	2.5×10^{-2}	Total aqueous component concentration
Cu ⁺⁺	6.4×10^{-9}	Equilibrium with chrysocolla2	1.0×10^{-8}	Total aqueous component concentration
Cl ⁻	3.7×10^{-3}	Charge Balance	5.0×10^{-3}	Total aqueous component concentration
Fe ⁺⁺	1.2×10^{-23}	Equilibrium with goethite	3.4×10^{-9}	Equilibrium with goethite
SiO _{2(aq)}	1.8×10^{-4}	Equilibrium with chalcedony	1.9×10^{-3}	Equilibrium with SiO ₂ (am)
HCO ₃ ⁻	-3.0	Equilibrium with CO ₂ (g)	-2.0	Equilibrium with CO ₂ (g)
SO ₄ ⁻	5.0×10^{-4}	Total aqueous component concentration	6.1×10^{-2}	Charge balance
O _{2(aq)}	-0.699	Equilibrium with O ₂ (g)	-0.699	Equilibrium with O ₂ (g)
Tracer	1.0×10^{-10}	Total aqueous component concentration	1.0	Total aqueous component concentration

Table 7-10. Minerals volume fractions present in initial water.

Parameter	Initial mineral volume fraction	Specific surface area (m ² mineral/m ³ bulk)
Alunite	0.0	1 cm ² /cm ³
Chrysocolla2	5.0×10^{-3}	1 cm ² /cm ³
Goethite	2.5×10^{-2}	1.0 cm ² /cm ³
Gypsum	0.0	1.0 cm ² /cm ³
Jarosite	0.0	1.0 cm ² /cm ³
Jurbanite	0.0	1.0 cm ² /cm ³
Kaolinite	5.0×10^{-2}	1.0 cm ² /cm ³
Muscovite	5.0×10^{-2}	1.0 cm ² /cm ³
SiO ₂ (am)	0.0	1.0 cm ² /cm ³
Quartz	8.2×10^{-1}	1.0 cm ² /cm ³

Compared to the original PFLOTTRAN problem, when running Copper Leaching in iDP two simplification assumptions are needed.

First, in the PFLOTTRAN original problem the SOURCE_SINK card is used, which allows flow and transport boundary conditions (typically Neumann and Dirichlet, respectively) to be defined along 1D and 2D geometrical entities. In iDP, flow conditions are simulated in DarcyTools whereas the PFLOTTRAN transport conditions always apply over cell faces.

As iDP only works over sub-domains of DarcyTools, the biggest DarcyTools sub-domain that can be extracted is one cell smaller in all the six directions (i.e. $x^+/x^-; y^+/y^-; z^+/z^-$), which in turns means that transport conditions, which are applied as Dirichlet conditions over the external faces of the first (bottom-left corner) and last cell (upper-right corner) of the sub-domain, are shifted one cell compared to the original problem.

The original problem set-up and the simplifications required by iDP are illustrated in Figure 7-27. To minimize the effect of these simplifications, an extremely fine mesh refinement has been used.

