Modelling Reactive Transport Process with OpenGeoSys

with Decay-Chain Benchmark as an Example

Dr. Haibing Shao

haibing.shao@ufz.de

Helmholtz Centre for Environmental Research - UFZ, Leipzig, Germany

Prepared for the Lecture “**Modellierung von Hydrosystemen**”

in TU Dresden on 04.07.2024

Overview

- Overview of Reactive Transport Process Features
- The Decay-Chain Benchmark
- Exercise: Set up the Decay-Chain prj File
- Simulation and Visualization

Overview

- **Overview of Reactive Transport Process**
- **The Decay-Chain Benchmark**
- **Exercise: Set up the Decay-Chain prj File**
- **Simulation and Visualization**

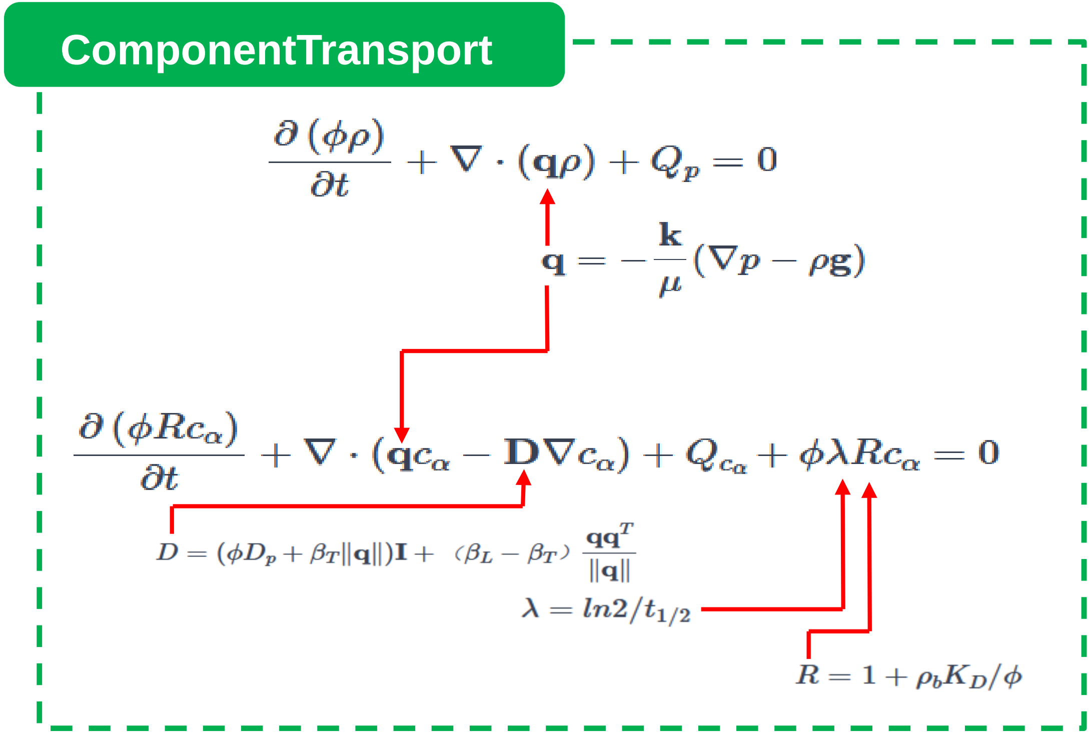
How is Reactive Transport simulated by the Component Transport Process?

The component transport process can handle

- Fully saturated condition
- Variable-density flow
- Component transport with advection, dispersion, sorption and decay
- Reactive transport

There are two sets of process variables

- Pressure
- Concentration of each component



ComponentTransport

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla\cdot(\mathbf{q}\rho) + Q_p = 0$$

$$\mathbf{q} = -\frac{\mathbf{k}}{\mu}(\nabla p - \rho\mathbf{g})$$

$$\frac{\partial(\phi R c_\alpha)}{\partial t} + \nabla\cdot(\mathbf{q}c_\alpha - \mathbf{D}\nabla c_\alpha) + Q_{c_\alpha} + \phi\lambda R c_\alpha = 0$$

$$D = (\phi D_p + \beta_T\|\mathbf{q}\|)\mathbf{I} + (\beta_L - \beta_T)\frac{\mathbf{q}\mathbf{q}^T}{\|\mathbf{q}\|}$$

$$\lambda = ln2/t_{1/2}$$

$$R = 1 + \rho_b K_D/\phi$$

Modelling Reactive Transport Process with different algorithms

OP – Operator Splitting

Transport stage

$$\begin{array}{|c|c|c|c|} \hline K_{c1c1} & & & \\ \hline & K_{c2c2} & & \\ \hline & & K_{c3c3} & \\ \hline & & & K_{c4c4} \\ \hline \end{array} \quad \begin{array}{|c|} \hline C_1 \\ \hline C_2 \\ \hline C_3 \\ \hline C_4 \\ \hline \end{array} = \begin{array}{|c|} \hline b_1 \\ \hline b_2 \\ \hline b_3 \\ \hline b_4 \\ \hline \end{array}$$

Reaction stage

Call Phreeqc to compute $\mathbf{R}^{min}$

$$\begin{array}{|c|c|c|c|} \hline M_{c1c1} & & & \\ \hline & M_{c2c2} & & \\ \hline & & M_{c3c3} & \\ \hline & & & M_{c4c4} \\ \hline \end{array} \quad \begin{array}{|c|} \hline C_1 \\ \hline C_2 \\ \hline C_3 \\ \hline C_4 \\ \hline \end{array} = \begin{array}{|c|} \hline b_1 \\ \hline b_2 \\ \hline b_3 \\ \hline b_4 \\ \hline \end{array}$$

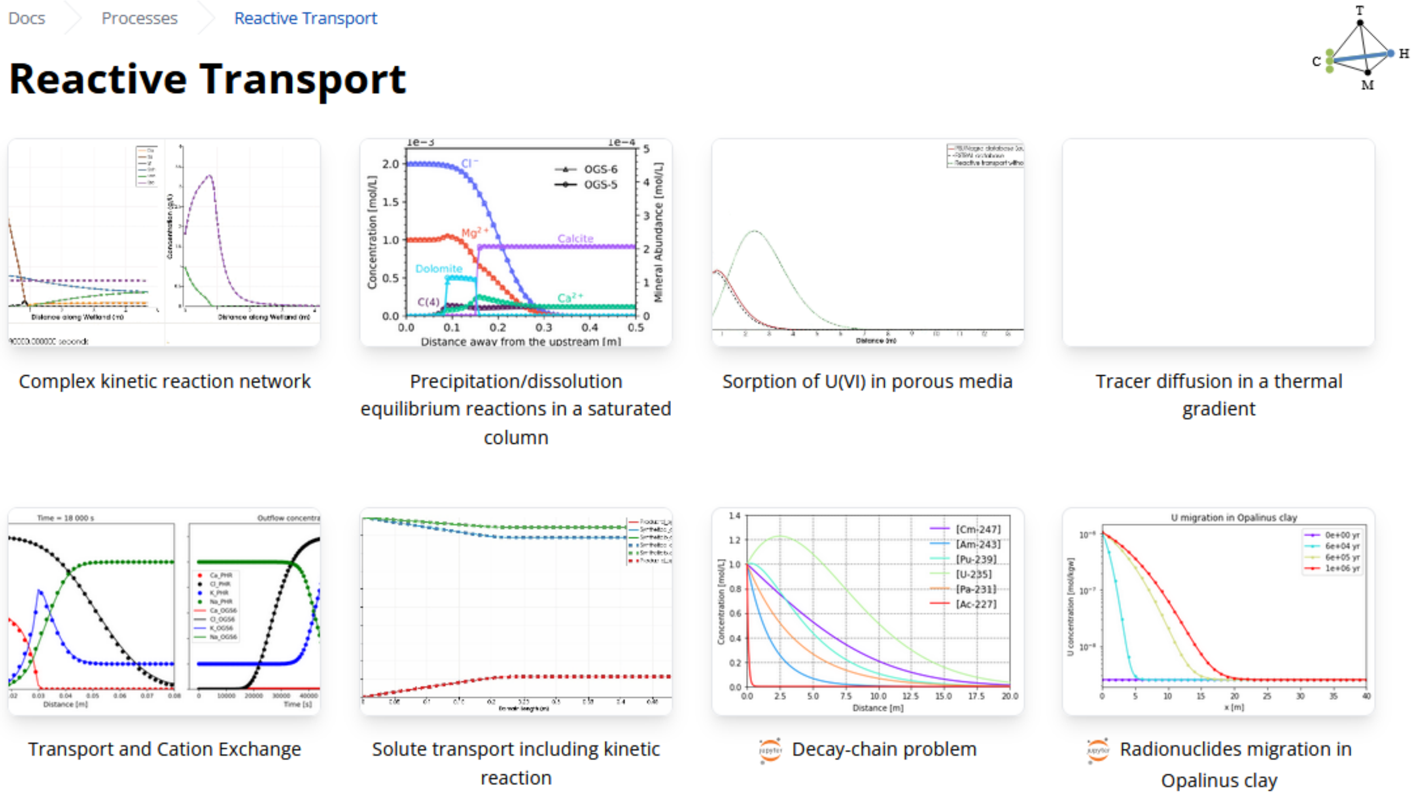
GIA - Global Implicit Approach

$$\begin{array}{|c|c|c|c|} \hline K_{c1c1} & K_{c1c2} & & \\ \hline & K_{c2c2} & K_{c2c3} & \\ \hline & & K_{c3c3} & K_{c3c4} \\ \hline & & & K_{c4c4} \\ \hline \end{array} \quad \begin{array}{|c|} \hline C_1 \\ \hline C_2 \\ \hline C_3 \\ \hline C_4 \\ \hline \end{array} = \begin{array}{|c|} \hline b_1 \\ \hline b_2 \\ \hline b_3 \\ \hline b_4 \\ \hline \end{array}$$

```
<chemical_system chemical solver = “SelfContained”>
    <chemical_reactions>
        <chemical_reaction>
            <!-- 0 = -1 [Cm-247] + 1 [Am-243] -->
            <stoichiometric_coefficients>-1 1 0 0 0 0</stoichiometric_coefficients>
            <!-- t1_half_life = 1.56e7;
                   LOG(2) / t1_half_life / 3.1536e7 -->
            <rate_constant>1.4089456993390242e-15</rate_constant>
        </chemical_reaction>
    </chemical_reactions>
</chemical_system>
```

Available benchmarks with Reactive Transport Process

Find all RTP benchmarks here, along with the descriptions: https://www.opengeosys.org/docs/benchmarks/reactive-transport/



Docs › Processes › Reactive Transport

Reactive Transport

Complex kinetic reaction network

Precipitation/dissolution equilibrium reactions in a saturated column

Sorption of U(VI) in porous media

Tracer diffusion in a thermal gradient

Transport and Cation Exchange

Solute transport including kinetic reaction

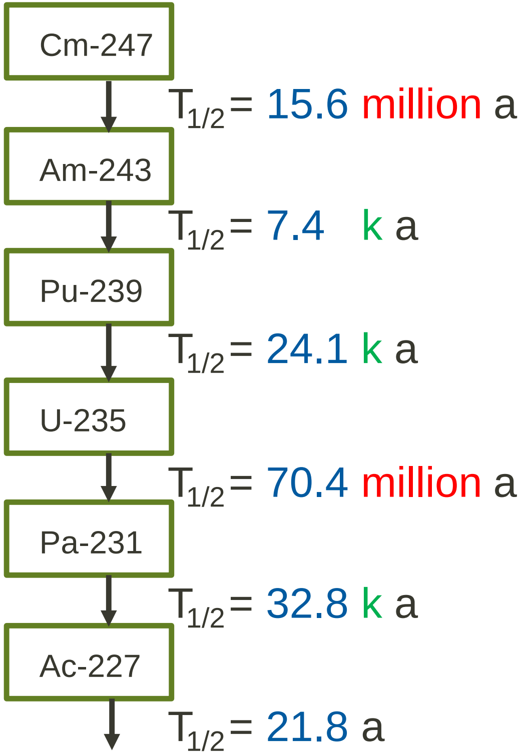
Decay-chain problem

Radionuclides migration in Opalinus clay

Overview

- Overview of Reactive Transport Process Features
- **The Decay-Chain Benchmark**
- Exercise: Set up the Decay-Chain prj File
- Simulation and Visualization

