

PS-MMM : Parallel Simulator for (IM-)Miscible Multi-Phase Mixing Flow

Generated by Doxygen 1.8.10

Thu Mar 10 2016 05:38:03

## Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Namespace List . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>1</b>
2.1	Class Hierarchy . . . . .	1
<b>3</b>	<b>Class Index</b>	<b>2</b>
3.1	Class List . . . . .	3
<b>4</b>	<b>File Index</b>	<b>4</b>
4.1	File List . . . . .	4
<b>5</b>	<b>Namespace Documentation</b>	<b>5</b>
5.1	Assembly Namespace Reference . . . . .	5
5.2	Assembly::CopyData Namespace Reference . . . . .	6
5.3	Assembly::Scratch Namespace Reference . . . . .	6
5.4	EquationData Namespace Reference . . . . .	6
5.4.1	Variable Documentation . . . . .	6
5.5	ps_mmm Namespace Reference . . . . .	7
5.6	ps_mmm::immiscible Namespace Reference . . . . .	7
5.7	ps_mmm::Particle Namespace Reference . . . . .	7
5.7.1	Typedef Documentation . . . . .	7
5.7.2	Variable Documentation . . . . .	7
5.8	ps_mmm::Particle::Generator Namespace Reference . . . . .	8
5.8.1	Function Documentation . . . . .	8
5.9	ps_mmm::Particle::Integrator Namespace Reference . . . . .	8
5.9.1	Function Documentation . . . . .	9
5.10	ps_mmm::Particle::Output Namespace Reference . . . . .	9
5.10.1	Function Documentation . . . . .	9
<b>6</b>	<b>Class Documentation</b>	<b>10</b>
6.1	ps_mmm::Particle::Output::ASCIIOutput< dim, T > Class Template Reference . . . . .	10
6.1.1	Constructor & Destructor Documentation . . . . .	10
6.1.2	Member Function Documentation . . . . .	10
6.2	ps_mmm::Particle::BaseParticle< dim > Class Template Reference . . . . .	11
6.2.1	Detailed Description . . . . .	12
6.2.2	Constructor & Destructor Documentation . . . . .	12
6.2.3	Member Function Documentation . . . . .	13
6.3	EquationData::concentrInitialValues< dim > Class Template Reference . . . . .	16
6.3.1	Constructor & Destructor Documentation . . . . .	17

6.3.2	Member Function Documentation	17
6.3.3	Member Data Documentation	17
6.4	EquationData::concentrInletValues< dim > Class Template Reference	17
6.4.1	Constructor & Destructor Documentation	18
6.4.2	Member Function Documentation	18
6.5	Assembly::Scratch::concentrMatrix< dim > Struct Template Reference	18
6.5.1	Constructor & Destructor Documentation	18
6.5.2	Member Data Documentation	18
6.6	Assembly::CopyData::concentrMatrix< dim > Struct Template Reference	19
6.6.1	Constructor & Destructor Documentation	19
6.6.2	Member Data Documentation	19
6.7	Assembly::Scratch::concentrRHS< dim > Struct Template Reference	19
6.7.1	Constructor & Destructor Documentation	20
6.7.2	Member Data Documentation	20
6.8	Assembly::CopyData::concentrRHS< dim > Struct Template Reference	21
6.8.1	Constructor & Destructor Documentation	21
6.8.2	Member Data Documentation	21
6.9	ps_mmm::Particle::DataParticle< dim, data_dim > Class Template Reference	21
6.9.1	Detailed Description	22
6.10	Assembly::Scratch::diffusion_step< dim > Struct Template Reference	22
6.10.1	Constructor & Destructor Documentation	23
6.10.2	Member Data Documentation	23
6.11	Assembly::CopyData::diffusion_step< dim > Struct Template Reference	24
6.11.1	Constructor & Destructor Documentation	24
6.11.2	Member Data Documentation	24
6.12	ps_mmm::Particle::Integrator::EulerIntegrator< dim, T > Class Template Reference	25
6.12.1	Detailed Description	25
6.12.2	Member Function Documentation	25
6.13	ps_mmm::Particle::Output::HDF5Output< dim, T > Class Template Reference	27
6.13.1	Constructor & Destructor Documentation	27
6.13.2	Member Function Documentation	27
6.14	ps_mmm::Particle::Integrator::HybridIntegrator< dim, T > Class Template Reference	28
6.14.1	Detailed Description	28
6.14.2	Constructor & Destructor Documentation	28
6.14.3	Member Function Documentation	28
6.15	EquationData::Inflow_Velocity< dim > Class Template Reference	30
6.15.1	Constructor & Destructor Documentation	30
6.15.2	Member Function Documentation	30
6.15.3	Member Data Documentation	30
6.16	ps_mmm::Particle::Output::Interface< dim, T > Class Template Reference	31

6.16.1 Detailed Description . . . . .	31
6.16.2 Constructor & Destructor Documentation . . . . .	31
6.16.3 Member Function Documentation . . . . .	32
6.16.4 Member Data Documentation . . . . .	32
6.17 ps_mmm::Particle::Generator::Interface< dim, T > Class Template Reference . . . . .	33
6.17.1 Detailed Description . . . . .	33
6.17.2 Constructor & Destructor Documentation . . . . .	33
6.17.3 Member Function Documentation . . . . .	33
6.18 ps_mmm::Particle::Integrator::Interface< dim, T > Class Template Reference . . . . .	34
6.18.1 Detailed Description . . . . .	34
6.18.2 Constructor & Destructor Documentation . . . . .	34
6.18.3 Member Function Documentation . . . . .	35
6.19 ps_mmm::immicible::Levelset< dim > Class Template Reference . . . . .	37
6.19.1 Constructor & Destructor Documentation . . . . .	37
6.19.2 Member Function Documentation . . . . .	37
6.20 ps_mmm::Particle::MPIDataInfo Class Reference . . . . .	38
6.20.1 Constructor & Destructor Documentation . . . . .	38
6.20.2 Member Data Documentation . . . . .	38
6.21 ps_mmm::Particle::Output::NullOutput< dim, T > Class Template Reference . . . . .	38
6.21.1 Constructor & Destructor Documentation . . . . .	39
6.21.2 Member Function Documentation . . . . .	39
6.22 EquationData::Outflow_Pressure< dim > Class Template Reference . . . . .	39
6.22.1 Constructor & Destructor Documentation . . . . .	40
6.22.2 Member Function Documentation . . . . .	40
6.22.3 Member Data Documentation . . . . .	40
6.23 UBC_mis_mixing< dim >::Parameters Struct Reference . . . . .	40
6.23.1 Constructor & Destructor Documentation . . . . .	42
6.23.2 Member Function Documentation . . . . .	42
6.23.3 Member Data Documentation . . . . .	42
6.24 UBC_mis_mixing< dim >::Postprocessor< dim > Class Template Reference . . . . .	45
6.24.1 Constructor & Destructor Documentation . . . . .	45
6.24.2 Member Function Documentation . . . . .	45
6.25 Assembly::Scratch::pressure_rot_step< dim > Struct Template Reference . . . . .	46
6.25.1 Constructor & Destructor Documentation . . . . .	46
6.25.2 Member Data Documentation . . . . .	46
6.26 Assembly::CopyData::pressure_rot_step< dim > Struct Template Reference . . . . .	47
6.26.1 Constructor & Destructor Documentation . . . . .	47
6.26.2 Member Data Documentation . . . . .	47
6.27 Assembly::CopyData::projection_step< dim > Struct Template Reference . . . . .	47
6.27.1 Constructor & Destructor Documentation . . . . .	48

6.27.2	Member Data Documentation . . . . .	48
6.28	Assembly::Scratch::projection_step< dim > Struct Template Reference . . . . .	48
6.28.1	Constructor & Destructor Documentation . . . . .	49
6.28.2	Member Data Documentation . . . . .	49
6.29	ps_mmm::Particle::Generator::RandomUniformGenerator< dim, T > Class Template Reference . . . . .	49
6.29.1	Constructor & Destructor Documentation . . . . .	50
6.29.2	Member Function Documentation . . . . .	51
6.30	Assembly::CopyData::relaxation_div_velocity_step< dim > Struct Template Reference . . . . .	51
6.30.1	Constructor & Destructor Documentation . . . . .	51
6.30.2	Member Data Documentation . . . . .	52
6.31	Assembly::Scratch::relaxation_div_velocity_step< dim > Struct Template Reference . . . . .	52
6.31.1	Constructor & Destructor Documentation . . . . .	52
6.31.2	Member Data Documentation . . . . .	52
6.32	ps_mmm::Particle::Integrator::RK2Integrator< dim, T > Class Template Reference . . . . .	53
6.32.1	Detailed Description . . . . .	53
6.32.2	Constructor & Destructor Documentation . . . . .	54
6.32.3	Member Function Documentation . . . . .	54
6.33	ps_mmm::Particle::Integrator::RK4Integrator< dim, T > Class Template Reference . . . . .	55
6.33.1	Detailed Description . . . . .	56
6.33.2	Constructor & Destructor Documentation . . . . .	56
6.33.3	Member Function Documentation . . . . .	56
6.34	UBC_mis_mixing< dim > Class Template Reference . . . . .	57
6.34.1	Constructor & Destructor Documentation . . . . .	57
6.34.2	Member Function Documentation . . . . .	57
6.35	ps_mmm::Particle::Output::VTUOutput< dim, T > Class Template Reference . . . . .	58
6.35.1	Constructor & Destructor Documentation . . . . .	58
6.35.2	Member Function Documentation . . . . .	59
6.36	ps_mmm::Particle::World< dim, T > Class Template Reference . . . . .	59
6.36.1	Constructor & Destructor Documentation . . . . .	60
6.36.2	Member Function Documentation . . . . .	61
<b>7</b>	<b>File Documentation</b>	<b>65</b>
7.1	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/assembly_↵ copydata.h File Reference . . . . .	65
7.2	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/class.h File Ref- erence . . . . .	66
7.3	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/equation_data.h File Reference . . . . .	66
7.4	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/include.h File Reference . . . . .	66
7.5	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/parameter.h File Reference . . . . .	68