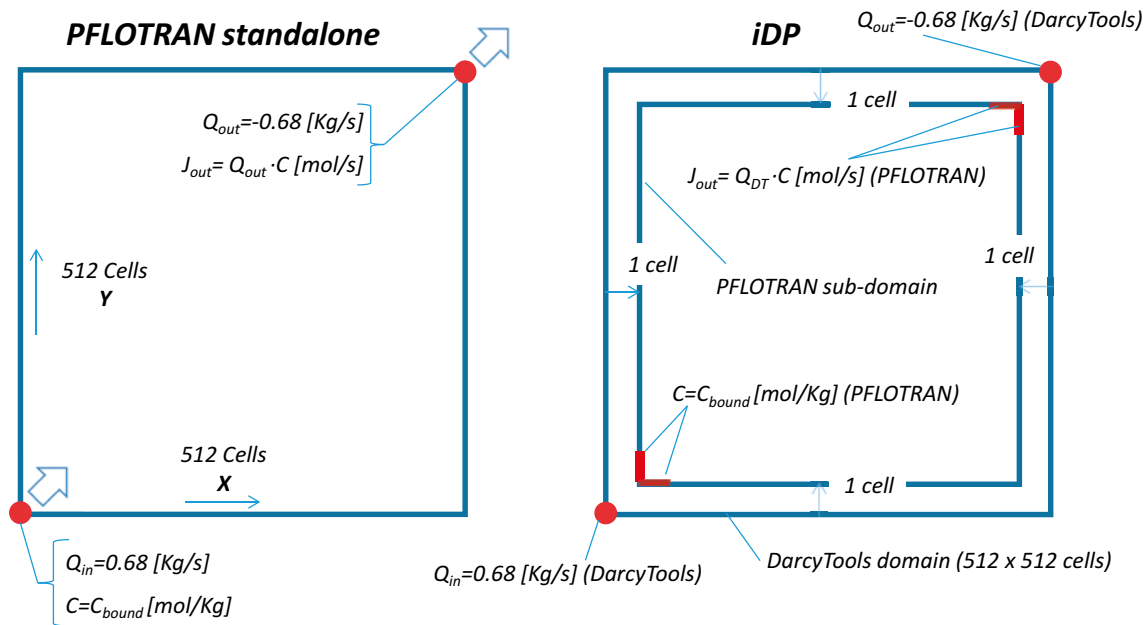


Figure 7-27. Original set-up of PFLOTTRAN (left) and set-up used in iDP (right).

The results computed by PFLOTTRAN standalone and iDP are used to produce maps of concentrations of Tracer, pH, Chrysocolla (volume fraction), Cu^{2+} and Jurbanite (volume fraction) (Figure 7-28 to Figure 7-32). The agreement between the two solutions (i.e. PFLOTTRAN standalone vs. iDP) is good. Slight discrepancies are observed close to the boundary as a result of the simplification assumptions discussed before.

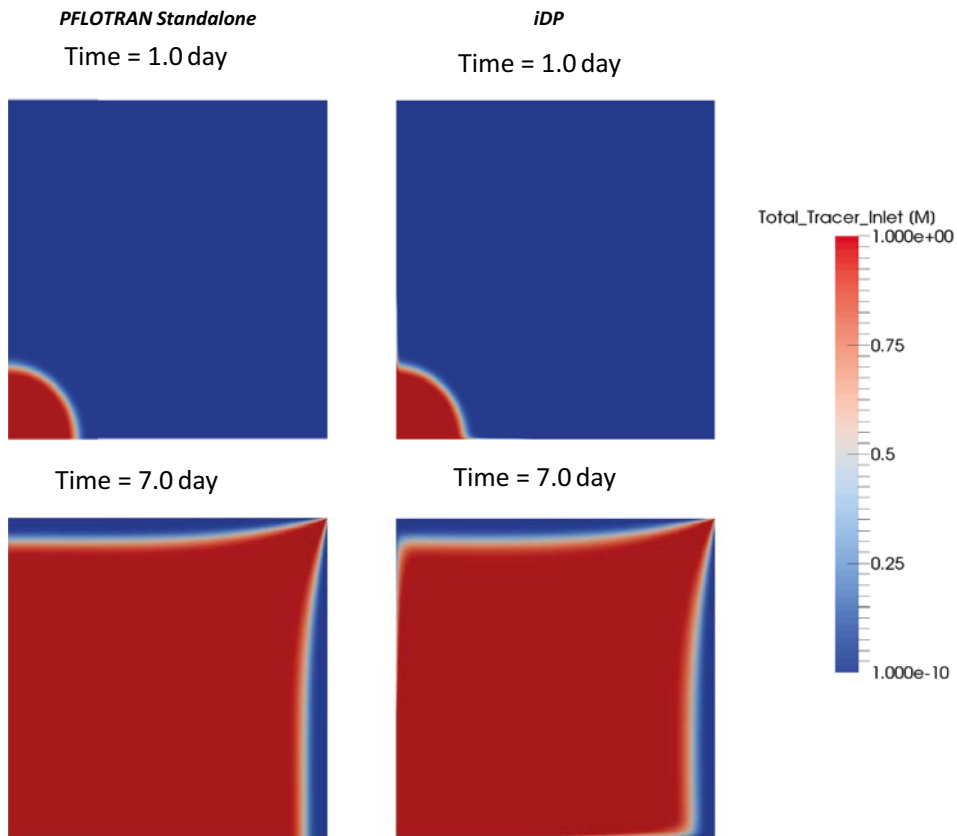


Figure 7-28. Tracer concentration computed by PFLOTTRAN (left) and iDP (right) at 1.0 day (top) and 7.0 day (bottom).