The Decay-Chain Benchmark



Cm-247

$T_{1/2}$= 15.6 million a

Am-243

$T_{1/2}$= 7.4 k a

Pu-239

$T_{1/2}$= 24.1 k a

U-235

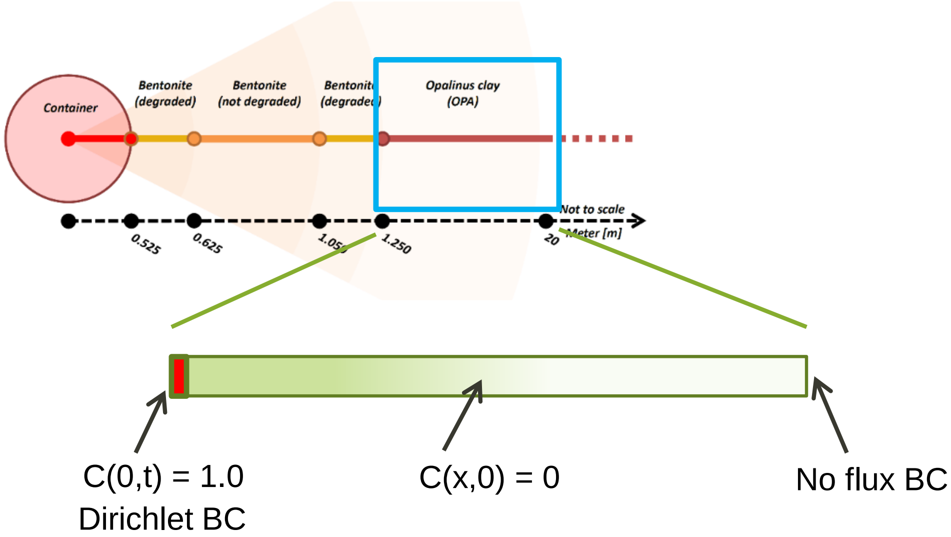
$T_{1/2}$= 70.4 million a

Pa-231

$T_{1/2}$= 32.8 k a

Ac-227

$T_{1/2}$= 21.8 a



C(0,t) = 1.0
Dirichlet BC

C(x,0) = 0

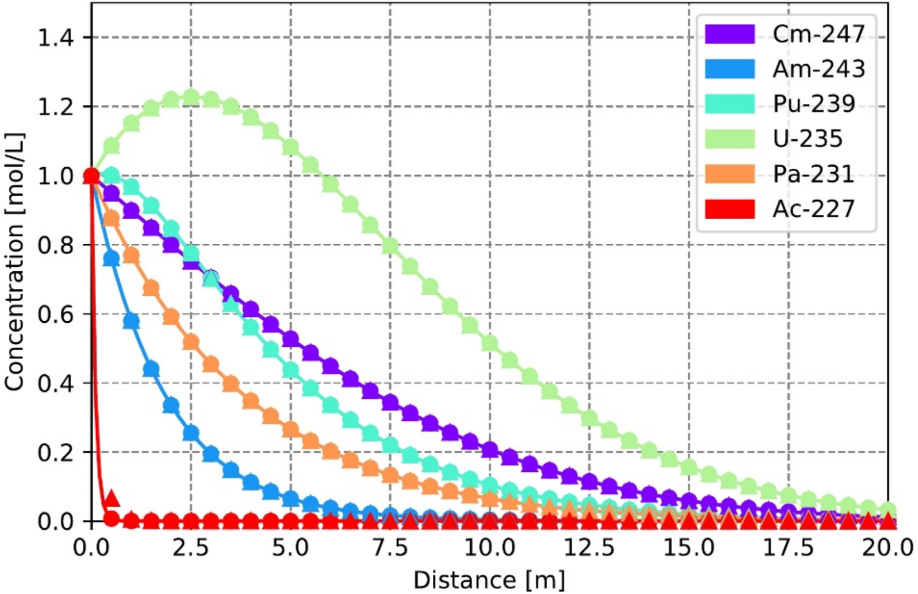
No flux BC

- 1D Domain [200 m × 10^6 y]
- Only diffusion + decay is considerred
- Uniform transport properties of OPA
  Porosity 0.12
  Pore diffusion coefficient 1e-11

Source: Dr. Christoph Behrens (BGE)

The Decay-Chain Benchmark

After 10k years, the simulated results as follows, also verified against the analytical solution



- In this case each component has the same pore diffusion coefficient, i.e some travels faster than others.
- U-235 is the slowest decaying component in this chain, therefore its concentration will accumulate over time and get more than 1.0
- Decay of Cm-247 is also slow, therefore it diffuses the second far
- Ac-227 is the fastest decaying component (only 23 years of half-life), therefore it can barely travel some distance

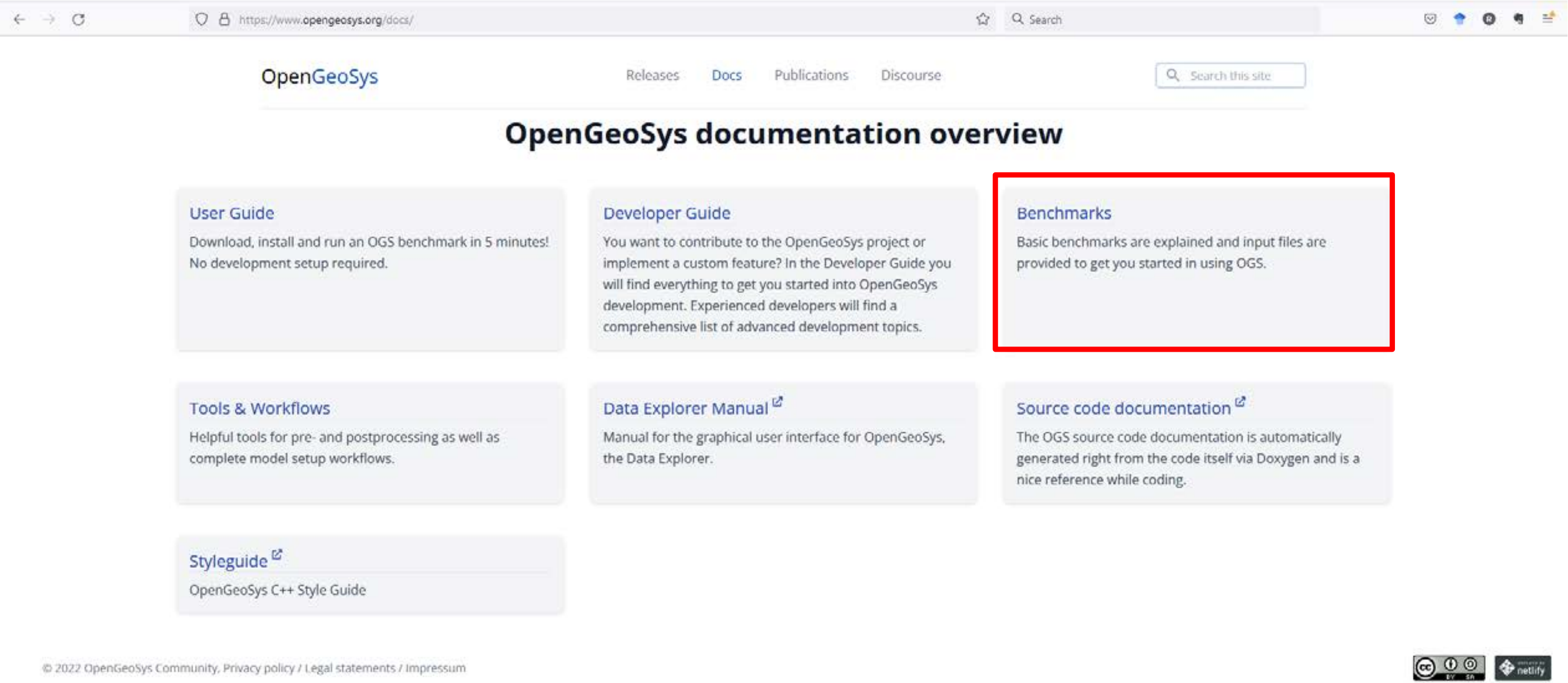
Analytical Solution: Sun, Y., Petersen, J. N., & Clement, T. P. (1999). *Journal of contaminant hydrology*, *35*(4), 429-440.

Overview

- Overview of Reactive Transport Process Features
- The Decay-Chain Benchmark
- **Exercise: Set up the Decay-Chain prj File**
- Simulation and Visualization

Exercise: Set up the Decay-Chain prj File

https://www.opengeosys.org/docs/



https://www.opengeosys.org/docs/

Search

OpenGeoSys

Releases Docs Publications Discourse

Search this site

OpenGeoSys documentation overview

User Guide
Download, install and run an OGS benchmark in 5 minutes! No development setup required.

Developer Guide
You want to contribute to the OpenGeoSys project or implement a custom feature? In the Developer Guide you will find everything to get you started into OpenGeoSys development. Experienced developers will find a comprehensive list of advanced development topics.

Benchmarks
Basic benchmarks are explained and input files are provided to get you started in using OGS.

Tools & Workflows
Helpful tools for pre- and postprocessing as well as complete model setup workflows.

Data Explorer Manual
Manual for the graphical user interface for OpenGeoSys, the Data Explorer.

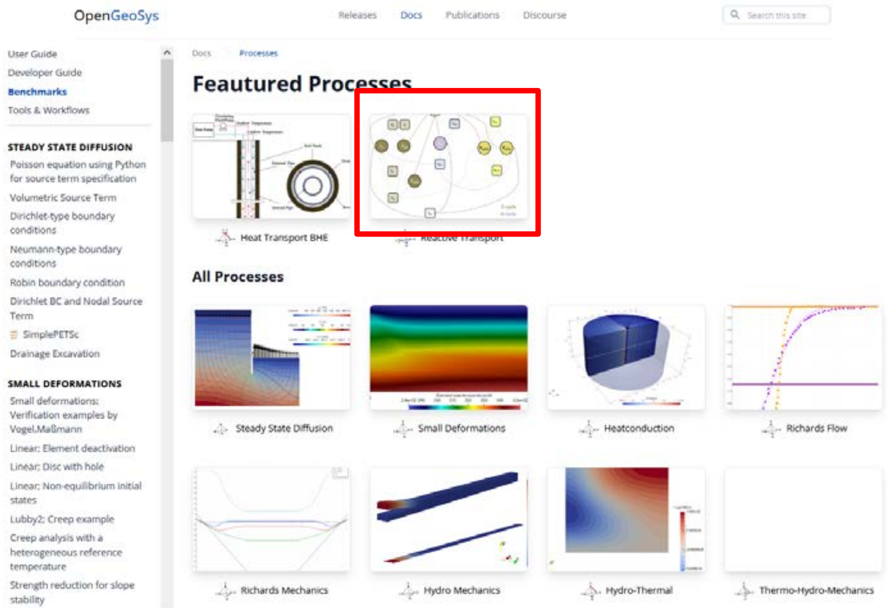
Source code documentation
The OGS source code documentation is automatically generated right from the code itself via Doxygen and is a nice reference while coding.