7.6	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/generator.h File Reference	68
7.7	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/integrator.h File Reference	68
7.8	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/output.h File Reference	69
7.9	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/particle.h File Reference	69
7.10	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/world.h File Reference	70
7.11	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/phase/levelset.h File Reference	70
7.12	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/amr/amr.cc File Reference	71
7.13	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/constitutive_model/consti- _model.cc File Reference	71
7.14	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/main/main.cc File Refer- ence	71
7.14.1	Function Documentation	71
7.15	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/main/run.cc File Refer- ence	71
7.16	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/generator.cc File Reference	72
7.17	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/integrator.cc File Reference	72
7.18	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/output.cc File Reference	73
7.19	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/particle.cc File Reference	73
7.20	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/post_process/extract_↵ data.cc File Reference	74
7.21	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/post_process/post_↵ processing.cc File Reference	74
7.22	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/constructor.cc File Reference	74
7.23	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/mesh_↵ in.cc File Reference	74
7.24	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/read_↵ and_write.cc File Reference	75
7.25	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/setup_↵ dofs.cc File Reference	75
7.26	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/navier_stokes/control.cc File Reference	75
7.27	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/navier_stokes/projection_↵ _for_div_velocity.cc File Reference	75
7.28	/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/navier_stokes/solve_↵ _ns_equation.cc File Reference	75

7.29 /Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/phase/levelset.cc File Reference . . . . .	76
7.30 /Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/phase/solve_↔ hyperbolic_equation.cc File Reference . . . . .	76
7.31 /Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/support/utilities.cc File Reference . . . . .	76
<b>Index</b>	<b>77</b>

## 1 Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<b>Assembly</b>	<b>5</b>
<b>Assembly::CopyData</b>	<b>6</b>
<b>Assembly::Scratch</b>	<b>6</b>
<b>EquationData</b>	<b>6</b>
<b>ps_mmm</b>	<b>7</b>
<b>ps_mmm::immiscible</b>	<b>7</b>
<b>ps_mmm::Particle</b>	<b>7</b>
<b>ps_mmm::Particle::Generator</b>	<b>8</b>
<b>ps_mmm::Particle::Integrator</b>	<b>8</b>
<b>ps_mmm::Particle::Output</b>	<b>9</b>

## 2 Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

<b>ps_mmm::Particle::BaseParticle&lt; dim &gt;</b>	<b>11</b>
<b>ps_mmm::Particle::DataParticle&lt; dim, data_dim &gt;</b>	<b>21</b>
<b>Assembly::Scratch::concentrMatrix&lt; dim &gt;</b>	<b>18</b>
<b>Assembly::CopyData::concentrMatrix&lt; dim &gt;</b>	<b>19</b>
<b>Assembly::Scratch::concentrRHS&lt; dim &gt;</b>	<b>19</b>
<b>Assembly::CopyData::concentrRHS&lt; dim &gt;</b> DataPostprocessor	<b>21</b>
<b>UBC_mis_mixing&lt; dim &gt;::Postprocessor&lt; dim &gt;</b>	<b>45</b>

<b>Assembly::Scratch::diffusion_step&lt; dim &gt;</b>	<b>22</b>
<b>Assembly::CopyData::diffusion_step&lt; dim &gt;</b>	<b>24</b>
Function	
<b>EquationData::concentrInitialValues&lt; dim &gt;</b>	<b>16</b>
<b>EquationData::concentrInletValues&lt; dim &gt;</b>	<b>17</b>
<b>EquationData::Inflow_Velocity&lt; dim &gt;</b>	<b>30</b>
<b>EquationData::Outflow_Pressure&lt; dim &gt;</b>	<b>39</b>
<b>ps_mmm::Particle::Output::Interface&lt; dim, T &gt;</b>	<b>31</b>
<b>ps_mmm::Particle::Output::ASCIIOutput&lt; dim, T &gt;</b>	<b>10</b>
<b>ps_mmm::Particle::Output::HDF5Output&lt; dim, T &gt;</b>	<b>27</b>
<b>ps_mmm::Particle::Output::NullOutput&lt; dim, T &gt;</b>	<b>38</b>
<b>ps_mmm::Particle::Output::VTUOutput&lt; dim, T &gt;</b>	<b>58</b>
<b>ps_mmm::Particle::Generator::Interface&lt; dim, T &gt;</b>	<b>33</b>
<b>ps_mmm::Particle::Generator::RandomUniformGenerator&lt; dim, T &gt;</b>	<b>49</b>
<b>ps_mmm::Particle::Integrator::Interface&lt; dim, T &gt;</b>	<b>34</b>
<b>ps_mmm::Particle::Integrator::EulerIntegrator&lt; dim, T &gt;</b>	<b>25</b>
<b>ps_mmm::Particle::Integrator::HybridIntegrator&lt; dim, T &gt;</b>	<b>28</b>
<b>ps_mmm::Particle::Integrator::RK2Integrator&lt; dim, T &gt;</b>	<b>53</b>
<b>ps_mmm::Particle::Integrator::RK4Integrator&lt; dim, T &gt;</b>	<b>55</b>
<b>ps_mmm::immiscible::Levelset&lt; dim &gt;</b>	<b>37</b>
<b>ps_mmm::Particle::MPIDataInfo</b>	<b>38</b>
<b>UBC_mis_mixing&lt; dim &gt;::Parameters</b>	<b>40</b>
<b>Assembly::Scratch::pressure_rot_step&lt; dim &gt;</b>	<b>46</b>
<b>Assembly::CopyData::pressure_rot_step&lt; dim &gt;</b>	<b>47</b>
<b>Assembly::CopyData::projection_step&lt; dim &gt;</b>	<b>47</b>
<b>Assembly::Scratch::projection_step&lt; dim &gt;</b>	<b>48</b>
<b>Assembly::CopyData::relaxation_div_velocity_step&lt; dim &gt;</b>	<b>51</b>
<b>Assembly::Scratch::relaxation_div_velocity_step&lt; dim &gt;</b>	<b>52</b>
<b>UBC_mis_mixing&lt; dim &gt;</b>	<b>57</b>
<b>ps_mmm::Particle::World&lt; dim, T &gt;</b>	<b>59</b>

### 3 Class Index



## 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">ps_mmm::Particle::Output::ASCIIOutput&lt; dim, T &gt;</a>	10
<a href="#">ps_mmm::Particle::BaseParticle&lt; dim &gt;</a> Base class of particles - represents a particle with position, velocity, and an ID number	11
<a href="#">EquationData::concentrInitialValues&lt; dim &gt;</a>	16
<a href="#">EquationData::concentrInletValues&lt; dim &gt;</a>	17
<a href="#">Assembly::Scratch::concentrMatrix&lt; dim &gt;</a>	18
<a href="#">Assembly::CopyData::concentrMatrix&lt; dim &gt;</a>	19
<a href="#">Assembly::Scratch::concentrRHS&lt; dim &gt;</a>	19
<a href="#">Assembly::CopyData::concentrRHS&lt; dim &gt;</a>	21
<a href="#">ps_mmm::Particle::DataParticle&lt; dim, data_dim &gt;</a> <a href="#">DataParticle</a> provides an example of how to extend the <a href="#">BaseParticle</a> class to include related particle data	21
<a href="#">Assembly::Scratch::diffusion_step&lt; dim &gt;</a>	22
<a href="#">Assembly::CopyData::diffusion_step&lt; dim &gt;</a>	24
<a href="#">ps_mmm::Particle::Integrator::EulerIntegrator&lt; dim, T &gt;</a> Euler scheme integrator, where $y_{n+1} = y_n + dt * v(y_n)$	25
<a href="#">ps_mmm::Particle::Output::HDF5Output&lt; dim, T &gt;</a>	27
<a href="#">ps_mmm::Particle::Integrator::HybridIntegrator&lt; dim, T &gt;</a> <a href="#">Integrator</a> which chooses Euler, RK2 or RK4 depending on characteristics of the cell a particle is in	28
<a href="#">EquationData::Inflow_Velocity&lt; dim &gt;</a>	30
<a href="#">ps_mmm::Particle::Output::Interface&lt; dim, T &gt;</a> Abstract base class used for classes that generate particle output	31
<a href="#">ps_mmm::Particle::Generator::Interface&lt; dim, T &gt;</a> Abstract base class used for classes that generate particles	33
<a href="#">ps_mmm::Particle::Integrator::Interface&lt; dim, T &gt;</a> An abstract class defining virtual methods for performing integration of particle paths through the simulation velocity field	34
<a href="#">ps_mmm::immiscible::Levelset&lt; dim &gt;</a>	37
<a href="#">ps_mmm::Particle::MPIDataInfo</a>	38
<a href="#">ps_mmm::Particle::Output::NullOutput&lt; dim, T &gt;</a>	38
<a href="#">EquationData::Outflow_Pressure&lt; dim &gt;</a>	39
<a href="#">UBC_mis_mixing&lt; dim &gt;::Parameters</a>	40
<a href="#">UBC_mis_mixing&lt; dim &gt;::Postprocessor&lt; dim &gt;</a>	45