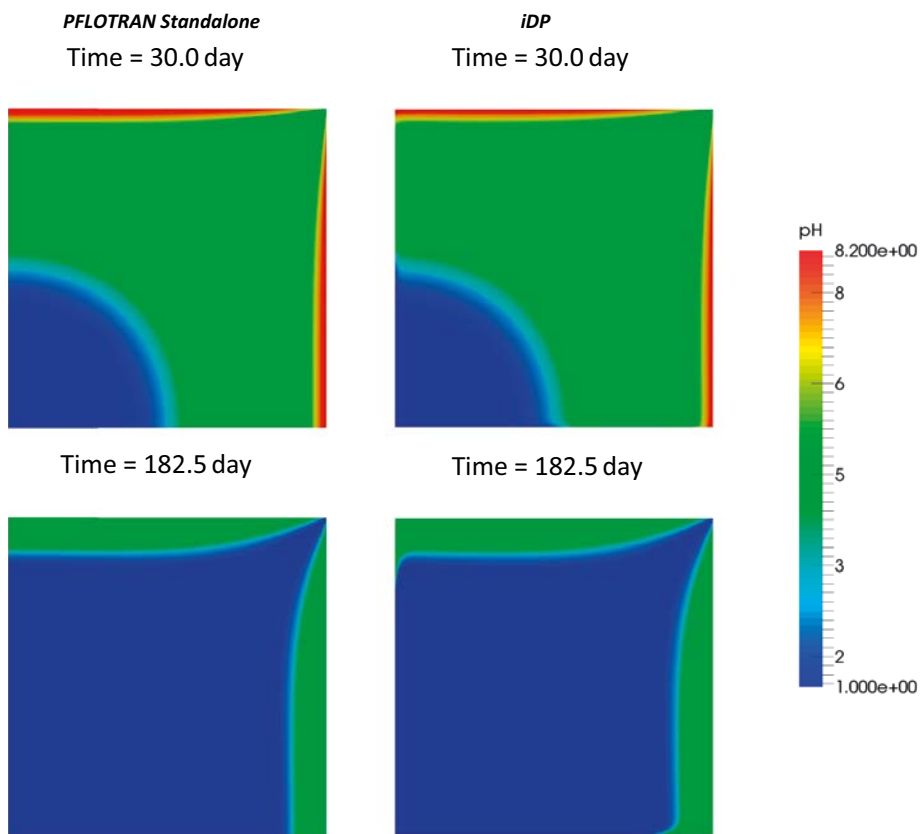


Figure 7-29. pH computed by PFLOTTRAN (left) and iDP (right) at 30.0 day (top) and 182.5 day (bottom).