Styleguide
OpenGeoSys C++ Style Guide

© 2022 OpenGeoSys Community, Privacy policy / Legal statements / Impressum

Exercise: Set up the Decay-Chain prj File

https://www.opengeosys.org/docs/benchmarks/reactive-transport/decaychain/



OpenGeoSys

Releases Docs Publications Discourse

Search this site

User Guide
Developer Guide
Benchmarks
Tools & Workflows

STEADY STATE DIFFUSION

Poisson equation using Python for source term specification
Volumetric Source Term
Dirichlet-type boundary conditions
Neumann-type boundary conditions
Robin boundary condition
Dirichlet BC and Nodal Source Term
SimplePETSc
Drainage Excavation

SMALL DEFORMATIONS

Small deformations: Verification examples by Vogel.Maßmann
Linear: Element deactivation
Linear: Disc with hole
Linear: Non-equilibrium initial states
Lubby2: Creep example
Creep analysis with a heterogeneous reference temperature
Strength reduction for slope stability

Docs > Processes

Feautured Processes

Heat Transport BHE

Reactive Transport

All Processes

Steady State Diffusion

Small Deformations

Heatconduction

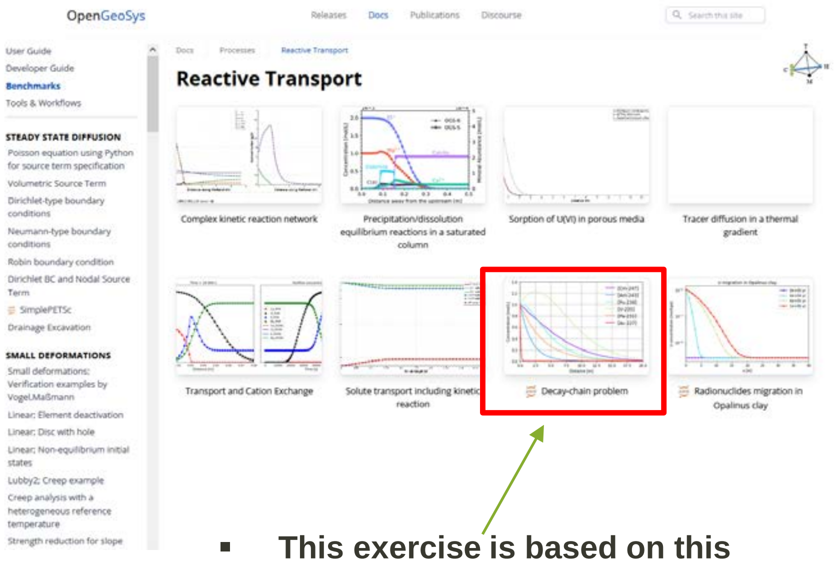
Richards Flow

Richards Mechanics

Hydro Mechanics

Hydro-Thermal

Thermo-Hydro-Mechanics



OpenGeoSys

Releases Docs Publications Discourse

Search this site

User Guide
Developer Guide
Benchmarks
Tools & Workflows

STEADY STATE DIFFUSION

Poisson equation using Python for source term specification
Volumetric Source Term
Dirichlet-type boundary conditions
Neumann-type boundary conditions
Robin boundary condition
Dirichlet BC and Nodal Source Term
SimplePETSc
Drainage Excavation

SMALL DEFORMATIONS

Small deformations: Verification examples by Vogel.Maßmann
Linear: Element deactivation
Linear: Disc with hole
Linear: Non-equilibrium initial states
Lubby2: Creep example
Creep analysis with a heterogeneous reference temperature
Strength reduction for slope

Docs > Processes > Reactive Transport

Reactive Transport

Complex kinetic reaction network

Precipitation/dissolution equilibrium reactions in a saturated column

Sorption of U(VI) in porous media

Tracer diffusion in a thermal gradient

Transport and Cation Exchange

Solute transport including kinetic reaction

Decay-chain problem

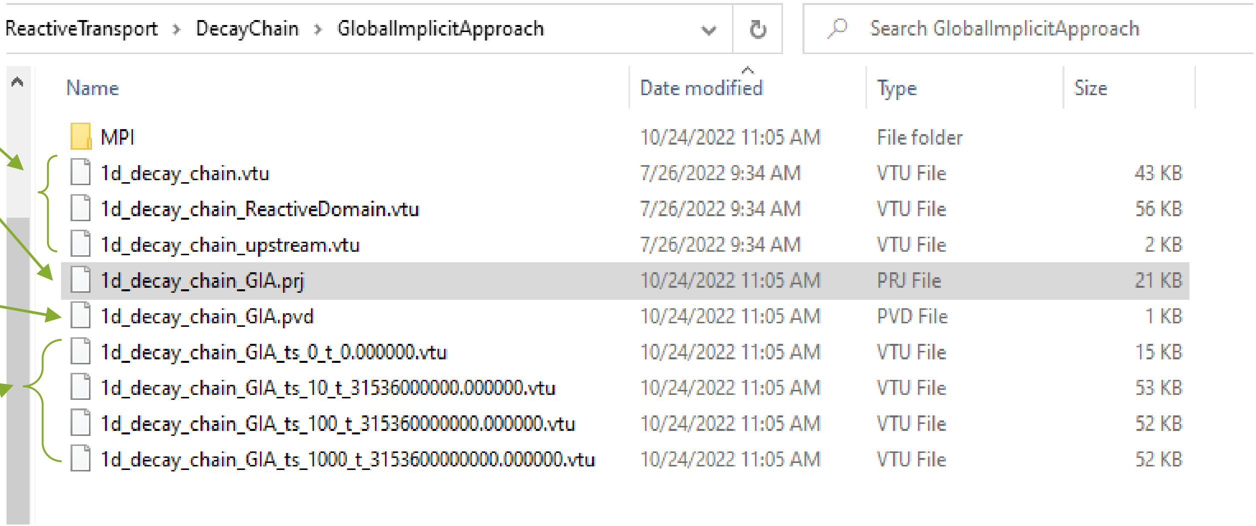
Radionuclides migration in Opalinus clay

- **This exercise is based on this benchmark**

Exercise: Set up the Decay-Chain prj File

In the source code folder
\Tests\Data\Parabolic\ComponentTransport\ReactiveTransport\DecayChain\GlobalImplicitApproach

- **3 mesh files with *.vtu**
- ***.prj project configuration file**
- ***.PVD is the Paraview index file referring to all result files**
- **4 *.vtu files containing the simulation results from time step #0, #10, #100 and #1000**

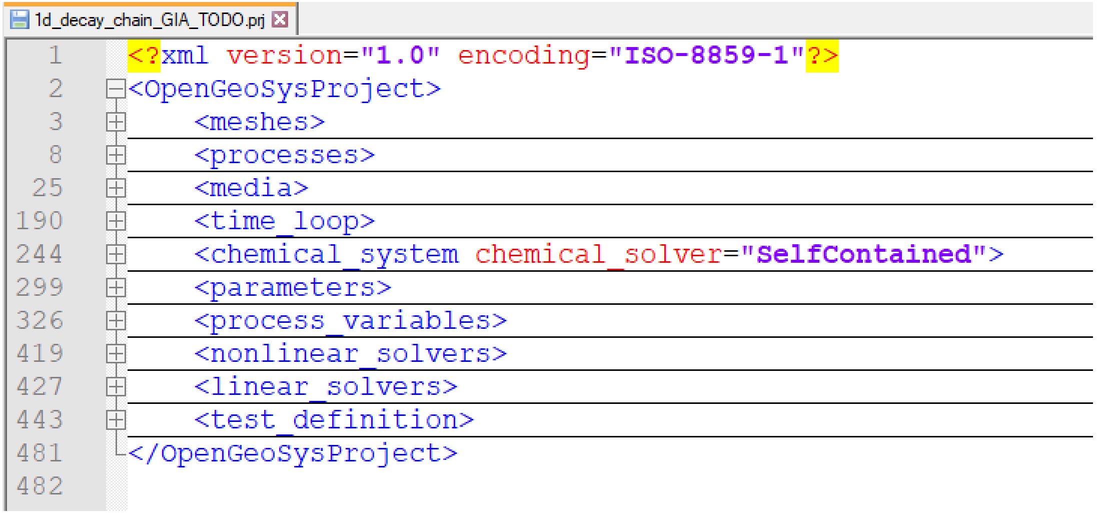


ReactiveTransport › DecayChain › GlobalImplicitApproach

Name	Date modified	Type	Size
MPI	10/24/2022 11:05 AM	File folder	
1d_decay_chain.vtu	7/26/2022 9:34 AM	VTU File	43 KB
1d_decay_chain_ReactiveDomain.vtu	7/26/2022 9:34 AM	VTU File	56 KB
1d_decay_chain_upstream.vtu	7/26/2022 9:34 AM	VTU File	2 KB
1d_decay_chain_GIA.prj	10/24/2022 11:05 AM	PRJ File	21 KB
1d_decay_chain_GIA.pvd	10/24/2022 11:05 AM	PVD File	1 KB
1d_decay_chain_GIA_ts_0_t_0.000000.vtu	10/24/2022 11:05 AM	VTU File	15 KB
1d_decay_chain_GIA_ts_10_t_31536000000.000000.vtu	10/24/2022 11:05 AM	VTU File	53 KB
1d_decay_chain_GIA_ts_100_t_315360000000.000000.vtu	10/24/2022 11:05 AM	VTU File	52 KB
1d_decay_chain_GIA_ts_1000_t_3153600000000.000000.vtu	10/24/2022 11:05 AM	VTU File	52 KB

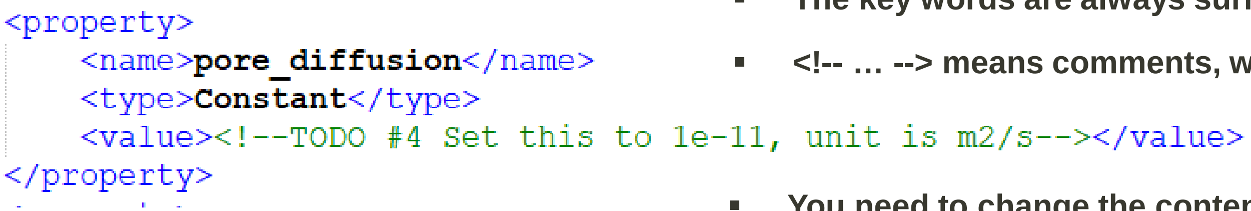
Exercise: Set up the Decay-Chain prj File

- The prj file is the model configuration file in XML format
- You can expand / collapse each section by clicking on the + and – symbol
- The order of the section does not matter
- When error pops up while reading the configuration, messages will be recorded in the log file or on screen



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OpenGeoSysProject>
    <meshes>
    <processes>
    <media>
    <time_loop>
    <chemical_system chemical_solver="SelfContained">
    <parameters>
    <process_variables>
    <nonlinear_solvers>
    <linear_solvers>
    <test_definition>
</OpenGeoSysProject>
```

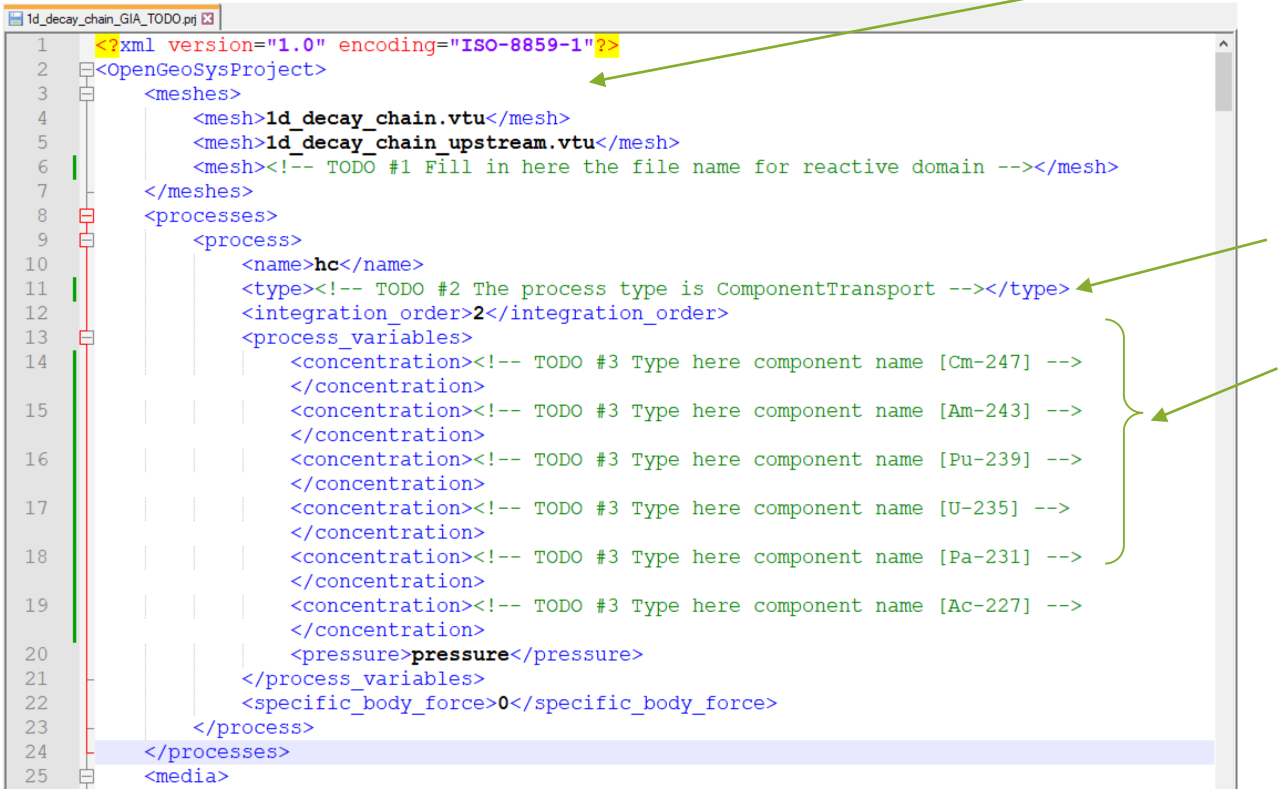
- The key words are always surrounded by the “Klammer” structure
- <!-- … --> means comments, which will not be read into the program