<a href="#">Assembly::Scratch::pressure_rot_step&lt; dim &gt;</a>	46
<a href="#">Assembly::CopyData::pressure_rot_step&lt; dim &gt;</a>	47
<a href="#">Assembly::CopyData::projection_step&lt; dim &gt;</a>	47
<a href="#">Assembly::Scratch::projection_step&lt; dim &gt;</a>	48
<a href="#">ps_mmm::Particle::Generator::RandomUniformGenerator&lt; dim, T &gt;</a>	49
<a href="#">Assembly::CopyData::relaxation_div_velocity_step&lt; dim &gt;</a>	51
<a href="#">Assembly::Scratch::relaxation_div_velocity_step&lt; dim &gt;</a>	52
<a href="#">ps_mmm::Particle::Integrator::RK2Integrator&lt; dim, T &gt;</a> Runge Kutta second order integrator, where $y_{n+1} = y_n + dt*v(0.5*k_1)$ , $k_1 = dt*v(y_n)$	53
<a href="#">ps_mmm::Particle::Integrator::RK4Integrator&lt; dim, T &gt;</a> Runge Kutta fourth order integrator, where $y_{n+1} = y_n + (1/6)*k_1 + (1/3)*k_2 + (1/3)*k_3 + (1/6)*k_4$ and $k_1, k_2, k_3, k_4$ are defined as usual	55
<a href="#">UBC_mis_mixing&lt; dim &gt;</a>	57
<a href="#">ps_mmm::Particle::Output::VTUOutput&lt; dim, T &gt;</a>	58
<a href="#">ps_mmm::Particle::World&lt; dim, T &gt;</a>	59

## 4 File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/assembly_↵ copydata.h</a>	65
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/class.h</a>	66
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/equation_↵ data.h</a>	66
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/include.h</a>	66
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/parameter.h</a>	68
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/generator.↵ h</a>	68
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/integrator.↵ h</a>	68
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/output.↵ h</a>	69
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/particle.↵ h</a>	69
<a href="#">/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/world.↵ h</a>	70

/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/phase/ <a href="#">levelset.h</a>	70
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/amr/ <a href="#">amr.cc</a>	71
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/constitutive_model/ <a href="#">constitutive_model.cc</a>	71
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/main/ <a href="#">main.cc</a>	71
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/main/ <a href="#">run.cc</a>	71
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/ <a href="#">generator.cc</a>	72
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/ <a href="#">integrator.cc</a>	72
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/ <a href="#">output.cc</a>	73
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/ <a href="#">particle.cc</a>	73
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/post_process/ <a href="#">extract_data.cc</a>	74
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/post_process/ <a href="#">post_processing.cc</a>	74
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/ <a href="#">constructor.cc</a>	74
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/ <a href="#">mesh_in.cc</a>	74
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/ <a href="#">read_and_write.cc</a>	75
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/pre_process/ <a href="#">setup_dofs.cc</a>	75
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/navier_stokes/ <a href="#">control.cc</a>	75
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/navier_stokes/ <a href="#">projection_for_div_velocity.cc</a>	75
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/navier_stokes/ <a href="#">solve_ns_equation.cc</a>	75
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/phase/ <a href="#">levelset.cc</a>	76
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/solver/phase/ <a href="#">solve_hyperbolic_equation.cc</a>	76
/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/support/ <a href="#">utilities.cc</a>	76

## 5 Namespace Documentation

### 5.1 Assembly Namespace Reference

## Namespaces

- [CopyData](#)
- [Scratch](#)

## 5.2 Assembly::CopyData Namespace Reference

### Classes

- struct [concentrMatrix](#)
- struct [concentrRHS](#)
- struct [diffusion\\_step](#)
- struct [pressure\\_rot\\_step](#)
- struct [projection\\_step](#)
- struct [relaxation\\_div\\_velocity\\_step](#)

## 5.3 Assembly::Scratch Namespace Reference

### Classes

- struct [concentrMatrix](#)
- struct [concentrRHS](#)
- struct [diffusion\\_step](#)
- struct [pressure\\_rot\\_step](#)
- struct [projection\\_step](#)
- struct [relaxation\\_div\\_velocity\\_step](#)

## 5.4 EquationData Namespace Reference

### Classes

- class [concentrInitialValues](#)
- class [concentrInletValues](#)
- class [Inflow\\_Velocity](#)
- class [Outflow\\_Pressure](#)

### Variables

- const double [pipe\\_diameter](#) = 19.05
- const double [gravityy\\_accelation](#) = 9800
- const double [upstream\\_concentr](#) = 0.0
- const double [downstream\\_concentr](#) = 1.0
- const double [kinematic\\_viscosity](#) = 1.0

### 5.4.1 Variable Documentation

5.4.1.1 const double EquationData::downstream\_concentr = 1.0

5.4.1.2 const double EquationData::gravityy\_accelation = 9800

5.4.1.3 const double EquationData::kinematic\_viscosity = 1.0

5.4.1.4 const double EquationData::pipe\_diameter = 19.05

5.4.1.5 const double EquationData::upstream\_concentr = 0.0

## 5.5 ps\_mmm Namespace Reference

### Namespaces

- [immicible](#)
- [Particle](#)

## 5.6 ps\_mmm::immicible Namespace Reference

### Classes

- class [Levelset](#)

## 5.7 ps\_mmm::Particle Namespace Reference

### Namespaces

- [Generator](#)
- [Integrator](#)
- [Output](#)

### Classes

- class [BaseParticle](#)  
*Base class of particles - represents a particle with position, velocity, and an ID number.*
- class [DataParticle](#)  
*[DataParticle](#) provides an example of how to extend the [BaseParticle](#) class to include related particle data.*
- class [MPIDataInfo](#)
- class [World](#)

### Typedefs

- typedef std::pair< int, int > [LevelInd](#)  
*Typedef of cell level/index pair.*

### Variables

- const int [PARTICLE\\_XFER\\_TAG](#) = 382  
*MPI tag for particle transfers.*

### 5.7.1 Typedef Documentation

#### 5.7.1.1 typedef std::pair<int, int> ps\_mmm::Particle::LevelInd

Typedef of cell level/index pair.

### 5.7.2 Variable Documentation

#### 5.7.2.1 const int ps\_mmm::Particle::PARTICLE\_XFER\_TAG = 382

MPI tag for particle transfers.

## 5.8 ps\_mmm::Particle::Generator Namespace Reference

### Classes

- class [Interface](#)  
*Abstract base class used for classes that generate particles.*
- class [RandomUniformGenerator](#)

### Functions

- `template<int dim, class T > Interface< dim, T > * create\_generator\_object (const std::string &generator_type)`  
*Create a generator object.*
- `std::string generator\_object\_names ()`  
*Return a list of names (separated by '|') of possible particle generators.*

### 5.8.1 Function Documentation

**5.8.1.1** `template<int dim, class T > Interface< dim, T > * ps\_mmm::Particle::Generator::create\_generator\_object ( const std::string & generator_type )`

Create a generator object.

#### Parameters

<code>in</code>	<code><i>generator_type</i></code>	Name of the type of generator to create
-----------------	------------------------------------	---

#### Returns

pointer to the generator. Caller needs to delete this pointer.

**5.8.1.2** `std::string ps\_mmm::Particle::Generator::generator\_object\_names ( )`

Return a list of names (separated by '|') of possible particle generators.