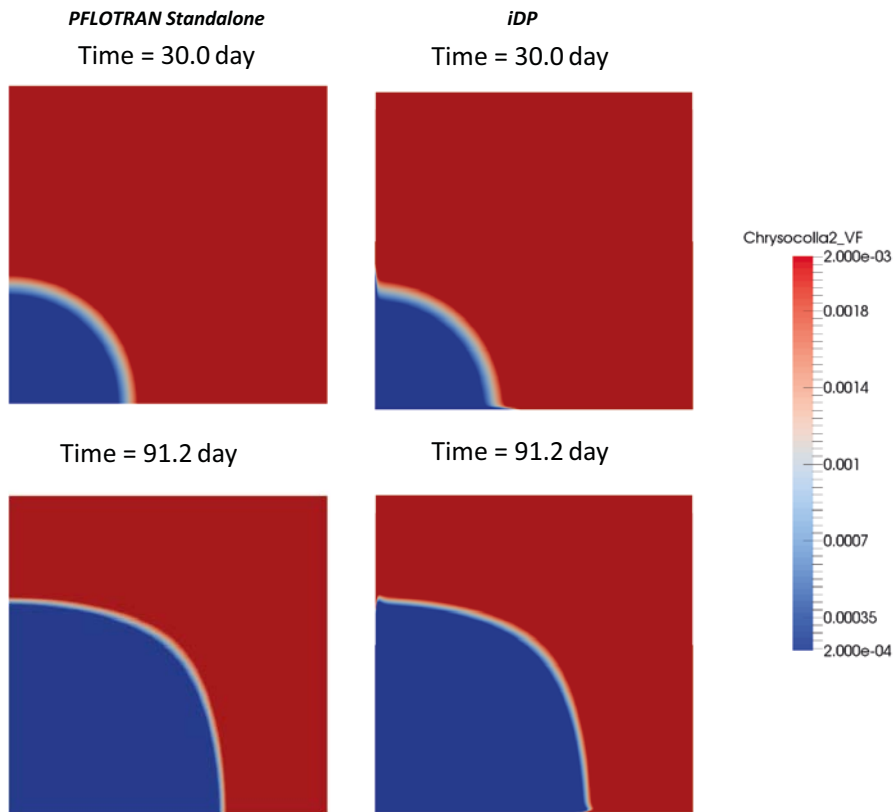


Figure 7-30. Chrysocolla volume fraction computed by PFLOTTRAN (left) and iDP (right) at 30.0 day (top) and 91.2 day (bottom).

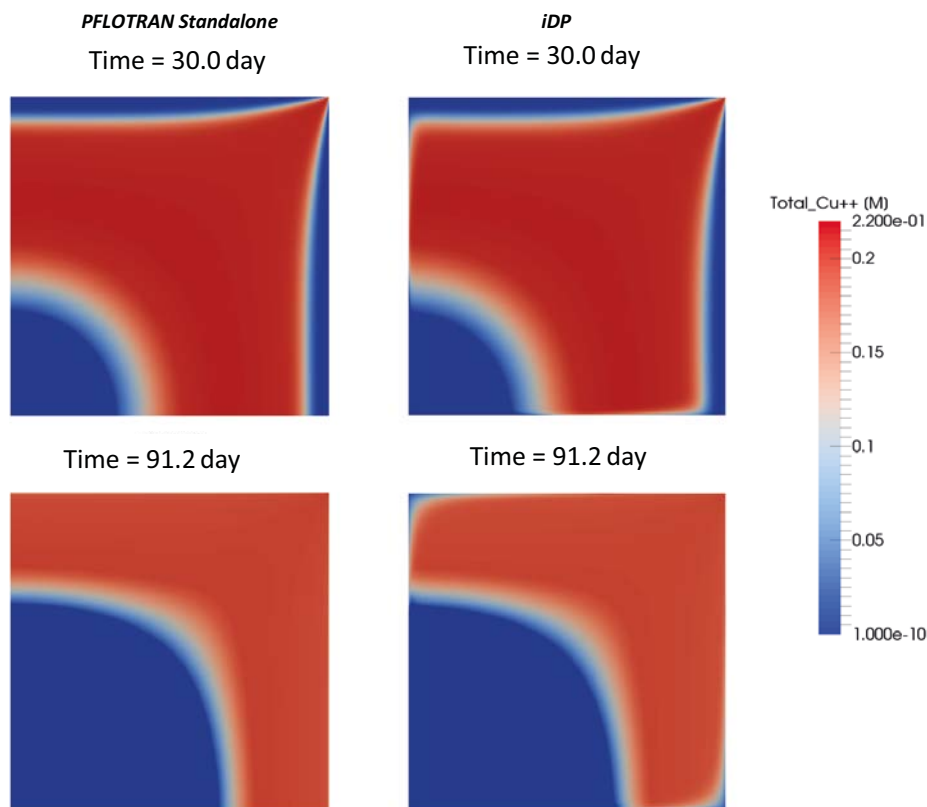


Figure 7-31. Cu²⁺ concentration computed by PFLOTTRAN (left) and iDP (right) at 30.0 day (top) and 91.2 day (bottom).