```
<property>
    <name>pore_diffusion</name>
    <type>Constant</type>
    <value><!--TODO #4 Set this to 1e-11, unit is m2/s--></value>
</property>
```

- You need to change the content when seeing the TODOs
- There are 10 TODOs in total

Exercise: Set up the Decay-Chain prj File :: Mesh and Processes



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<OpenGeoSysProject>
    <meshes>
        <mesh>1d_decay_chain.vtu</mesh>
        <mesh>1d_decay_chain_upstream.vtu</mesh>
        <mesh><!-- TODO #1 Fill in here the file name for reactive domain --></mesh>
    </meshes>
    <processes>
        <process>
            <name>hc</name>
            <type><!-- TODO #2 The process type is ComponentTransport --></type>
            <integration_order>2</integration_order>
            <process_variables>
                <concentration><!-- TODO #3 Type here component name [Cm-247] -->
                </concentration>
                <concentration><!-- TODO #3 Type here component name [Am-243] -->
                </concentration>
                <concentration><!-- TODO #3 Type here component name [Pu-239] -->
                </concentration>
                <concentration><!-- TODO #3 Type here component name [U-235] -->
                </concentration>
                <concentration><!-- TODO #3 Type here component name [Pa-231] -->
                </concentration>
                <concentration><!-- TODO #3 Type here component name [Ac-227] -->
                </concentration>
                <pressure>pressure</pressure>
            </process_variables>
            <specific_body_force>0</specific_body_force>
        </process>
    </processes>
    <media>
```

There are 3 mesh files here

- The entire domain
- Upstream boundary nodes
- The reaction domain (TODO#1)

We have to set the process type to

- ComponentTransport (TODO#2)

The process variables include the concentration of each component (TODO#3)

- [Cm-247]
- [Am-243]
- [Pu-239]
- [U-235]
- [Pa-231]
- [Ac-227]

Exercise: Set up the Decay-Chain prj File :: Media Properties

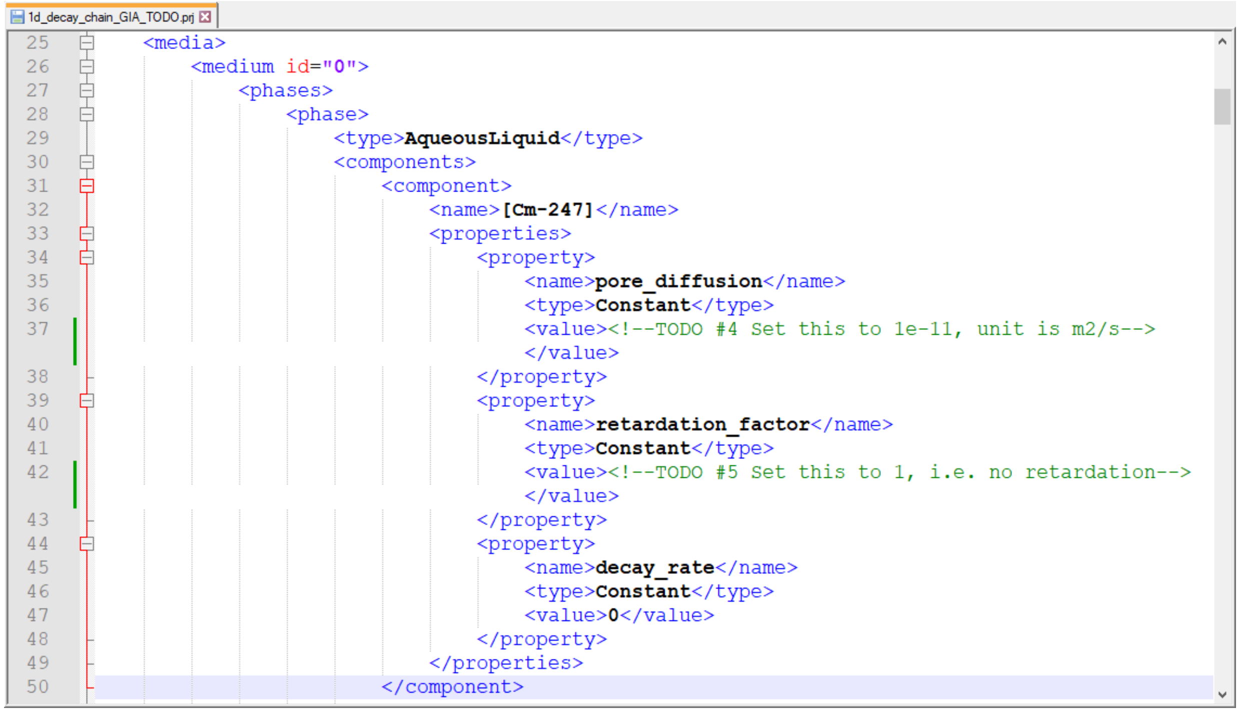
We devide the media into 3 phases

- AqueousLiquid
  (the only phase in DecayChain)
- Solid
- Gas

For each component in Aq. phase

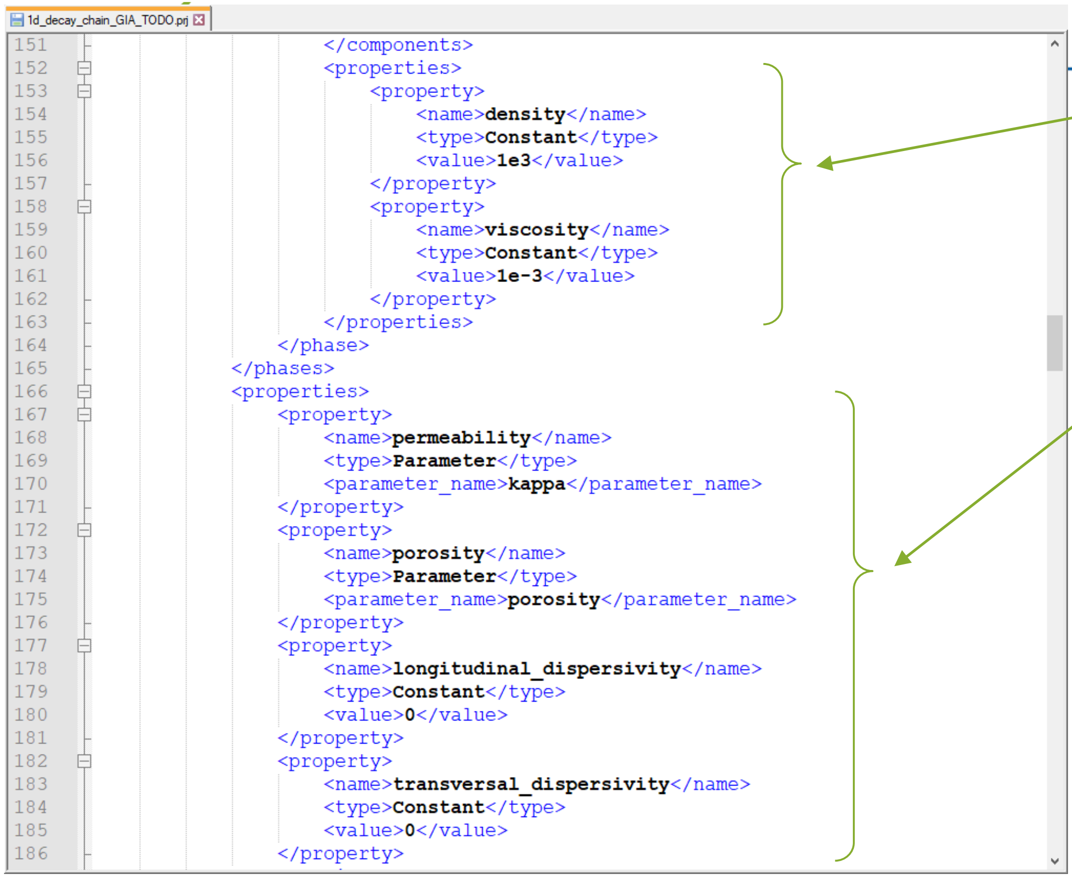
- Pore diffusion coefficient
- Retardation Factor set to 1 (no sorption)
- Decay Rate (set to 0 in this case)

Please repeat it for all components



```
    <media>
        <medium id="0">
            <phases>
                <phase>
                    <type>AqueousLiquid</type>
                    <components>
                        <component>
                            <name>[Cm-247]</name>
                            <properties>
                                <property>
                                    <name>pore_diffusion</name>
                                    <type>Constant</type>
                                    <value><!--TODO #4 Set this to 1e-11, unit is m2/s-->
                                    </value>
                                </property>
                                <property>
                                    <name>retardation_factor</name>
                                    <type>Constant</type>
                                    <value><!--TODO #5 Set this to 1, i.e. no retardation-->
                                    </value>
                                </property>
                                <property>
                                    <name>decay_rate</name>
                                    <type>Constant</type>
                                    <value>0</value>
                                </property>
                            </properties>
                        </component>
```

Exercise: Set up the Decay-Chain prj File :: Media Properties



```
                </components>
                <properties>
                    <property>
                        <name>density</name>
                        <type>Constant</type>
                        <value>1e3</value>
                    </property>
                    <property>
                        <name>viscosity</name>
                        <type>Constant</type>
                        <value>1e-3</value>
                    </property>
                </properties>
            </phase>
        </phases>
        <properties>
            <property>
                <name>permeability</name>
                <type>Parameter</type>
                <parameter_name>kappa</parameter_name>
            </property>
            <property>
                <name>porosity</name>
                <type>Parameter</type>
                <parameter_name>porosity</parameter_name>
            </property>
            <property>
                <name>longitudinal_dispersivity</name>
                <type>Constant</type>
                <value>0</value>
            </property>
            <property>
                <name>transversal_dispersivity</name>
                <type>Constant</type>
                <value>0</value>
            </property>
```

This section is the property of water

- Density (1000 kg/m3)
- Viscosity (0.001 Pa sec)

This section is about the porous media properties

- Permeability

  (refer to the kappa in the parameters list)
- Porosity

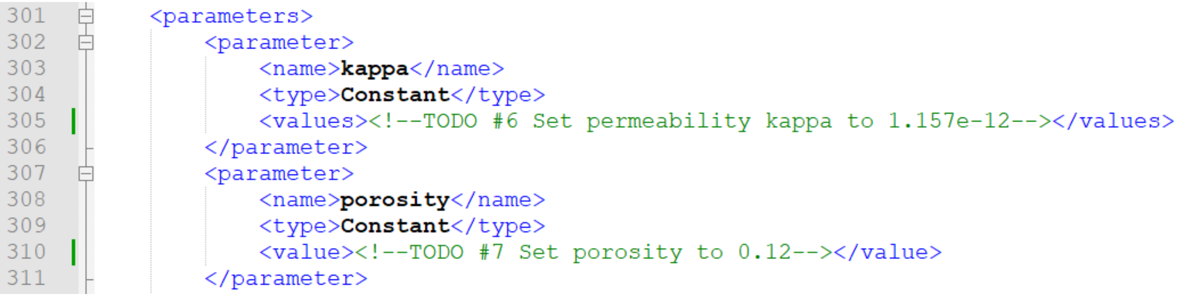
  (refer to the value in the parameters list)
- Longitudinal and transversal dispersivity

  (0 = no dispersion)

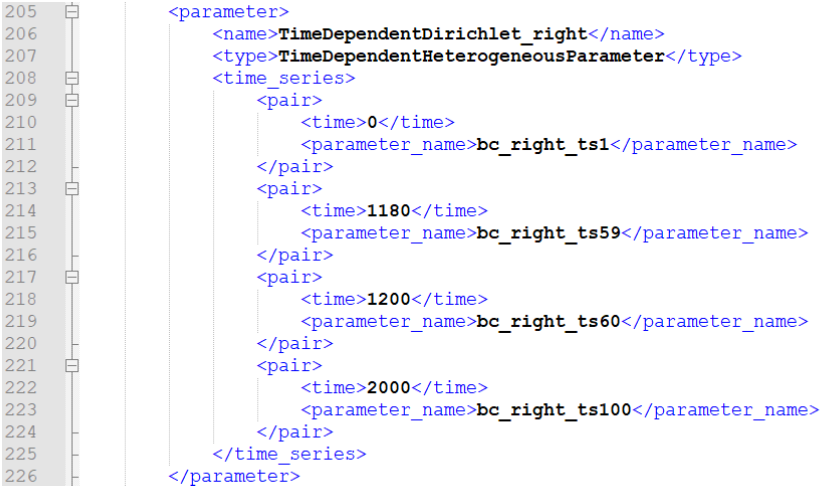
Exercise: Set up the parameters (TODO #6 and #7)