## 5.9 ps\_mmm::Particle::Integrator Namespace Reference

### Classes

- class [EulerIntegrator](#)  
*Euler scheme integrator, where  $y_{n+1} = y_n + dt * v(y_n)$ .*
- class [HybridIntegrator](#)  
*[Integrator](#) which chooses Euler, RK2 or RK4 depending on characteristics of the cell a particle is in.*
- class [Interface](#)  
*An abstract class defining virtual methods for performing integration of particle paths through the simulation velocity field.*
- class [RK2Integrator](#)  
*Runge Kutta second order integrator, where  $y_{n+1} = y_n + dt * v(0.5 * k_1)$ ,  $k_1 = dt * v(y_n)$ .*
- class [RK4Integrator](#)  
*Runge Kutta fourth order integrator, where  $y_{n+1} = y_n + (1/6)*k_1 + (1/3)*k_2 + (1/3)*k_3 + (1/6)*k_4$  and  $k_1, k_2, k_3, k_4$  are defined as usual.*

## Functions

- `template<int dim, class T >`  
`Interface< dim, T > * create\_integrator\_object (const std::string &integrator_name)`  
*Create an integrator object.*
- `std::string integrator\_object\_names ()`  
*Return a list of names (separated by '|') of possible integrator classes for particles.*

## 5.9.1 Function Documentation

5.9.1.1 `template<int dim, class T > Interface< dim, T > * ps_mmm::Particle::Integrator::create_integrator_object ( const std::string & integrator_name )`

Create an integrator object.

## Parameters

<code>in</code>	<code>integrator_name</code>	Name of the type of integrator.
-----------------	------------------------------	---------------------------------

## Returns

Pointer to instantiated generator object

5.9.1.2 `std::string ps_mmm::Particle::Integrator::integrator_object_names ( )`

Return a list of names (separated by '|') of possible integrator classes for particles.

## 5.10 ps\_mmm::Particle::Output Namespace Reference

## Classes

- class [ASCIIOutput](#)
- class [HDF5Output](#)
- class [Interface](#)  
*Abstract base class used for classes that generate particle output.*
- class [NullOutput](#)
- class [VTUOutput](#)

## Functions

- `template<int dim, class T >`  
`Interface< dim, T > * create\_output\_object (const std::string &data_format_name, const std::string &output_directory, const MPI_Comm communicator)`  
*Create an output object.*
- `std::string output\_object\_names ()`  
*Return a list of names (separated by '|') of possible writers of graphical output formats for particle data.*

## 5.10.1 Function Documentation

5.10.1.1 `template<int dim, class T > Interface< dim, T > * ps_mmm::Particle::Output::create_output_object ( const std::string & data_format_name, const std::string & output_directory, const MPI_Comm communicator )`

Create an output object.

**Parameters**

in	<i>data_format_↔ name</i>	Name of the format in which the created output writer should produce its files
in	<i>output_directory</i>	Directory into which to put the data files
in	<i>communicator</i>	MPI communicator object that describes this simulation

**Returns**

a pointer to an output object, needs to be deleted by the caller.

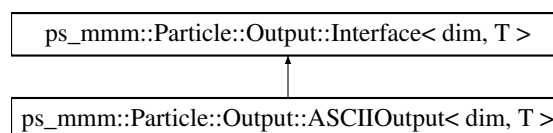
### 5.10.1.2 `std::string ps_mmm::Particle::Output::output_object_names ( )`

Return a list of names (separated by '|') of possible writers of graphical output formats for particle data.

## 6 Class Documentation

### 6.1 `ps_mmm::Particle::Output::ASCIIOutput< dim, T >` Class Template Reference

Inheritance diagram for `ps_mmm::Particle::Output::ASCIIOutput< dim, T >`:

**Public Member Functions**

- `ASCIIOutput` (const std::string &output\_directory, const MPI\_Comm communicator)  
*Constructor.*
- virtual std::string `output_particle_data` (const std::multimap< `LevelInd`, T > &particles, const double &current\_time)  
*Write data about the particles specified in the first argument to a file.*

**Additional Inherited Members**

#### 6.1.1 Constructor & Destructor Documentation

##### 6.1.1.1 `template<int dim, class T > ps_mmm::Particle::Output::ASCIIOutput< dim, T >::ASCIIOutput ( const std::string & output_directory, const MPI_Comm communicator ) [inline]`

Constructor.

**Parameters**

in	<i>The</i>	directory into which output files shall be placed.
in	<i>The</i>	MPI communicator that describes this simulation.

#### 6.1.2 Member Function Documentation



6.1.2.1 `template<int dim, class T > virtual std::string ps_mmm::Particle::Output::ASCIIOutput< dim, T >::output_particle_data ( const std::multimap< LevelInd, T > & particles, const double & current_time ) [inline], [virtual]`

Write data about the particles specified in the first argument to a file.

If possible, encode the current simulation time into this file using the data provided in the second argument.

#### Parameters

<code>in</code>	<code>particles</code>	The set of particles to generate a graphical representation for
<code>in</code>	<code>current_time</code>	Current time of the simulation, given as either years or seconds, as selected in the input file. In other words, output writers do not need to know the units in which time is described.

#### Returns

The name of the file that was written, or any other information that describes what output was produced if for example multiple files were created.

Implements `ps_mmm::Particle::Output::Interface< dim, T >`.

The documentation for this class was generated from the following file:

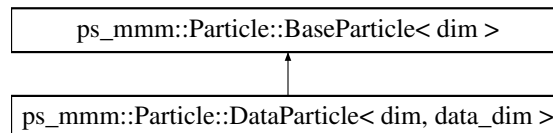
- `/Users/miramus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/output.cc`

## 6.2 ps\_mmm::Particle::BaseParticle< dim > Class Template Reference

Base class of particles - represents a particle with position, velocity, and an ID number.

`#include <particle.h>`

Inheritance diagram for `ps_mmm::Particle::BaseParticle< dim >`:



#### Public Member Functions

- `BaseParticle ()`  
*Empty constructor for `BaseParticle`, creates a particle at the origin with zero velocity.*
- `BaseParticle (const Point< dim > &new_loc, const double &new_id)`  
*Constructor for `BaseParticle`, creates a particle with the specified ID at the specified location with zero velocity.*
- `virtual ~BaseParticle ()`  
*Destructor for `BaseParticle`.*
- `virtual unsigned int read_data (const std::vector< double > &data, const unsigned int &pos)`  
*Read the particle data from the specified vector of doubles.*
- `virtual void write_data (std::vector< double > &data) const`  
*Write particle data to a vector of doubles.*
- `void set_location (const Point< dim > &new_loc)`  
*Set the location of this particle.*
- `Point< dim > get_location () const`  
*Get the location of this particle.*
- `void set_velocity (Point< dim > new_vel)`

- Set the velocity of this particle.*

  - `Point< dim > get_velocity () const`

*Get the velocity of this particle.*
- `double get_id () const`

*Get the ID number of this particle.*
- `bool local () const`

*Check whether the particle is marked as being local to this subdomain.*
- `void set_local (bool new_local)`

*Mark the particle as being local or not.*
- `bool vel_check () const`

*Whether to check the particle velocity at its current location.*
- `void set_vel_check (bool new_vel_check)`

*Mark whether to check the particle velocity.*

#### Static Public Member Functions

- `static unsigned int data_len ()`

*Get the number of doubles required to represent this particle for communication.*
- `static void add_mpi_types (std::vector< MPIDataInfo > &data_info)`

*Add the MPI data description for this particle type to the vector.*

#### 6.2.1 Detailed Description

`template<int dim>class ps_mmm::Particle::BaseParticle< dim >`

Base class of particles - represents a particle with position, velocity, and an ID number.

This class can be extended to include data related to a particle. An example of this is shown in the [DataParticle](#) class.

#### 6.2.2 Constructor & Destructor Documentation

**6.2.2.1** `template<int dim> ps_mmm::Particle::BaseParticle< dim >::BaseParticle ( ) [inline]`

Empty constructor for [BaseParticle](#), creates a particle at the origin with zero velocity.

**6.2.2.2** `template<int dim> ps_mmm::Particle::BaseParticle< dim >::BaseParticle ( const Point< dim > & new_loc, const double & new_id ) [inline]`

Constructor for [BaseParticle](#), creates a particle with the specified ID at the specified location with zero velocity.

Note that [ps\\_mmm](#) does not check for duplicate particle IDs so the user must be sure the IDs are unique over all processes.

##### Parameters

<code>in</code>	<code>new_loc</code>	Initial location of particle.
<code>in</code>	<code>new_id</code>	Globally unique ID number of particle.

**6.2.2.3** `template<int dim> ps_mmm::Particle::BaseParticle< dim >::~~BaseParticle ( ) [inline], [virtual]`

Destructor for [BaseParticle](#).

### 6.2.3 Member Function Documentation

6.2.3.1 `template<int dim> void ps_mmm::Particle::BaseParticle< dim >::add_mpi_types ( std::vector< MPIDataInfo > & data_info ) [static]`

Add the MPI data description for this particle type to the vector.