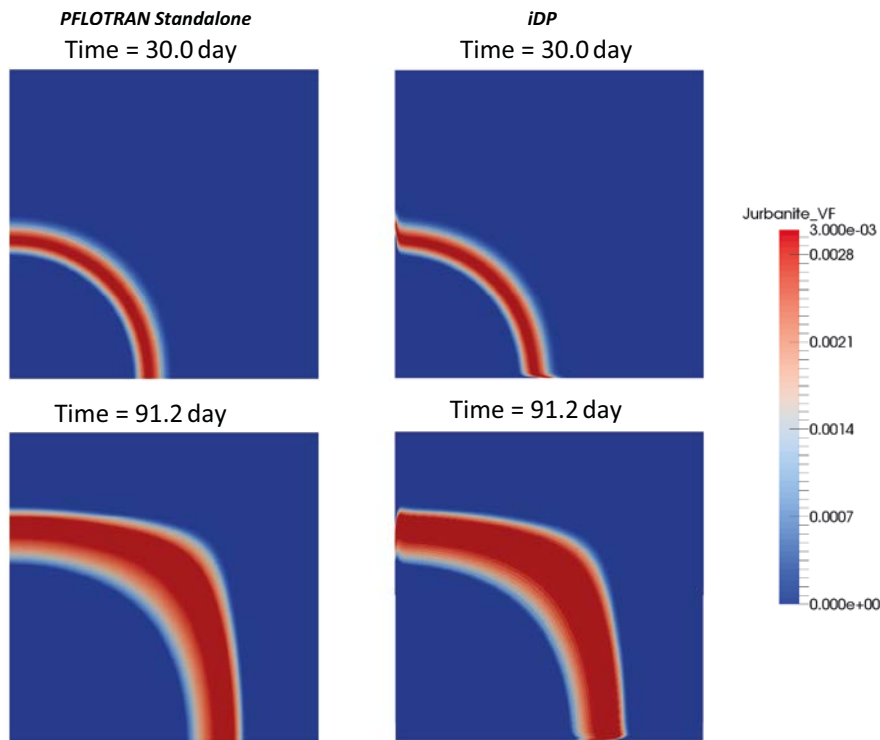


Figure 7-32. Jurbanite Volume Fraction computed by PFLOTRAN (left) and iDP (right) at 30.0 day (top) and 91.2 day (bottom).

Test Case 5: Matrix diffusion and oxygen consumption

Here, we simulate the ingress of oxygen in a fracture-matrix system and its consumption by abiotic processes. The results (i.e. steady-state concentration profiles of oxygen) are compared to the analytical solution developed by Sidborn et al. (2010).

A sketch of the model set-up is shown in Figure 7-33. Oxygenated water enters the model domain from the left boundary of the fracture and is progressively displaced by advective fluxes along the fracture. Part of this oxygen diffuses into the matrix where is consumed by the homogeneous oxidation of ferrous iron. Ferrous iron is in turn released by the non-oxidative dissolution of biotite, which is present in the matrix (volume fraction, $V_f = 0.078$) and is absent in the fracture. Given the relatively high biotite concentration, at some point in time the concentration profile of oxygen reaches a steady-state.

In this numerical exercise, the groundwater flow velocity field has been computed using DarcyTools. Darcy's velocities and material properties (i.e. porosity) have then been extracted using iDP and used as input to PFLOTRAN.

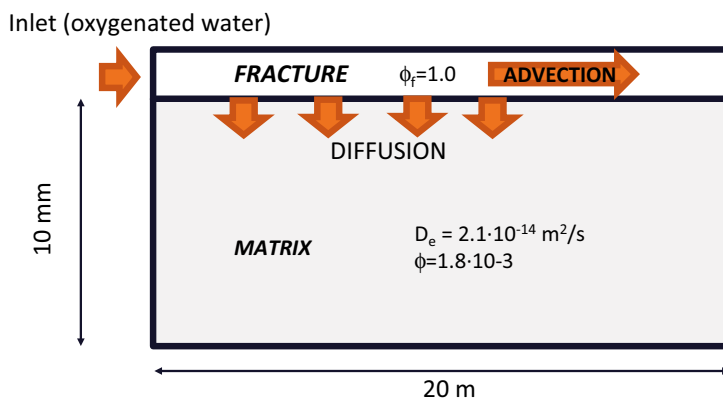


Figure 7-33. Sketch of the oxygen intrusion problem. All the parameters are taken from Sidborn et al. (2010).

As mentioned before, biotite is assumed to be the source of dissolved ferrous iron. Its non-oxidative dissolution is represented using the following rate-law:

$$r_{diss} [mol_{Fe} m^{-2} s^{-1}] = k_{Fe,diss} \cdot \left(\frac{V_f^{biot}}{V_{f0}^{biot}} \right)^{2/3} \cdot \left(1 - \frac{C_{Fe}}{C_{Fe,sol}} \right) \quad \text{Equation 7-1}$$

where $k_{Fe,diss}$ is the maximum release rate of ferrous iron, C_{Fe} is the iron aqueous concentration, $C_{Fe,sol}$ is the iron solubility and V_{f0}^{biot} and V_f^{biot} are, respectively, the initial and the computed biotite volume fraction.