Give the following values in the parameters list (TODO #6 and #7)

- Set permeability to 1.157e-12 m2/s
- Set porosity to 0.12



```
<parameters>
    <parameter>
        <name>kappa</name>
        <type>Constant</type>
        <values><!--TODO #6 Set permeability kappa to 1.157e-12--></values>
    </parameter>
    <parameter>
        <name>porosity</name>
        <type>Constant</type>
        <value><!--TODO #7 Set porosity to 0.12--></value>
    </parameter>
```



```
<parameter>
    <name>TimeDependentDirichlet_right</name>
    <type>TimeDependentHeterogeneousParameter</type>
    <time_series>
        <pair>
            <time>0</time>
            <parameter_name>bc_right_ts1</parameter_name>
        </pair>
        <pair>
            <time>1180</time>
            <parameter_name>bc_right_ts59</parameter_name>
        </pair>
        <pair>
            <time>1200</time>
            <parameter_name>bc_right_ts60</parameter_name>
        </pair>
        <pair>
            <time>2000</time>
            <parameter_name>bc_right_ts100</parameter_name>
        </pair>
    </time_series>
</parameter>
```

A parameter can be defined as

- A constant value
- Time dependent
- Space dependent

https://www.opengeosys.org/docs/benchmarks/liquid-flow/time-dependent-heterogeneous-source-term-and-boundary-conditions/

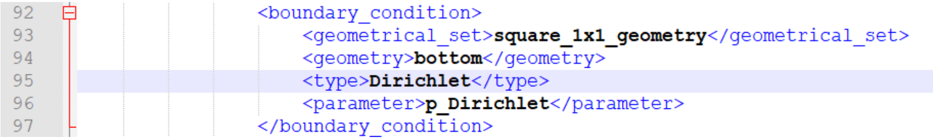
Exercise: Set up the Decay-Chain prj File :: Types of Boundary Conditions

For solving the elliptic problem $k\,\Delta h = 0 \quad \text{in } \Omega$

https://www.opengeosys.org/docs/benchmarks/elliptic/elliptic-neumann/

- **Dirichlet → Given Value of Primary Variable**

$$h = g_D \quad \text{on } \Gamma_D,$$

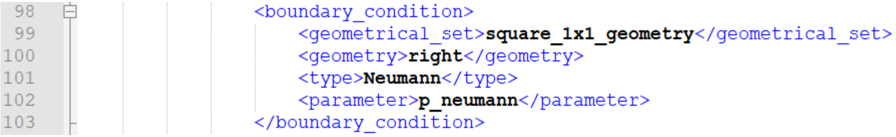


```xml
<boundary_condition>
    <geometrical_set>square_1x1_geometry</geometrical_set>
    <geometry>bottom</geometry>
    <type>Dirichlet</type>
    <parameter>p_Dirichlet</parameter>
</boundary_condition>
```

https://www.opengeosys.org/docs/benchmarks/elliptic/elliptic-neumann/

- **Neumann → Given Flux Value**

$$k\frac{\partial h}{\partial n} = g_N \quad \text{on } \Gamma_N,$$

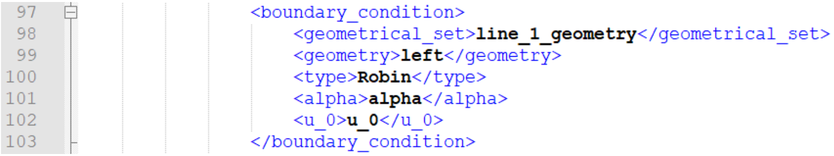


```xml
<boundary_condition>
    <geometrical_set>square_1x1_geometry</geometrical_set>
    <geometry>right</geometry>
    <type>Neumann</type>
    <parameter>p_neumann</parameter>
</boundary_condition>
```

https://www.opengeosys.org/docs/benchmarks/elliptic/elliptic-robin/

- **Robin → Given Flux calculated by Primary Variable**

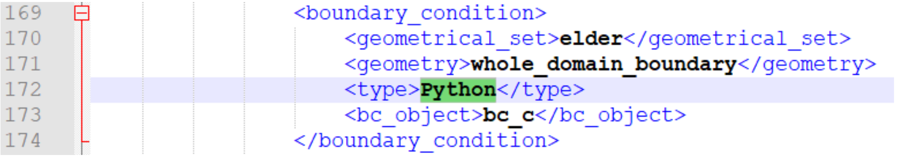
$$\frac{\partial h}{\partial n} = \alpha(h_0 - h(x)) \quad \text{on } \Gamma_R,$$



```xml
<boundary_condition>
    <geometrical_set>line_1_geometry</geometrical_set>
    <geometry>left</geometry>
    <type>Robin</type>
    <alpha>alpha</alpha>
    <u_0>u_0</u_0>
</boundary_condition>
```

https://www.opengeosys.org/docs/benchmarks/python-bc/elder/

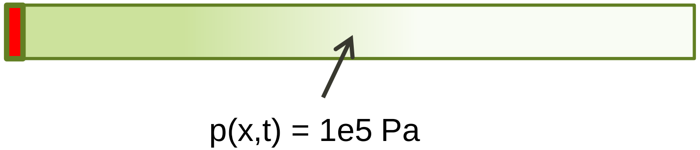
- **More complex BC via Python script**

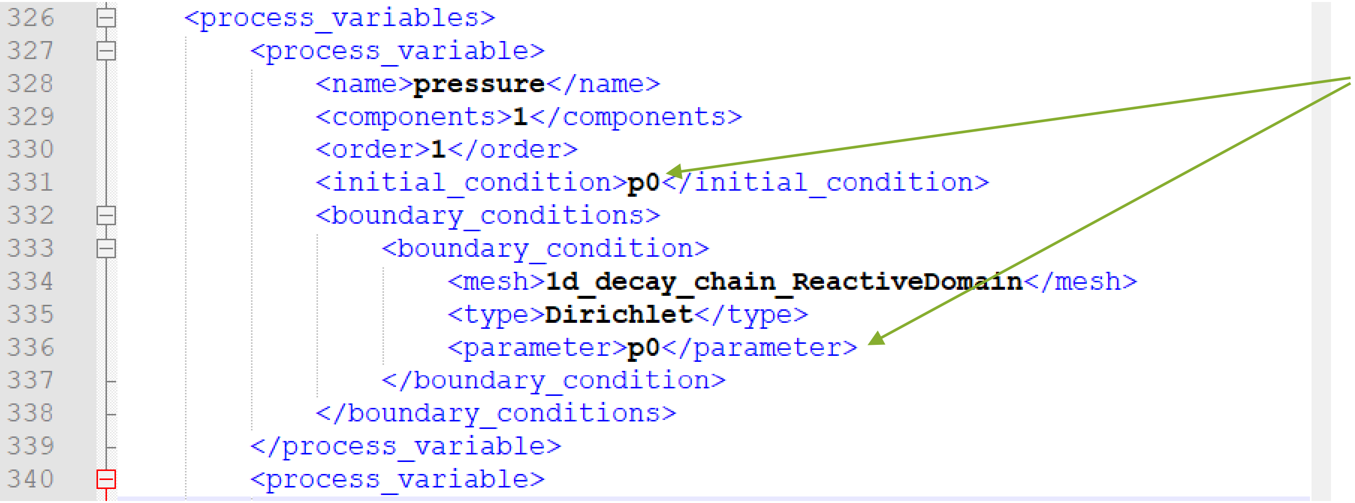


```xml
<boundary_condition>
    <geometrical_set>elder</geometrical_set>
    <geometry>whole_domain_boundary</geometry>
    <type>Python</type>
    <bc_object>bc_c</bc_object>
</boundary_condition>
```

Exercise: Set up the Decay-Chain prj File :: Initial and Boundary Conditions

For pressure, We set both initial and boundary conditions to 1 bar (1e5 Pa),



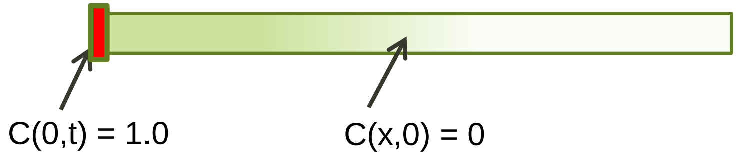


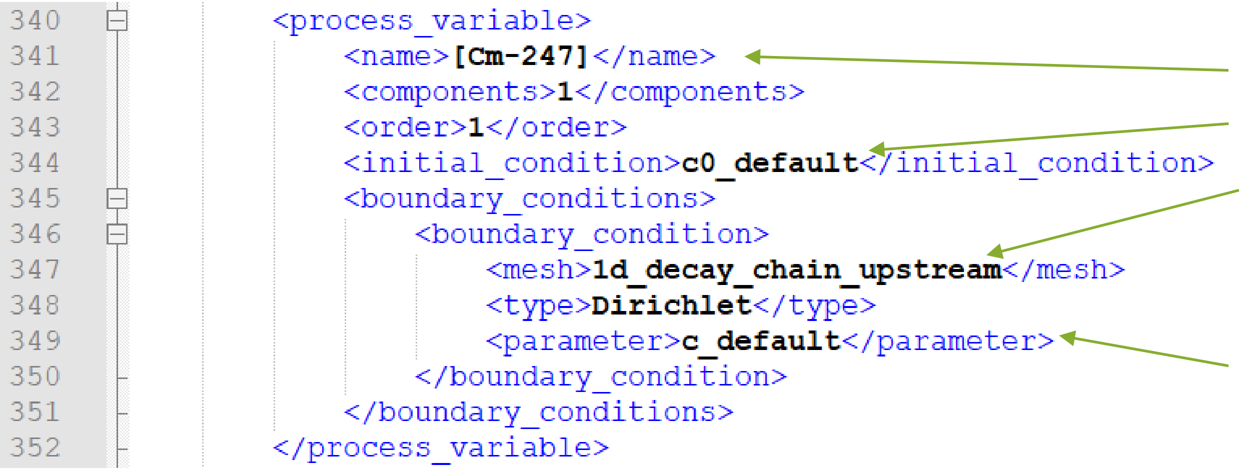
```
    <process_variables>
        <process_variable>
            <name>pressure</name>
            <components>1</components>
            <order>1</order>
            <initial_condition>p0</initial_condition>
            <boundary_conditions>
                <boundary_condition>
                    <mesh>1d_decay_chain_ReactiveDomain</mesh>
                    <type>Dirichlet</type>
                    <parameter>p0</parameter>
                </boundary_condition>
            </boundary_conditions>
        </process_variable>
        <process_variable>
```

- Since there is no difference in pressure, there will be no advection. Hence, the transport is only controlled by diffusion and decay.