**Parameters**

<i>in, out</i>	<i>data_info</i>	Vector to which MPI data description is appended.
----------------	------------------	---

**6.2.3.2** `template<int dim> unsigned int ps_mmm::Particle::BaseParticle< dim >::data_len ( ) [static]`

Get the number of doubles required to represent this particle for communication.

**Returns**

Number of doubles required to represent this particle

**6.2.3.3** `template<int dim> double ps_mmm::Particle::BaseParticle< dim >::get_id ( ) const`

Get the ID number of this particle.

**Returns**

The id of this particle.

**6.2.3.4** `template<int dim> Point< dim > ps_mmm::Particle::BaseParticle< dim >::get_location ( ) const`

Get the location of this particle.

**Returns**

The location of this particle.

**6.2.3.5** `template<int dim> Point< dim > ps_mmm::Particle::BaseParticle< dim >::get_velocity ( ) const`

Get the velocity of this particle.

**Returns**

The velocity of this particle.

**6.2.3.6** `template<int dim> bool ps_mmm::Particle::BaseParticle< dim >::local ( ) const`

Check whether the particle is marked as being local to this subdomain.

Note that this function does not actually perform the check for locality.

**Returns**

Whether the particle is marked as local.

**6.2.3.7** `template<int dim> unsigned int ps_mmm::Particle::BaseParticle< dim >::read_data ( const std::vector< double > & data, const unsigned int & pos ) [virtual]`

Read the particle data from the specified vector of doubles.

**Parameters**

<i>in</i>	<i>data</i>	The vector of double data to read from.
<i>in</i>	<i>pos</i>	The position in the data vector to start reading from.

**Returns**

The position in the vector of the next unread double.

6.2.3.8 `template<int dim> void ps_mmm::Particle::BaseParticle< dim >::set_local ( bool new_local )`

Mark the particle as being local or not.

Note that this function does not perform the check for locality.

**Parameters**

<i>in</i>	<i>new_local</i>	Whether to mark the particle as local.
-----------	------------------	--

**6.2.3.9** `template<int dim> void ps_mmm::Particle::BaseParticle< dim >::set_location ( const Point< dim > & new_loc )`

Set the location of this particle.

Note that this does not check whether this is a valid location in the simulation domain.

**Parameters**

<i>in</i>	<i>new_loc</i>	The new location for this particle.
-----------	----------------	-------------------------------------

**6.2.3.10** `template<int dim> void ps_mmm::Particle::BaseParticle< dim >::set_vel_check ( bool new_vel_check )`

Mark whether to check the particle velocity.

**Parameters**

<i>in</i>	<i>new_vel_check</i>	Whether to check the particle velocity.
-----------	----------------------	---

**6.2.3.11** `template<int dim> void ps_mmm::Particle::BaseParticle< dim >::set_velocity ( Point< dim > new_vel )`

Set the velocity of this particle.

**Parameters**

<i>in</i>	<i>new_vel</i>	The new velocity for this particle.
-----------	----------------	-------------------------------------

**6.2.3.12** `template<int dim> bool ps_mmm::Particle::BaseParticle< dim >::vel_check ( ) const`

Whether to check the particle velocity at its current location.

This is used for integrators where the particle velocity may not need to be checked every step.

**Returns**

Whether to check the particle velocity

**6.2.3.13** `template<int dim> void ps_mmm::Particle::BaseParticle< dim >::write_data ( std::vector< double > & data ) const [virtual]`

Write particle data to a vector of doubles.

**Parameters**

<i>in, out</i>	<i>data</i>	The vector of doubles to write integrator data into.
----------------	-------------	--

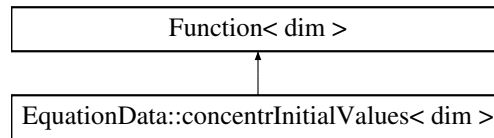
The documentation for this class was generated from the following files:

- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/[particle.h](#)
- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/particle/[particle.cc](#)

## 6.3 EquationData::concentrInitialValues< dim > Class Template Reference

```
#include <equation_data.h>
```

Inheritance diagram for EquationData::concentrInitialValues< dim >:



#### Public Member Functions

- [concentrInitialValues](#) (double *x*)
- virtual double [value](#) (const Point< dim > &*p*, const unsigned int component=0) const
- virtual void [vector\\_value](#) (const Point< dim > &*p*, Vector< double > &*value*) const
- virtual void [vector\\_value\\_list](#) (const std::vector< Point< dim > > &*p*, std::vector< Vector< double > > &*values*) const

#### Public Attributes

- double *x*

#### 6.3.1 Constructor & Destructor Documentation

6.3.1.1 `template<int dim> EquationData::concentrInitialValues< dim >::concentrInitialValues ( double x )`

#### 6.3.2 Member Function Documentation

6.3.2.1 `template<int dim> double EquationData::concentrInitialValues< dim >::value ( const Point< dim > & p, const unsigned int component = 0 ) const` [virtual]

6.3.2.2 `template<int dim> void EquationData::concentrInitialValues< dim >::vector_value ( const Point< dim > & p, Vector< double > & value ) const` [virtual]

6.3.2.3 `template<int dim> void EquationData::concentrInitialValues< dim >::vector_value_list ( const std::vector< Point< dim > > & p, std::vector< Vector< double > > & values ) const` [virtual]

#### 6.3.3 Member Data Documentation

6.3.3.1 `template<int dim> double EquationData::concentrInitialValues< dim >::x`

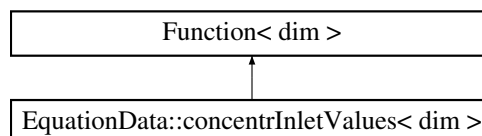
The documentation for this class was generated from the following file:

- `/Users/miramus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/equation_data.h`

## 6.4 EquationData::concentrInletValues< dim > Class Template Reference

```
#include <equation_data.h>
```

Inheritance diagram for EquationData::concentrInletValues< dim >:



## Public Member Functions

- [concentrInletValues](#) ()
- virtual double [value](#) (const Point< dim > &p, const unsigned int component=0) const
- virtual void [vector\\_value](#) (const Point< dim > &p, Vector< double > &value) const

### 6.4.1 Constructor & Destructor Documentation

6.4.1.1 `template<int dim> EquationData::concentrInletValues< dim >::concentrInletValues ( ) [inline]`

### 6.4.2 Member Function Documentation

6.4.2.1 `template<int dim> double EquationData::concentrInletValues< dim >::value ( const Point< dim > & p, const unsigned int component = 0 ) const [virtual]`

6.4.2.2 `template<int dim> void EquationData::concentrInletValues< dim >::vector_value ( const Point< dim > & p, Vector< double > & value ) const [virtual]`

The documentation for this class was generated from the following file:

- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/[equation\\_data.h](#)

## 6.5 Assembly::Scratch::concentrMatrix< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

## Public Member Functions

- [concentrMatrix](#) (const FiniteElement< dim > &concentr\_fe, const Mapping< dim > &mapping, const Quadrature< dim > &concentr\_quadrature)
- [concentrMatrix](#) (const [concentrMatrix](#) &data)

## Public Attributes

- FEValues< dim > [concentr\\_fe\\_values](#)
- std::vector< double > [phi\\_T](#)
- std::vector< Tensor< 1, dim > > [grad\\_phi\\_T](#)

### 6.5.1 Constructor & Destructor Documentation

6.5.1.1 `template<int dim> Assembly::Scratch::concentrMatrix< dim >::concentrMatrix ( const FiniteElement< dim > & concentr_fe, const Mapping< dim > & mapping, const Quadrature< dim > & concentr_quadrature )`

6.5.1.2 `template<int dim> Assembly::Scratch::concentrMatrix< dim >::concentrMatrix ( const concentrMatrix< dim > & data )`

### 6.5.2 Member Data Documentation

6.5.2.1 `template<int dim> FEValues<dim> Assembly::Scratch::concentrMatrix< dim >::concentr_fe_values`

6.5.2.2 `template<int dim> std::vector<Tensor<1,dim>> Assembly::Scratch::concentrMatrix< dim >::grad_phi_T`

6.5.2.3 `template<int dim> std::vector<double> Assembly::Scratch::concentrMatrix< dim >::phi_T`

The documentation for this struct was generated from the following file:



- [/Users/miramus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/assembly\\_copydata.h](#)

## 6.6 Assembly::CopyData::concentrMatrix< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- [concentrMatrix](#) (const FiniteElement< dim > &concentr\_fe)
- [concentrMatrix](#) (const [concentrMatrix](#) &data)

### Public Attributes

- FullMatrix< double > [local\\_mass\\_matrix](#)
- FullMatrix< double > [local\\_stiffness\\_matrix](#)
- std::vector< types::global\_dof\_index > [local\\_dof\\_indices](#)

### 6.6.1 Constructor & Destructor Documentation

6.6.1.1 `template<int dim> Assembly::CopyData::concentrMatrix< dim >::concentrMatrix ( const FiniteElement< dim > & concentr_fe )`

6.6.1.2 `template<int dim> Assembly::CopyData::concentrMatrix< dim >::concentrMatrix ( const concentrMatrix< dim > & data )`