The dissolved ferrous ions are oxidized by oxygen as follows:



widely used rate law for this reaction can be written for $pH > 4.5$ (Stumm and Lee 1961):

$$R_{ox} [mol_{Fe} L^{-1} s^{-1}] = k_r C_{Fe} P_{O_2} C_{OH^-}^2 \quad \text{Equation 7-3}$$

where k_r is a homogeneous oxidation rate constant, P_{O_2} is the partial pressure of oxygen and C_{OH^-} is the concentration of OH^- .

In PFLOTRAN, custom kinetic reaction rates have to be defined using the “Reaction Sandbox” (see appendix of the PFLOTRAN User’s Manual for further details). To implement a new reaction within the reaction sandbox, the user has to hardcode it in the related subroutines and classes of the code (i.e. “reaction_sandbox.F90” and “reaction_sandbox_base.F90”). Although this is a very powerful code capability, it is evident that the use of the reaction sandbox is not straightforward as it requires changes in the PFLOTRAN source code, which has then to be recompiled.

In order to keep this exercise at a simple level of complexity, Equation 7-3 has been simplified by assuming a constant OH^- concentration (as in Sidborn et al. 2010, who also fixed the hydroxide ion concentration, $[OH^-]$ to 1×10^{-6} mol/L_w). In this way, Equation 7-2 reduces to a second order kinetic reaction, which in PFLOTRAN can be easily defined using the “General_Reaction” card.

By default, dissolution/precipitation reactions are written in PFLOTRAN using transition state rate laws. The implementation of alternative formulations is not straightforward (they need to be hard-coded in the “Reaction Sandbox” subroutine). Thus, we have simplified the rate of biotite dissolution as follows:

$$r_{diss} [mol_{Fe} m^{-2} s^{-1}] = k_{Fe,diss} \cdot \left(1 - \frac{C_{Fe}}{C_{Fe,sol}} \right) \quad \text{Equation 7-4}$$

It is worthwhile noting that in Equation 7-4, the rate of dissolution of biotite is no longer weighted by the relative mineral abundance.

The results of the iDP simulation show that a steady-state profile of oxygen is reached after 200 days of simulation time, approximately (Figure 7-34).



Figure 7-34. Oxygen concentration computed using iDP after 200 days of simulation time. The domain is magnified by a factor of 400 in the direction perpendicular to the fracture.

The oxygen concentration profiles in the rock matrix at different fracture locations are shown in Figure 7-35. The results are compared to those obtained using the analytical solution of Sidborn et al. (2010). When the concentration in the fracture is high (i.e. close to the inlet boundary), iDP underestimates the extent of oxygen penetration, whereas far from the inlet boundary (i.e. low oxygen concentrations in the fracture) the agreement between the two solutions is good.

The reason for the discrepancy observed at high fracture concentrations is to be found in the simplified approach used to represent the biotite dissolution rate (i.e. Equation 7-4). In fact, in PFLOTRAN the rate law is not weighted by the relative mineral abundance. This means that close to the injection boundary, that is where more biotite is depleted, the rate of mineral dissolution computed by PFLOTRAN is actually higher than that considered by the analytical solution resulting in a stronger buffering effect.