Exercise: Set up the Decay-Chain prj File :: Initial and Boundary Conditions

For concentration, we set initial condition to 0 and boundary condition to 1,





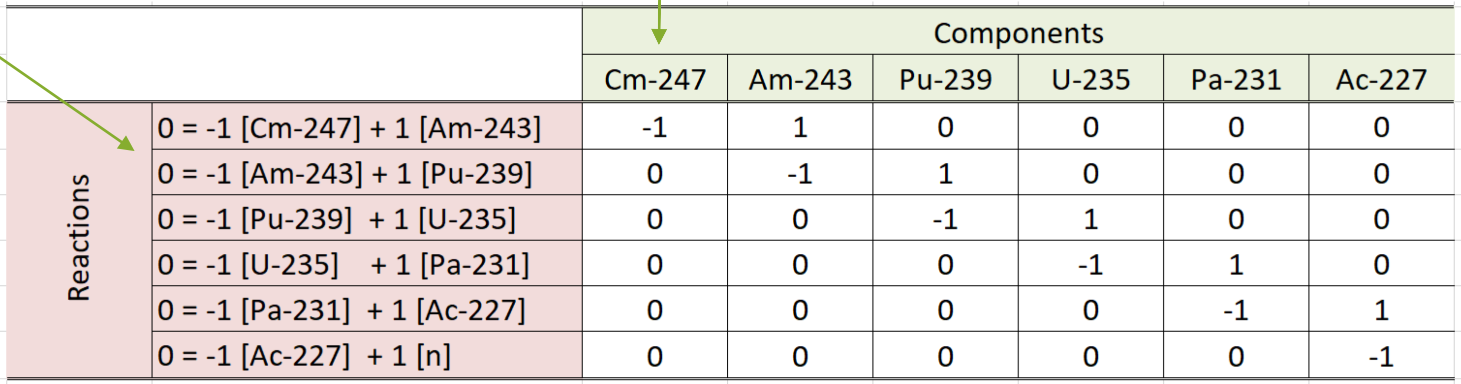
```xml
<process_variable>
    <name>[Cm-247]</name>
    <components>1</components>
    <order>1</order>
    <initial_condition>c0_default</initial_condition>
    <boundary_conditions>
        <boundary_condition>
            <mesh>1d_decay_chain_upstream</mesh>
            <type>Dirichlet</type>
            <parameter>c_default</parameter>
        </boundary_condition>
    </boundary_conditions>
</process_variable>
```

- Values set for each component
- Initial condition set to 0
- Boundary condition applied on a subset of the domain (just the boundary node in this case)
- Boundary condition set to 1

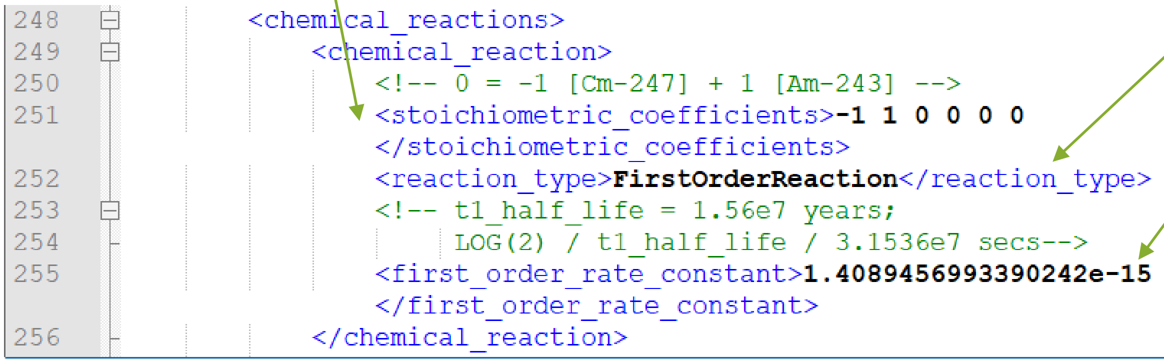
Exercise: Set up the Decay-Chain prj File :: Decay Reactions

- **The stoichiometry is filling in following each reactions**

- **The order of participating components are following the order of process variables**



		Components					
		Cm-247	Am-243	Pu-239	U-235	Pa-231	Ac-227
Reactions	0 = -1 [Cm-247] + 1 [Am-243]	-1	1	0	0	0	0
	0 = -1 [Am-243] + 1 [Pu-239]	0	-1	1	0	0	0
	0 = -1 [Pu-239] + 1 [U-235]	0	0	-1	1	0	0
	0 = -1 [U-235] + 1 [Pa-231]	0	0	0	-1	1	0
	0 = -1 [Pa-231] + 1 [Ac-227]	0	0	0	0	-1	1
	0 = -1 [Ac-227] + 1 [n]	0	0	0	0	0	-1



```
        <chemical_reactions>
            <chemical_reaction>
                <!-- 0 = -1 [Cm-247] + 1 [Am-243] -->
                <stoichiometric_coefficients>-1 1 0 0 0 0 </stoichiometric_coefficients>
                <reaction_type>FirstOrderReaction</reaction_type>
                <!-- t1_half_life = 1.56e7 years;
                     LOG(2) / t1_half_life / 3.1536e7 secs-->
                <first_order_rate_constant>1.4089456993390242e-15 </first_order_rate_constant>
            </chemical_reaction>
```

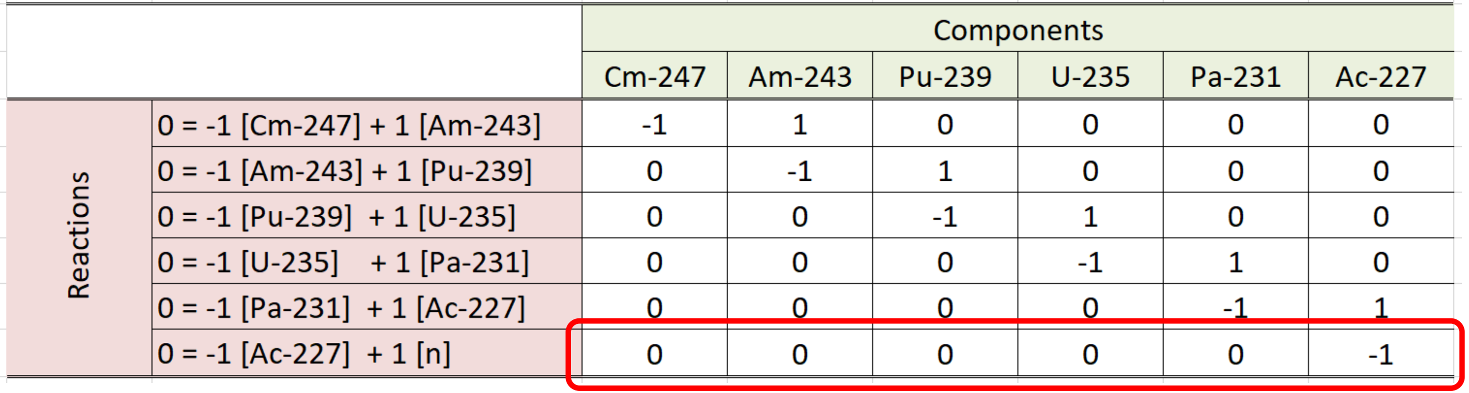
- **Rate of reaction is first order type**

$$Rate = \frac{d[C]}{dt} = k[C]$$

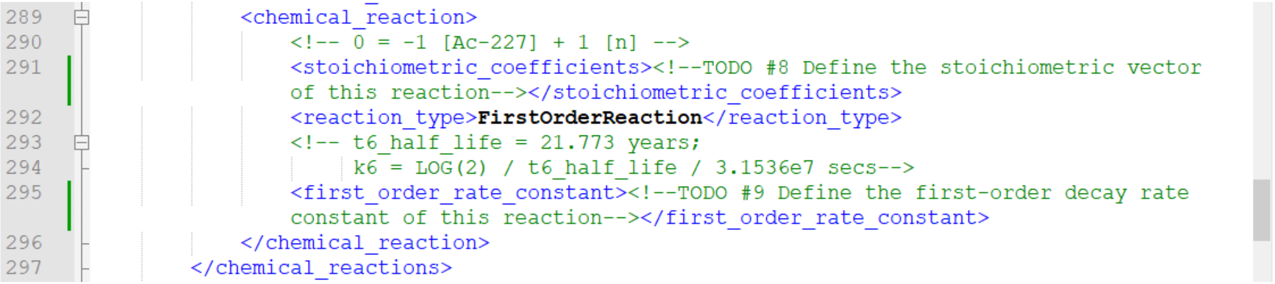
- **Rate constant is according to the half-life of this radionuclide**

Exercise: Set up the Decay-Chain prj File :: Decay Reactions (TODO #8 and TODO #9)

- What is the stoichiometric coefficients of the last reaction?
- How much is the reaction rate constant for this reaction?

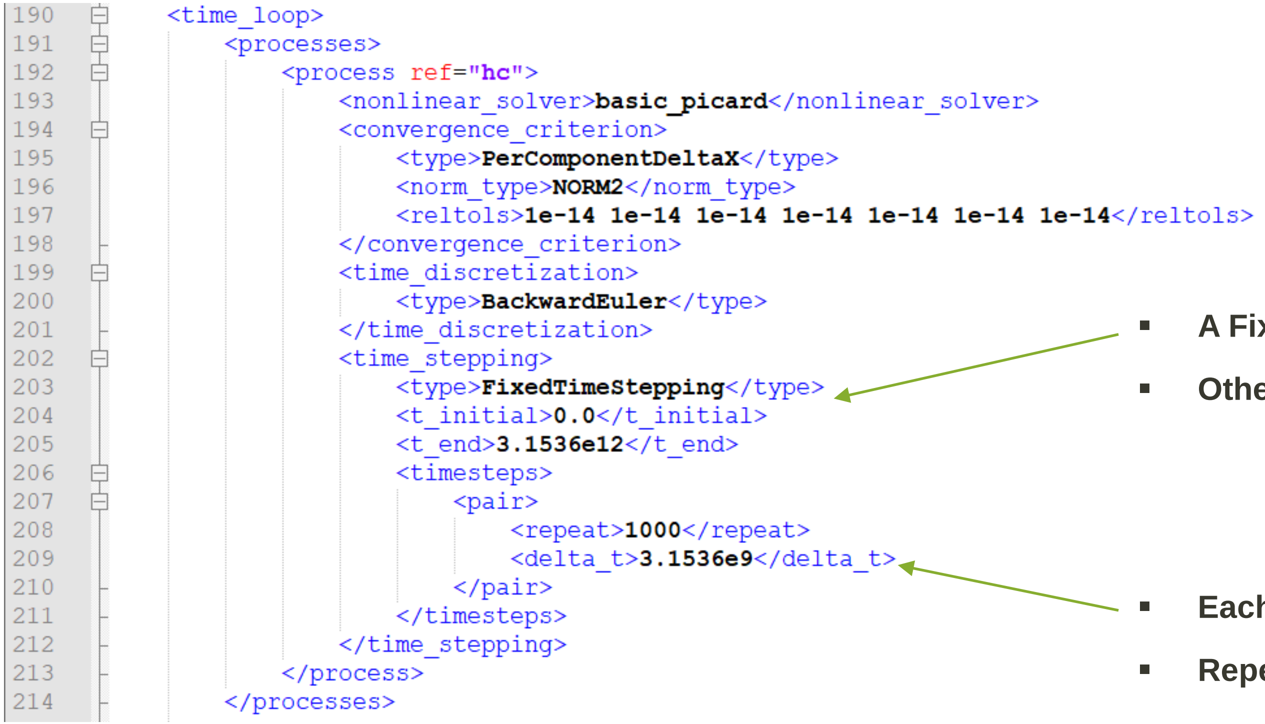


		Components					
		Cm-247	Am-243	Pu-239	U-235	Pa-231	Ac-227
Reactions	0 = -1 [Cm-247] + 1 [Am-243]	-1	1	0	0	0	0
	0 = -1 [Am-243] + 1 [Pu-239]	0	-1	1	0	0	0
	0 = -1 [Pu-239] + 1 [U-235]	0	0	-1	1	0	0
	0 = -1 [U-235] + 1 [Pa-231]	0	0	0	-1	1	0
	0 = -1 [Pa-231] + 1 [Ac-227]	0	0	0	0	-1	1
	0 = -1 [Ac-227] + 1 [n]	0	0	0	0	0	-1



```
    <chemical_reaction>
        <!-- 0 = -1 [Ac-227] + 1 [n] -->
        <stoichiometric_coefficients><!--TODO #8 Define the stoichiometric vector
        of this reaction--></stoichiometric_coefficients>
        <reaction_type>FirstOrderReaction</reaction_type>
        <!-- t6_half_life = 21.773 years;
             k6 = LOG(2) / t6_half_life / 3.1536e7 secs-->
        <first_order_rate_constant><!--TODO #9 Define the first-order decay rate
        constant of this reaction--></first_order_rate_constant>
    </chemical_reaction>
</chemical_reactions>
```

Exercise: Set up the Decay-Chain prj File :: Time Stepping Scheme



```
    <time_loop>
        <processes>
            <process ref="hc">
                <nonlinear_solver>basic_picard</nonlinear_solver>
                <convergence_criterion>
                    <type>PerComponentDeltaX</type>
                    <norm_type>NORM2</norm_type>
                    <reltols>1e-14 1e-14 1e-14 1e-14 1e-14 1e-14 1e-14</reltols>
                </convergence_criterion>
                <time_discretization>
                    <type>BackwardEuler</type>
                </time_discretization>
                <time_stepping>
                    <type>FixedTimeStepping</type>
                    <t_initial>0.0</t_initial>
                    <t_end>3.1536e12</t_end>
                    <timesteps>
                        <pair>
                            <repeat>1000</repeat>
                            <delta_t>3.1536e9</delta_t>
                        </pair>
                    </timesteps>
                </time_stepping>
            </process>
        </processes>
```