### 6.6.2 Member Data Documentation

6.6.2.1 `template<int dim> std::vector<types::global_dof_index> Assembly::CopyData::concentrMatrix< dim >::local_dof_indices`

6.6.2.2 `template<int dim> FullMatrix<double> Assembly::CopyData::concentrMatrix< dim >::local_mass_matrix`

6.6.2.3 `template<int dim> FullMatrix<double> Assembly::CopyData::concentrMatrix< dim >::local_stiffness_matrix`

The documentation for this struct was generated from the following file:

- [/Users/miramus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/assembly\\_copydata.h](#)

## 6.7 Assembly::Scratch::concentrRHS< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- [concentrRHS](#) (const FiniteElement< dim > &concentr\_fe, const FiniteElement< dim > &fe\_velocity, const Mapping< dim > &mapping, const Quadrature< dim > &quadrature)
- [concentrRHS](#) (const [concentrRHS](#) &data)

### Public Attributes

- FEValues< dim > [concentr\\_fe\\_values](#)
- FEValues< dim > [fe\\_velocity\\_values](#)
- std::vector< double > [phi\\_T](#)

- `std::vector< Tensor< 1, dim > > grad_phi_T`
- `std::vector< Tensor< 1, dim > > old_velocity_values`
- `std::vector< Tensor< 1, dim > > old_old_velocity_values`
- `std::vector< SymmetricTensor< 2, dim > > old_strain_rates`
- `std::vector< SymmetricTensor< 2, dim > > old_old_strain_rates`
- `std::vector< double > old_concentr_values`
- `std::vector< double > old_old_concentr_values`
- `std::vector< Tensor< 1, dim > > old_concentr_grads`
- `std::vector< Tensor< 1, dim > > old_old_concentr_grads`
- `std::vector< double > old_concentr_laplacians`
- `std::vector< double > old_old_concentr_laplacians`

### 6.7.1 Constructor & Destructor Documentation

- 6.7.1.1 `template<int dim> Assembly::Scratch::concentrRHS< dim >::concentrRHS ( const FiniteElement< dim > & concentr_fe, const FiniteElement< dim > & fe_velocity, const Mapping< dim > & mapping, const Quadrature< dim > & quadrature )`
- 6.7.1.2 `template<int dim> Assembly::Scratch::concentrRHS< dim >::concentrRHS ( const concentrRHS< dim > & data )`

### 6.7.2 Member Data Documentation

- 6.7.2.1 `template<int dim> FEValues<dim> Assembly::Scratch::concentrRHS< dim >::concentr_fe_values`
- 6.7.2.2 `template<int dim> FEValues<dim> Assembly::Scratch::concentrRHS< dim >::fe_velocity_values`
- 6.7.2.3 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::concentrRHS< dim >::grad_phi_T`
- 6.7.2.4 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::concentrRHS< dim >::old_concentr_grads`
- 6.7.2.5 `template<int dim> std::vector<double> Assembly::Scratch::concentrRHS< dim >::old_concentr_laplacians`
- 6.7.2.6 `template<int dim> std::vector<double> Assembly::Scratch::concentrRHS< dim >::old_concentr_values`
- 6.7.2.7 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::concentrRHS< dim >::old_old_concentr_grads`
- 6.7.2.8 `template<int dim> std::vector<double> Assembly::Scratch::concentrRHS< dim >::old_old_concentr_laplacians`
- 6.7.2.9 `template<int dim> std::vector<double> Assembly::Scratch::concentrRHS< dim >::old_old_concentr_values`
- 6.7.2.10 `template<int dim> std::vector<SymmetricTensor<2,dim> > Assembly::Scratch::concentrRHS< dim >::old_old_strain_rates`
- 6.7.2.11 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::concentrRHS< dim >::old_old_velocity_values`
- 6.7.2.12 `template<int dim> std::vector<SymmetricTensor<2,dim> > Assembly::Scratch::concentrRHS< dim >::old_strain_rates`
- 6.7.2.13 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::concentrRHS< dim >::old_velocity_values`
- 6.7.2.14 `template<int dim> std::vector<double> Assembly::Scratch::concentrRHS< dim >::phi_T`

The documentation for this struct was generated from the following file:

- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/[assembly\\_copydata.h](#)

## 6.8 Assembly::CopyData::concentrRHS< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- [concentrRHS](#) (const FiniteElement< dim > &concentr\_fe)
- [concentrRHS](#) (const [concentrRHS](#) &data)

### Public Attributes

- Vector< double > [local\\_rhs](#)
- std::vector< types::global\_dof\_index > [local\\_dof\\_indices](#)
- FullMatrix< double > [matrix\\_for\\_bc](#)

### 6.8.1 Constructor & Destructor Documentation

6.8.1.1 `template<int dim> Assembly::CopyData::concentrRHS< dim >::concentrRHS ( const FiniteElement< dim > & concentr_fe )`

6.8.1.2 `template<int dim> Assembly::CopyData::concentrRHS< dim >::concentrRHS ( const concentrRHS< dim > & data )`

### 6.8.2 Member Data Documentation

6.8.2.1 `template<int dim> std::vector<types::global_dof_index> Assembly::CopyData::concentrRHS< dim >::local_dof_indices`

6.8.2.2 `template<int dim> Vector<double> Assembly::CopyData::concentrRHS< dim >::local_rhs`

6.8.2.3 `template<int dim> FullMatrix<double> Assembly::CopyData::concentrRHS< dim >::matrix_for_bc`

The documentation for this struct was generated from the following file:

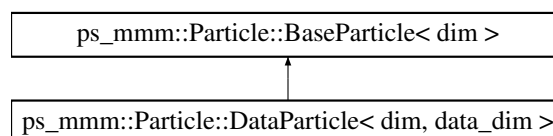
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/[assembly\\_copydata.h](#)

## 6.9 ps\_mmm::Particle::DataParticle< dim, data\_dim > Class Template Reference

[DataParticle](#) provides an example of how to extend the [BaseParticle](#) class to include related particle data.

```
#include <particle.h>
```

Inheritance diagram for ps\_mmm::Particle::DataParticle< dim, data\_dim >:



## Additional Inherited Members

### 6.9.1 Detailed Description

```
template<int dim, int data_dim>class ps_mmm::Particle::DataParticle< dim, data_dim >
```

[DataParticle](#) provides an example of how to extend the [BaseParticle](#) class to include related particle data.

This allows users to attach scalars/vectors/tensors/etc to particles and ensure they are transmitted correctly over MPI and written to output files.

The documentation for this class was generated from the following file:

- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/[particle.h](#)

## 6.10 Assembly::Scratch::diffusion\_step< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- [diffusion\\_step](#) (const FiniteElement< dim > &fe\_velocity, const Mapping< dim > &velocity\_mapping, const Quadrature< dim > &quadrature, const UpdateFlags velocity\_update\_flags, const FiniteElement< dim > &fe\_pressure, const Mapping< dim > &pressure\_mapping, const UpdateFlags pressure\_update\_flags, const FiniteElement< dim > &concentr\_fe, const Mapping< dim > &concentr\_mapping, const UpdateFlags concentr\_update\_flags)
- [diffusion\\_step](#) (const [diffusion\\_step](#) &data)

### Public Attributes

- FEValues< dim > [fe\\_velocity\\_values](#)
- FEValues< dim > [fe\\_pressure\\_values](#)
- FEValues< dim > [concentr\\_fe\\_values](#)
- std::vector< Tensor< 2, dim > > [grads\\_phi\\_u](#)
- std::vector< SymmetricTensor< 2, dim > > [symm\\_grads\\_phi\\_u](#)
- std::vector< Tensor< 1, dim > > [phi\\_u](#)
- std::vector< Tensor< 1, dim > > [divergence\\_phi\\_u](#)
- std::vector< Tensor< 1, dim > > [vel\\_star\\_values](#)
- std::vector< Tensor< 1, dim > > [vel\\_n\\_values](#)
- std::vector< Tensor< 1, dim > > [vel\\_n\\_minus\\_1\\_values](#)
- std::vector< Tensor< 2, dim > > [grad\\_vel\\_star\\_values](#)
- std::vector< Tensor< 1, dim > > [laplacian\\_vel\\_star\\_values](#)
- std::vector< Tensor< 1, dim > > [grad\\_aux\\_n\\_values](#)
- std::vector< Tensor< 1, dim > > [grad\\_aux\\_n\\_minus\\_1\\_values](#)
- std::vector< Tensor< 1, dim > > [grad\\_pre\\_n\\_values](#)
- std::vector< Tensor< 2, dim > > [grad\\_grad\\_aux\\_n\\_values](#)
- std::vector< Tensor< 2, dim > > [grad\\_grad\\_aux\\_n\\_minus\\_1\\_values](#)
- std::vector< Tensor< 2, dim > > [grad\\_grad\\_pre\\_n\\_values](#)
- std::vector< double > [aux\\_n\\_values](#)
- std::vector< double > [aux\\_n\\_minus\\_1\\_values](#)
- std::vector< double > [pre\\_n\\_values](#)
- std::vector< double > [concentr\\_values](#)
- std::vector< SymmetricTensor< 2, dim > > [symm\\_grads\\_vel\\_star](#)