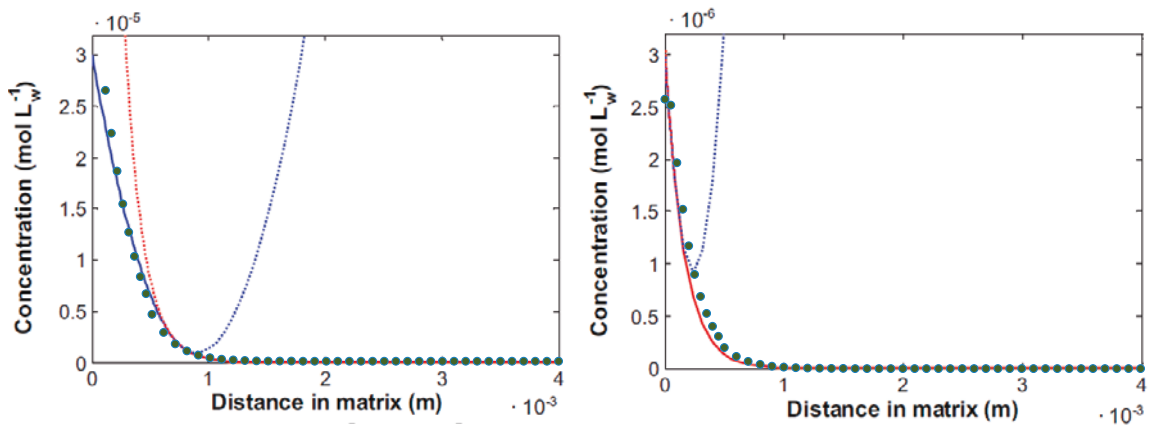


Figure 7-35. Steady-state oxygen concentration profiles computed in the matrix at different fracture concentrations. iDP results (green dots) are plotted over the results of Sidborn et al. (2010).

References

SKB's (Svensk Kärnbränslehantering AB) publications can be found at www.skb.com/publications.

Hammond G E, Lichtner P C, Rockhold M L, 2011. Stochastic simulation of uranium migration at the Hanford 300 Area. *Journal of Contaminant Hydrology* 120–121, 115–128.

Hammond G E, Lichtner P C, Mills R T, 2014. Evaluating the performance of parallel subsurface simulators: an illustrative example with PFLOTRAN. *Water Resources Research* 50, 208–228.

Kolditz O, Bauer S, Bilke L, Böttcher N, Delfs J-O, Fischer T, Görke U J, Kalbacher T, Kosakowski G, McDermott C, Park C H, Radu F, Rink K, Shao H, Shao H B, Sun F, Sun Y Y, Singh A K, Taron J, Walther M, Wang W, Watanabe N, Wu Y, Xie M, Xu W, Zehner B, 2012. OpenGeoSys: an open-source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media. *Environmental Earth Sciences* 67, 589–599.

Lichtner P C, 1996. Modeling reactive flow and transport in natural systems, in: *Proceedings of the Rome Seminar on Environmental Geochemistry: held in Castelnuovo Di Porto on May 22–26, 1996*. Pisa, Italy: Pacini Editore, 72.

Lichtner P, Hammond G, Lu C, Karra S, Bisht G, Andre B, Mills R, Kumar J, 2015. PFLOTRAN user manual: A massively parallel reactive flow and transport model for describing surface and subsurface processes. (Report No.: LA-UR-15-20403). Los Alamos National Laboratory, USA.

Molinero J, Trinchero P, de Vries L M, Ebrahimi H, Svensson U, 2016. The BRIDGE project. Development, testing and application of a high performance computing framework for reactive transport modelling in crystalline rocks (iDP). SKB R-15-17, Svensk Kärnbränslehantering AB.

Parkhurst D L, Appelo C A J, 2013. Description of input and examples for PHREEQC version 3: a computer program for speciation, batch-reaction, one-dimensional transport, and inverse geochemical calculations. *Techniques and Methods* 6, A43, U.S. Geological Survey, Denver, Colorado.

Sidborn M, Sandström B, Tullborg E-L, Salas J, Maia F, Delos, A, Molinero J, Hallbeck L, Pedersen K, 2010. SR-Site: Oxygen ingress in the rock at Forsmark during a glacial cycle. SKB TR-10-57, Svensk Kärnbränslehantering AB.

Stumm W, Lee GF, 1961. Oxygenation of ferrous iron. *Industrial & Engineering Chemistry Research* 53, 143–146.

Svensson U, Ferry M, 2014. DarcyTools: a computer code for hydrogeological analysis of nuclear waste repositories in fractured rock. *Journal of Applied Mathematics and Physics* 2, 365–383.

Svensson U, Ferry M, Kuylenstierna H-O, 2010. DarcyTools version 3.4 – concepts, methods and equations. SKB R-07-38, Svensk Kärnbränslehantering.

SKB is responsible for managing spent nuclear fuel and radioactive waste produced by the Swedish nuclear power plants such that man and the environment are protected in the near and distant future.

skb.se