- A FixedTimeStepping scheme is set here
- Other stepping schemes are:

- Each step is set to be 100 years
- Repeating 1000 times gives 100 k years

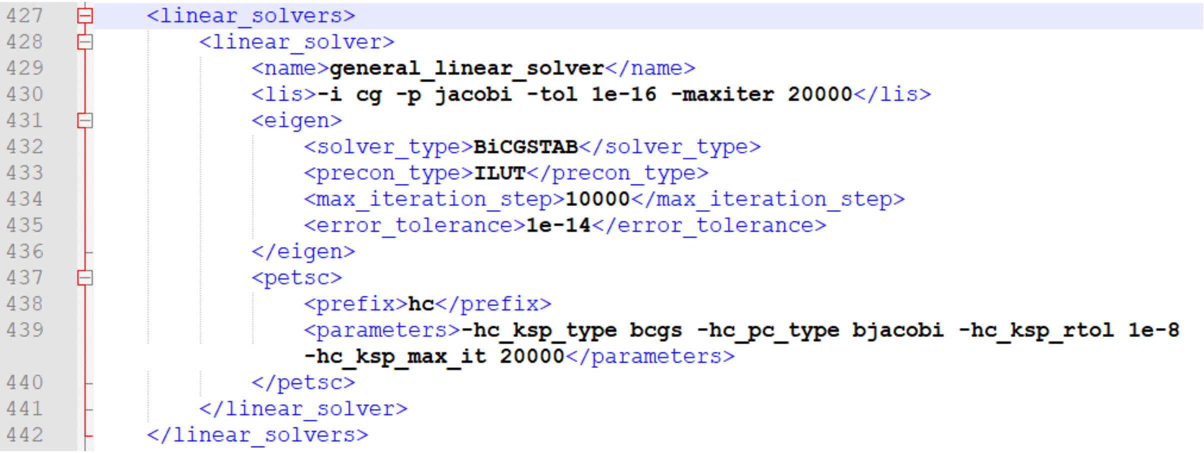
Exercise: Linear Solver

The linear solver is internally calling one of the following external lib:

- LIS solver
- Eigen solve (default)
- PETSC solver

The following sparse linear solvers in Eigen library has been included:

- CG
- BiCGSTAB
- IDRS
- IDRSTABL
- SparseLU
- PardisoLU
- GMRES

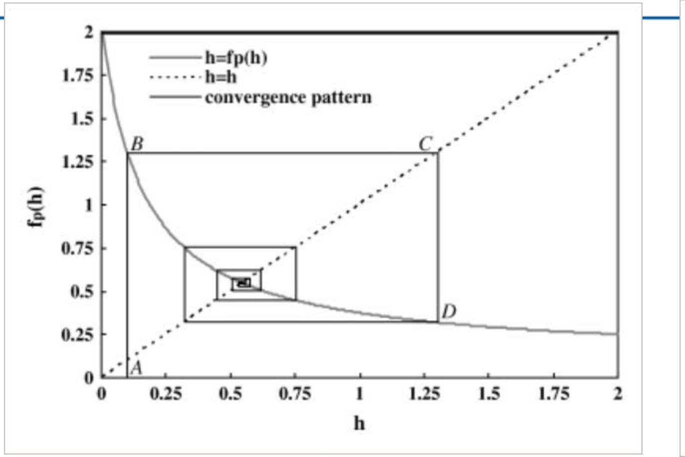


```xml
<linear_solvers>
    <linear_solver>
        <name>general_linear_solver</name>
        <lis>-i cg -p jacobi -tol 1e-16 -maxiter 20000</lis>
        <eigen>
            <solver_type>BiCGSTAB</solver_type>
            <precon_type>ILUT</precon_type>
            <max_iteration_step>10000</max_iteration_step>
            <error_tolerance>1e-14</error_tolerance>
        </eigen>
        <petsc>
            <prefix>hc</prefix>
            <parameters>-hc_ksp_type bcgs -hc_pc_type bjacobi -hc_ksp_rtol 1e-8
            -hc_ksp_max_it 20000</parameters>
        </petsc>
    </linear_solver>
</linear_solvers>
```

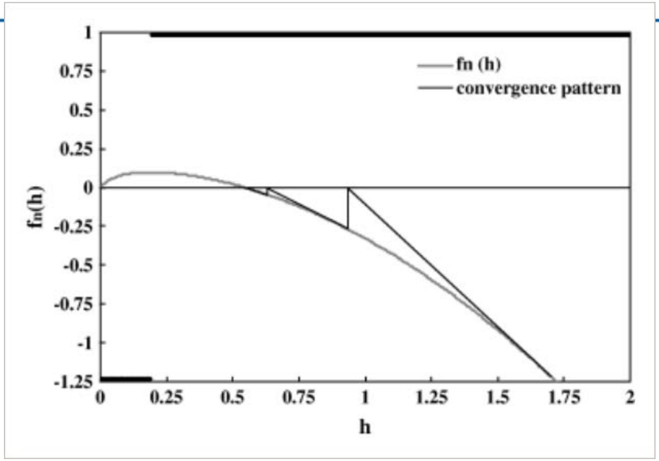
- In Eigen solver, one can choose different solver type and also preconditioners
- PETSC solver is prepared for parallel computing with MPI library (distributed memory type)
- Another choice of linear solve is the PADISO solver provided by Intel MKL lib

Exercise: Non-Linear Solvers

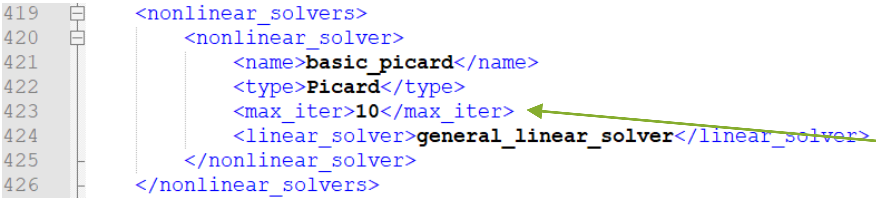
PICARD



NEWTON

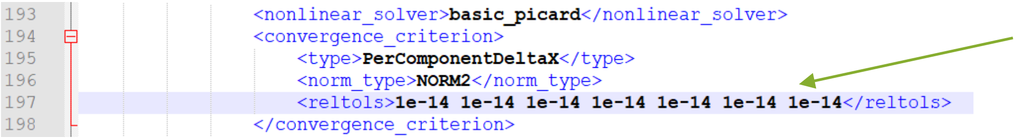


Source: Mehl (2006) doi: 10.1111/j.1745-6584.2006.00207.x



```
<nonlinear_solvers>
    <nonlinear_solver>
        <name>basic_picard</name>
        <type>Picard</type>
        <max_iter>10</max_iter>
        <linear_solver>general_linear_solver</linear_solver>
    </nonlinear_solver>
</nonlinear_solvers>
```

- The maximum num of iterations is set to 10

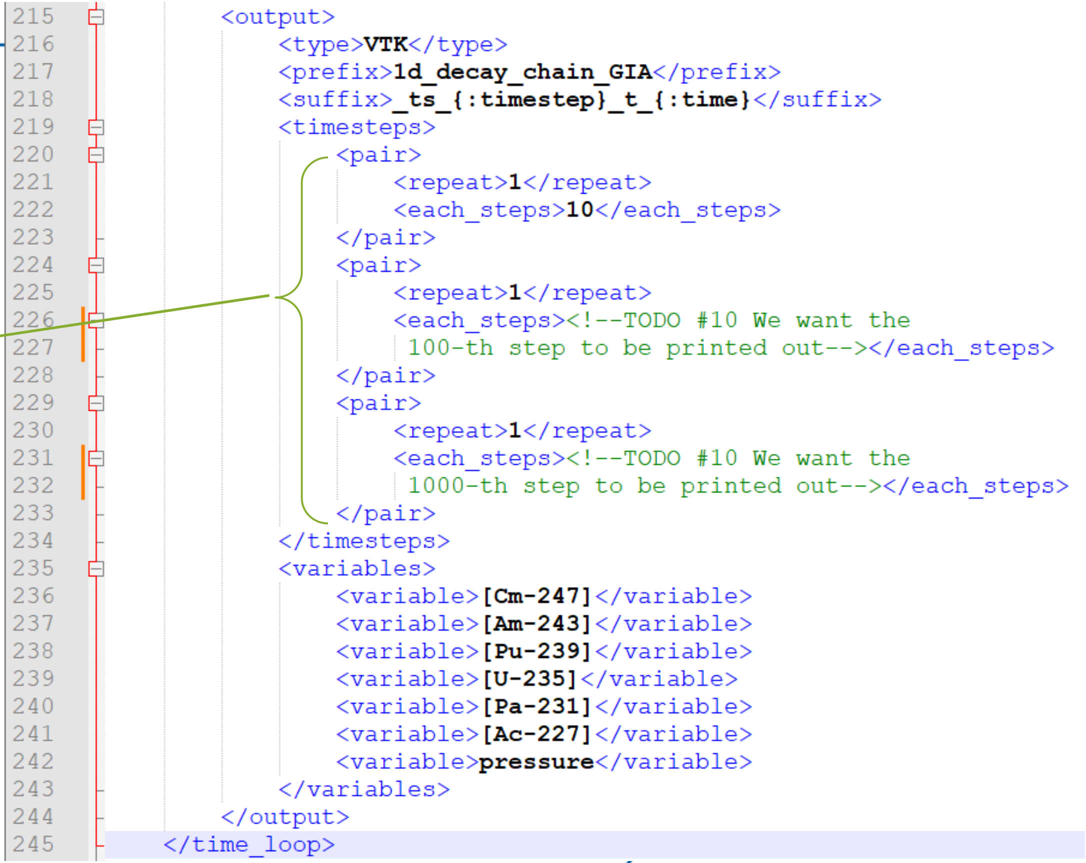


```
            <nonlinear_solver>basic_picard</nonlinear_solver>
            <convergence_criterion>
                <type>PerComponentDeltaX</type>
                <norm_type>NORM2</norm_type>
                <reltols>1e-14 1e-14 1e-14 1e-14 1e-14 1e-14 1e-14</reltols>
            </convergence_criterion>
```

- This is the relative tolerance for each component

Exercise: Set up the Decay-Chain prj File :: Output Control (TODO #10)

- The output file will always start with this prefix
- Followed by the time step and time info
- Here we define when we want to output
  - T = 0 is always printed out (initial condition, 0th step)
  - Then output after 10 steps (10th step)
  - Then after 90 steps (100th step)
  - Then after 900 steps (1000th step)
- Under the keyword <variables>, we specify all variables that will appear in the vtu result files

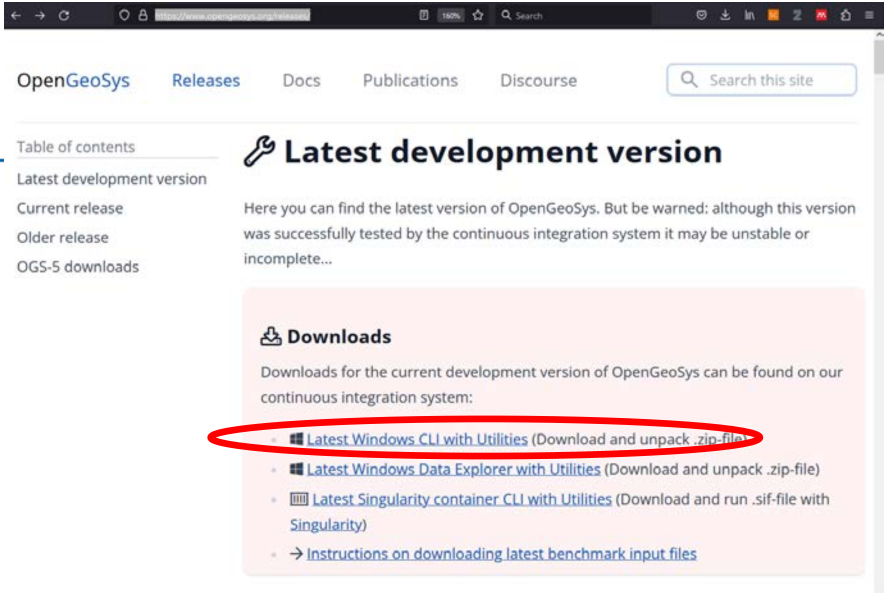


```
        <output>
            <type>VTK</type>
            <prefix>1d_decay_chain_GIA</prefix>
            <suffix>_ts_{:timestep}_t_{:time}</suffix>
            <timesteps>
                <pair>
                    <repeat>1</repeat>
                    <each_steps>10</each_steps>
                </pair>
                <pair>
                    <repeat>1</repeat>
                    <each_steps><!--TODO #10 We want the
                     100-th step to be printed out--></each_steps>
                </pair>
                <pair>
                    <repeat>1</repeat>
                    <each_steps><!--TODO #10 We want the
                     1000-th step to be printed out--></each_steps>
                </pair>
            </timesteps>
            <variables>
                <variable>[Cm-247]</variable>
                <variable>[Am-243]</variable>
                <variable>[Pu-239]</variable>
                <variable>[U-235]</variable>
                <variable>[Pa-231]</variable>
                <variable>[Ac-227]</variable>
                <variable>pressure</variable>
            </variables>
        </output>
    </time_loop>
```