## 6.10.1 Constructor &amp; Destructor Documentation

6.10.1.1 `template<int dim> Assembly::Scratch::diffusion_step< dim >::diffusion_step ( const FiniteElement< dim > & fe_velocity, const Mapping< dim > & velocity_mapping, const Quadrature< dim > & quadrature, const UpdateFlags velocity_update_flags, const FiniteElement< dim > & fe_pressure, const Mapping< dim > & pressure_mapping, const UpdateFlags pressure_update_flags, const FiniteElement< dim > & concentr_fe, const Mapping< dim > & concentr_mapping, const UpdateFlags concentr_update_flags )`

6.10.1.2 `template<int dim> Assembly::Scratch::diffusion_step< dim >::diffusion_step ( const diffusion_step< dim > & data )`

## 6.10.2 Member Data Documentation

6.10.2.1 `template<int dim> std::vector<double> Assembly::Scratch::diffusion_step< dim >::aux_n_minus_1_values`

6.10.2.2 `template<int dim> std::vector<double> Assembly::Scratch::diffusion_step< dim >::aux_n_values`

6.10.2.3 `template<int dim> FEValues<dim> Assembly::Scratch::diffusion_step< dim >::concentr_fe_values`

6.10.2.4 `template<int dim> std::vector<double> Assembly::Scratch::diffusion_step< dim >::concentr_values`

6.10.2.5 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::divergence_phi_u`

6.10.2.6 `template<int dim> FEValues<dim> Assembly::Scratch::diffusion_step< dim >::fe_pressure_values`

6.10.2.7 `template<int dim> FEValues<dim> Assembly::Scratch::diffusion_step< dim >::fe_velocity_values`

6.10.2.8 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::grad_aux_n_minus_1_values`

6.10.2.9 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::grad_aux_n_values`

6.10.2.10 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::diffusion_step< dim >::grad_grad_aux_n_minus_1_values`

6.10.2.11 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::diffusion_step< dim >::grad_grad_aux_n_values`

6.10.2.12 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::diffusion_step< dim >::grad_grad_pre_n_values`

6.10.2.13 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::grad_pre_n_values`

6.10.2.14 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::diffusion_step< dim >::grad_vel_star_values`

6.10.2.15 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::diffusion_step< dim >::grads_phi_u`

6.10.2.16 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::laplacian_vel_star_values`

6.10.2.17 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::phi_u`

6.10.2.18 `template<int dim> std::vector<double> Assembly::Scratch::diffusion_step< dim >::pre_n_values`

- 6.10.2.19 `template<int dim> std::vector<SymmetricTensor<2,dim> > Assembly::Scratch::diffusion_step< dim >::symm_grads_phi_u`
- 6.10.2.20 `template<int dim> std::vector<SymmetricTensor<2,dim> > Assembly::Scratch::diffusion_step< dim >::symm_grads_vel_star`
- 6.10.2.21 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::vel_n_minus_1_values`
- 6.10.2.22 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::vel_n_values`
- 6.10.2.23 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::diffusion_step< dim >::vel_star_values`

The documentation for this struct was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/assembly\\_copydata.h](#)

## 6.11 Assembly::CopyData::diffusion\_step< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- [diffusion\\_step](#) (const FiniteElement< dim > &fe\_velocity)
- [diffusion\\_step](#) (const [diffusion\\_step](#) &data)

### Public Attributes

- FullMatrix< double > [local\\_matrix](#)
- Vector< double > [local\\_rhs](#)
- std::vector< types::global\_dof\_index > [local\\_dof\\_indices](#)

### 6.11.1 Constructor & Destructor Documentation

- 6.11.1.1 `template<int dim> Assembly::CopyData::diffusion_step< dim >::diffusion_step ( const FiniteElement< dim > & fe_velocity )`
- 6.11.1.2 `template<int dim> Assembly::CopyData::diffusion_step< dim >::diffusion_step ( const diffusion_step< dim > & data )`

### 6.11.2 Member Data Documentation

- 6.11.2.1 `template<int dim> std::vector<types::global_dof_index> Assembly::CopyData::diffusion_step< dim >::local_dof_indices`
- 6.11.2.2 `template<int dim> FullMatrix<double> Assembly::CopyData::diffusion_step< dim >::local_matrix`
- 6.11.2.3 `template<int dim> Vector<double> Assembly::CopyData::diffusion_step< dim >::local_rhs`

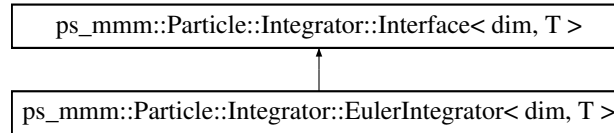
The documentation for this struct was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/assembly\\_copydata.h](#)

## 6.12 ps\_mmm::Particle::Integrator::EulerIntegrator< dim, T > Class Template Reference

Euler scheme integrator, where  $y_{n+1} = y_n + dt * v(y_n)$ .

Inheritance diagram for ps\_mmm::Particle::Integrator::EulerIntegrator< dim, T >:



### Public Member Functions

- virtual bool [integrate\\_step](#) ([Particle::World](#)< dim, T > \*world, const double dt)  
*Perform an integration step of moving the particles by the specified timestep dt.*
- virtual void [add\\_mpi\\_types](#) (std::vector< [MPIDataInfo](#) > &)  
*Secify the MPI types and data sizes involved in transferring integration related information between processes.*
- virtual unsigned int [data\\_len](#) () const  
*Return data length of the integration related data required for communication in terms of number of doubles.*
- virtual unsigned int [read\\_data](#) (const std::vector< double > &, const unsigned int &pos, const double &)  
*Read integration related data for a particle specified by id\_num from the data vector.*
- virtual void [write\\_data](#) (std::vector< double > &, const double &) const  
*Write integration related data to a vector for a particle specified by id\_num.*

### 6.12.1 Detailed Description

```
template<int dim, class T>class ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >
```

Euler scheme integrator, where  $y_{n+1} = y_n + dt * v(y_n)$ .

This requires only one step per integration, and doesn't involve any extra data.

### 6.12.2 Member Function Documentation

**6.12.2.1** `template<int dim, class T > virtual void ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >::add_mpi_types ( std::vector< MPIDataInfo > & data_info ) [inline], [virtual]`

Secify the MPI types and data sizes involved in transferring integration related information between processes.

If the integrator samples velocities at different locations and the particle moves between processes during the integration step, the sampled velocities must be transferred with the particle.

#### Parameters

in, out	<i>data_info</i>	Adds MPI data info to the specified vector indicating the quantity and type of values the integrator needs saved for this particle.
---------	------------------	---

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

**6.12.2.2** `template<int dim, class T > virtual unsigned int ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >::data_len ( ) const [inline], [virtual]`

Return data length of the integration related data required for communication in terms of number of doubles.

**Returns**

The number of doubles required to store the relevant integrator data.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

**6.12.2.3** `template<int dim, class T > virtual bool ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >::integrate_step ( Particle::World< dim, T > * world, const double dt ) [inline], [virtual]`

Perform an integration step of moving the particles by the specified timestep dt.

Implementations of this function must update the particle location. If the integrator requires multiple internal steps, this function must return true until all internal steps are finished. Between calls to this function the velocity at the updated particle positions is evaluated and passed to `integrate_step` during the next call.

**Parameters**

<i>in, out</i>	<i>world</i>	The world to integrate particles in. The particle positions will be changed in this function based on the integration scheme.
<i>in</i>	<i>dt</i>	The timestep length to perform the integration.

**Returns**

Whether this function needs to be called again (true) for additional integration steps or if all internal steps are complete (false).

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

**6.12.2.4** `template<int dim, class T > virtual unsigned int ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >::read_data ( const std::vector< double > & data, const unsigned int & pos, const double & id_num ) [inline], [virtual]`

Read integration related data for a particle specified by `id_num` from the data vector.

**Parameters**

<i>in</i>	<i>data</i>	The vector of double data to read from.
<i>in</i>	<i>pos</i>	The position in the data vector to start reading from.
<i>in</i>	<i>id_num</i>	The id number of the particle to read the data for.

**Returns**

The position in the vector of the next unread double.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

**6.12.2.5** `template<int dim, class T > virtual void ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >::write_data ( std::vector< double > & data, const double & id_num ) const [inline], [virtual]`

Write integration related data to a vector for a particle specified by `id_num`.