Overview

- Overview of Reactive Transport Process Features
- The Decay-Chain Benchmark
- Exercise: Set up the Decay-Chain prj File
- **Simulation and Visualization**

Different Approaches of Simulation



Latest development version

Here you can find the latest version of OpenGeoSys. But be warned: although this version was successfully tested by the continuous integration system it may be unstable or incomplete…

Downloads

Downloads for the current development version of OpenGeoSys can be found on our continuous integration system:

- Latest Windows CLI with Utilities (Download and unpack .zip-file)
- Latest Windows Data Explorer with Utilities (Download and unpack .zip-file)
- Latest Singularity container CLI with Utilities (Download and run .sif-file with Singularity)
- → Instructions on downloading latest benchmark input files

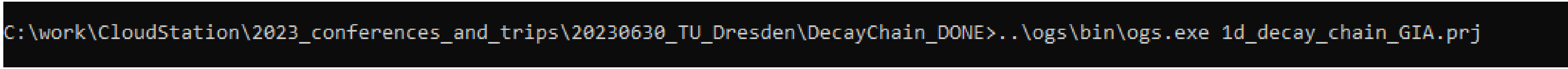
- **Download OGS software package from the web**

  https://www.opengeosys.org/releases/

- **Unzip the package into a folder called “ogs”**

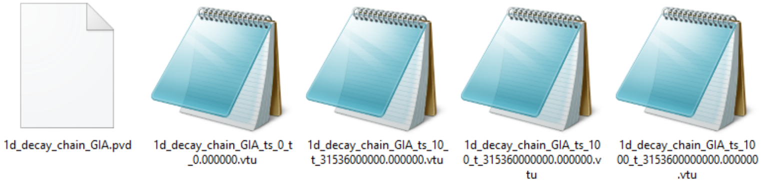
- **Run simulation**

  Syntax: ogs.exe   path_to_the_project_file



```
C:\work\CloudStation\2023_conferences_and_trips\20230630_TU_Dresden\DecayChain_DONE>..\ogs\bin\ogs.exe 1d_decay_chain_GIA.prj
```

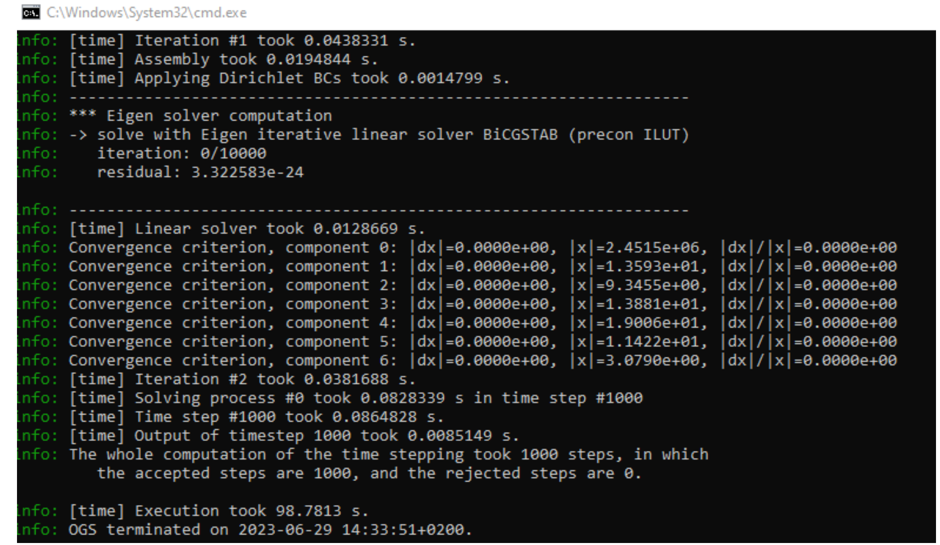
- **When simulation starts, time-series data files will be generated (.pvd file + .vtu files)**



1d_decay_chain_GIA.pvd
1d_decay_chain_GIA_ts_0_t_0.000000.vtu
1d_decay_chain_GIA_ts_10_t_31536000000.000000.vtu
1d_decay_chain_GIA_ts_100_t_315360000000.000000.vtu
1d_decay_chain_GIA_ts_1000_t_3153600000000.000000.vtu

Different Approaches of Simulation

- Display output messages

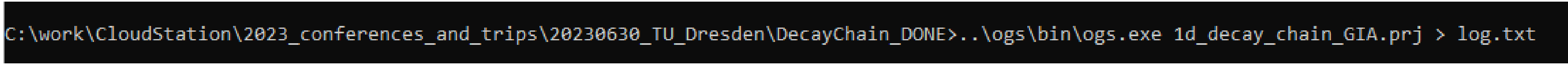


```
info: [time] Iteration #1 took 0.0438331 s.
info: [time] Assembly took 0.0194844 s.
info: [time] Applying Dirichlet BCs took 0.0014799 s.
info: ------------------------------------------------------------------
info: *** Eigen solver computation
info: -> solve with Eigen iterative linear solver BiCGSTAB (precon ILUT)
info:    iteration: 0/10000
info:    residual: 3.322583e-24

info: ------------------------------------------------------------------
info: [time] Linear solver took 0.0128669 s.
info: Convergence criterion, component 0: |dx|=0.0000e+00, |x|=2.4515e+06, |dx|/|x|=0.0000e+00
info: Convergence criterion, component 1: |dx|=0.0000e+00, |x|=1.3593e+01, |dx|/|x|=0.0000e+00
info: Convergence criterion, component 2: |dx|=0.0000e+00, |x|=9.3455e+00, |dx|/|x|=0.0000e+00
info: Convergence criterion, component 3: |dx|=0.0000e+00, |x|=1.3881e+01, |dx|/|x|=0.0000e+00
info: Convergence criterion, component 4: |dx|=0.0000e+00, |x|=1.9006e+01, |dx|/|x|=0.0000e+00
info: Convergence criterion, component 5: |dx|=0.0000e+00, |x|=1.1422e+01, |dx|/|x|=0.0000e+00
info: Convergence criterion, component 6: |dx|=0.0000e+00, |x|=3.0790e+00, |dx|/|x|=0.0000e+00
info: [time] Iteration #2 took 0.0381688 s.
info: [time] Solving process #0 took 0.0828339 s in time step #1000
info: [time] Time step #1000 took 0.0864828 s.
info: [time] Output of timestep 1000 took 0.0085149 s.
info: The whole computation of the time stepping took 1000 steps, in which
         the accepted steps are 1000, and the rejected steps are 0.

info: [time] Execution took 98.7813 s.
info: OGS terminated on 2023-06-29 14:33:51+0200.
```

Syntax: ogs.exe   path_to_the_project_file > log.txt



```
C:\work\CloudStation\2023_conferences_and_trips\20230630_TU_Dresden\DecayChain_DONE>..\ogs\bin\ogs.exe 1d_decay_chain_GIA.prj > log.txt
```

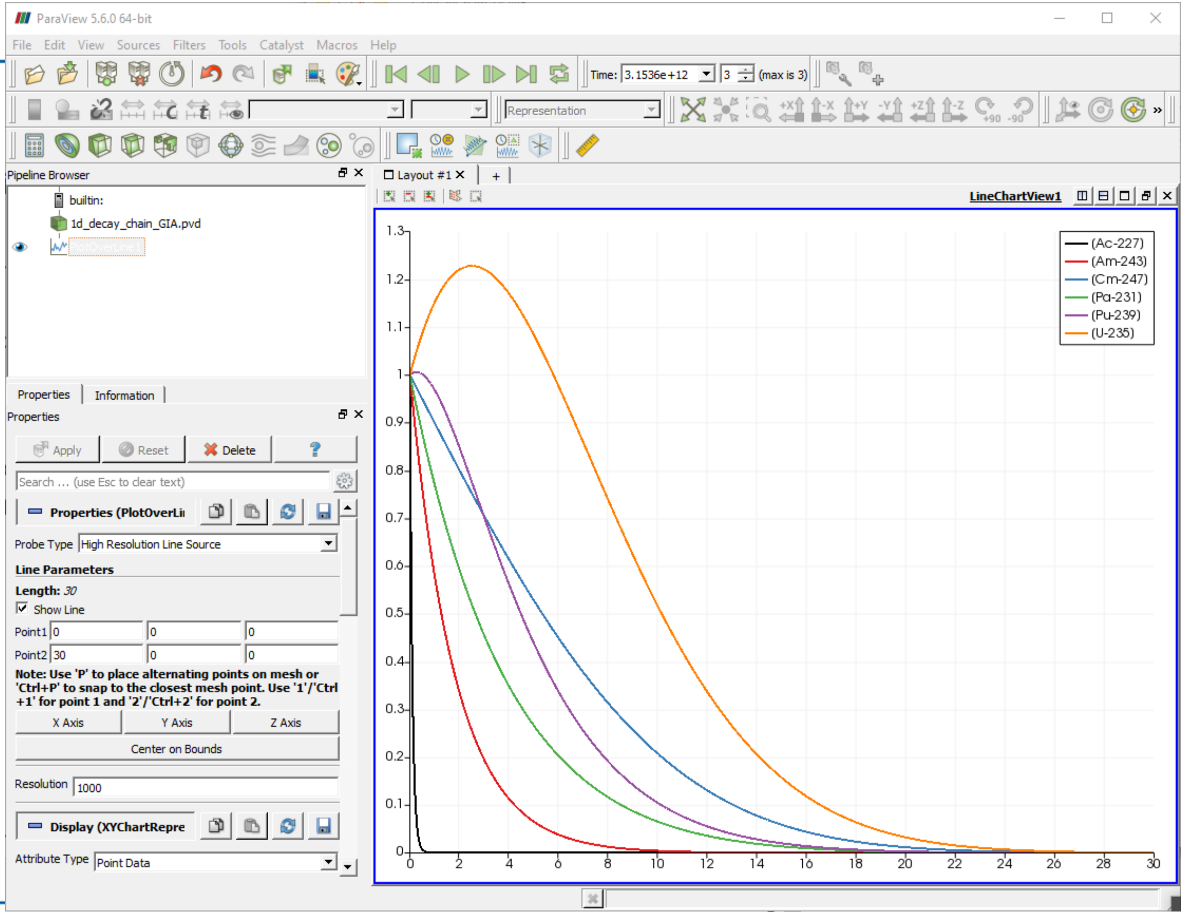
- Visualization

ParaView
Parallel Visualization Application

jupyter

Visualizing the simulation results

- Load the PVD file in Paraview
- Remember to “Apply” the View
- Add a “Plot over Line” Filter
- Just need to see from 0 to 30 m
- The results include from 0 to 100k years
- Choose which components to display (6 of them)
- Try to see the Spreadsheet View
- Try to export the data to CSV file



Time: 3.1536e+12 | 3 (max is 3)

builtin:
1d_decay_chain_GIA.pvd
PlotOverLine1

Probe Type High Resolution Line Source

Line Parameters

Length: *30*

☑ Show Line

Point1 0 0 0

Point2 30 0 0

Note: Use 'P' to place alternating points on mesh or 'Ctrl+P' to snap to the closest mesh point. Use '1'/'Ctrl+1' for point 1 and '2'/'Ctrl+2' for point 2.

Resolution 1000

Attribute Type Point Data

~The End~