**Parameters**

<i>in, out</i>	<i>data</i>	The vector of doubles to write integrator data into.
<i>in</i>	<i>id_num</i>	The id number of the particle to read the data for.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

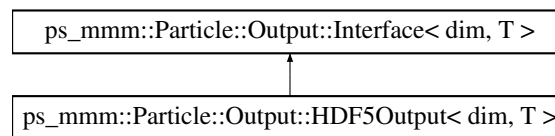
The documentation for this class was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/particle/integrator.cc](#)



## 6.13 ps\_mmm::Particle::Output::HDF5Output< dim, T > Class Template Reference

Inheritance diagram for ps\_mmm::Particle::Output::HDF5Output< dim, T >:



### Public Member Functions

- [HDF5Output](#) (const std::string &output\_directory, const MPI\_Comm communicator)  
*Constructor.*
- virtual std::string [output\\_particle\\_data](#) (const std::multimap< [Levellnd](#), T > &particles, const double &current\_time)  
*Write data about the particles specified in the first argument to a file.*

### Additional Inherited Members

#### 6.13.1 Constructor & Destructor Documentation

6.13.1.1 `template<int dim, class T > ps_mmm::Particle::Output::HDF5Output< dim, T >::HDF5Output ( const std::string & output_directory, const MPI_Comm communicator ) [inline]`

Constructor.

#### Parameters

in	<i>The</i>	directory into which output files shall be placed.
in	<i>The</i>	MPI communicator that describes this simulation.

#### 6.13.2 Member Function Documentation

6.13.2.1 `template<int dim, class T > virtual std::string ps_mmm::Particle::Output::HDF5Output< dim, T >::output_particle_data ( const std::multimap< Levellnd, T > & particles, const double & current_time ) [inline], [virtual]`

Write data about the particles specified in the first argument to a file.

If possible, encode the current simulation time into this file using the data provided in the second argument.

#### Parameters

in	<i>particles</i>	The set of particles to generate a graphical representation for
in	<i>current_time</i>	Current time of the simulation, given as either years or seconds, as selected in the input file. In other words, output writers do not need to know the units in which time is described.

#### Returns

The name of the file that was written, or any other information that describes what output was produced if for example multiple files were created.

Implements [ps\\_mmm::Particle::Output::Interface< dim, T >](#).

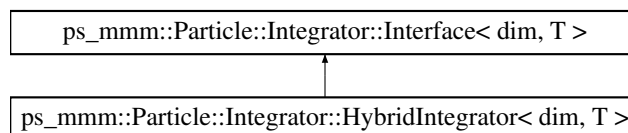
The documentation for this class was generated from the following file:

- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/particle/output.cc

## 6.14 ps\_mmm::Particle::Integrator::HybridIntegrator< dim, T > Class Template Reference

[Integrator](#) which chooses Euler, RK2 or RK4 depending on characteristics of the cell a particle is in.

Inheritance diagram for ps\_mmm::Particle::Integrator::HybridIntegrator< dim, T >:



### Public Member Functions

- [HybridIntegrator](#) ()
- virtual bool [integrate\\_step](#) ([Particle::World](#)< dim, T > \*world, const double dt)  
*Perform an integration step of moving the particles by the specified timestep dt.*
- virtual void [add\\_mpi\\_types](#) (std::vector< [MPIDataInfo](#) > &data\_info)  
*Secify the MPI types and data sizes involved in transferring integration related information between processes.*
- virtual unsigned int [data\\_len](#) () const  
*Return data length of the integration related data required for communication in terms of number of doubles.*
- virtual unsigned int [read\\_data](#) (const std::vector< double > &data, const unsigned int &pos, const double &id\_num)  
*Read integration related data for a particle specified by id\_num from the data vector.*
- virtual void [write\\_data](#) (std::vector< double > &data, const double &id\_num) const  
*Write integration related data to a vector for a particle specified by id\_num.*

### 6.14.1 Detailed Description

```
template<int dim, class T>class ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >
```

[Integrator](#) which chooses Euler, RK2 or RK4 depending on characteristics of the cell a particle is in.

Currently used for research only.

### 6.14.2 Constructor & Destructor Documentation

6.14.2.1 `template<int dim, class T > ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >::HybridIntegrator ( ) [inline]`

### 6.14.3 Member Function Documentation

6.14.3.1 `template<int dim, class T > virtual void ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >::add_mpi_types ( std::vector< MPIDataInfo > & data_info ) [inline],[virtual]`

Secify the MPI types and data sizes involved in transferring integration related information between processes.

If the integrator samples velocities at different locations and the particle moves between processes during the integration step, the sampled velocities must be transferred with the particle.

#### Parameters

<code>in, out</code>	<code>data_info</code>	Adds MPI data info to the specified vector indicating the quantity and type of values the integrator needs saved for this particle.
----------------------	------------------------	---

Implements `ps_mmm::Particle::Integrator::Interface< dim, T >`.

6.14.3.2 `template<int dim, class T> virtual unsigned int ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >::data_len ( ) const [inline], [virtual]`

Return data length of the integration related data required for communication in terms of number of doubles.

#### Returns

The number of doubles required to store the relevant integrator data.

Implements `ps_mmm::Particle::Integrator::Interface< dim, T >`.

6.14.3.3 `template<int dim, class T> virtual bool ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >::integrate_step ( Particle::World< dim, T > * world, const double dt ) [inline], [virtual]`

Perform an integration step of moving the particles by the specified timestep `dt`.

Implementations of this function must update the particle location. If the integrator requires multiple internal steps, this function must return true until all internal steps are finished. Between calls to this function the velocity at the updated particle positions is evaluated and passed to `integrate_step` during the next call.

#### Parameters

<code>in, out</code>	<code>world</code>	The world to integrate particles in. The particle positions will be changed in this function based on the integration scheme.
<code>in</code>	<code>dt</code>	The timestep length to perform the integration.

#### Returns

Whether this function needs to be called again (true) for additional integration steps or if all internal steps are complete (false).

Implements `ps_mmm::Particle::Integrator::Interface< dim, T >`.

6.14.3.4 `template<int dim, class T> virtual unsigned int ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >::read_data ( const std::vector< double > & data, const unsigned int & pos, const double & id_num ) [inline], [virtual]`

Read integration related data for a particle specified by `id_num` from the data vector.

#### Parameters

<code>in</code>	<code>data</code>	The vector of double data to read from.
<code>in</code>	<code>pos</code>	The position in the data vector to start reading from.
<code>in</code>	<code>id_num</code>	The id number of the particle to read the data for.

#### Returns

The position in the vector of the next unread double.

Implements `ps_mmm::Particle::Integrator::Interface< dim, T >`.

6.14.3.5 `template<int dim, class T> virtual void ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >::write_data ( std::vector< double > & data, const double & id_num ) const [inline], [virtual]`

Write integration related data to a vector for a particle specified by `id_num`.

## Parameters

<code>in, out</code>	<code>data</code>	The vector of doubles to write integrator data into.
<code>in</code>	<code>id_num</code>	The id number of the particle to read the data for.

Implements `ps_mmm::Particle::Integrator::Interface< dim, T >`.

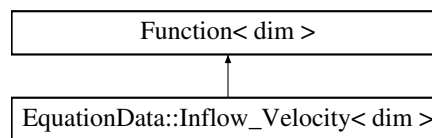
The documentation for this class was generated from the following file:

- `/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/source/particle/integrator.cc`

## 6.15 EquationData::Inflow\_Velocity< dim > Class Template Reference

```
#include <equation_data.h>
```

Inheritance diagram for `EquationData::Inflow_Velocity< dim >`:



## Public Member Functions

- `Inflow_Velocity` (double, unsigned int)
- virtual double `value` (const Point< dim > &p, const unsigned int component=0) const
- virtual void `vector_value` (const Point< dim > &p, Vector< double > &value) const
- virtual void `vector_value_list` (const std::vector< Point< dim > > &p, std::vector< Vector< double > > &values) const

## Public Attributes

- double `init_mean_vel`
- unsigned int `which_inflow_type`

### 6.15.1 Constructor & Destructor Documentation

6.15.1.1 `template<int dim> EquationData::Inflow_Velocity< dim >::Inflow_Velocity ( double init_mean_vel, unsigned int which_inflow_type )`

### 6.15.2 Member Function Documentation

6.15.2.1 `template<int dim> double EquationData::Inflow_Velocity< dim >::value ( const Point< dim > &p, const unsigned int component = 0 ) const` [virtual]

6.15.2.2 `template<int dim> void EquationData::Inflow_Velocity< dim >::vector_value ( const Point< dim > &p, Vector< double > &value ) const` [virtual]

6.15.2.3 `template<int dim> void EquationData::Inflow_Velocity< dim >::vector_value_list ( const std::vector< Point< dim > > &p, std::vector< Vector< double > > &values ) const` [virtual]

### 6.15.3 Member Data Documentation

6.15.3.1 `template<int dim> double EquationData::Inflow_Velocity< dim >::init_mean_vel`

## 6.15.3.2 template&lt;int dim&gt; unsigned int EquationData::Inflow\_Velocity&lt; dim &gt;::which\_inflow\_type

The documentation for this class was generated from the following file:

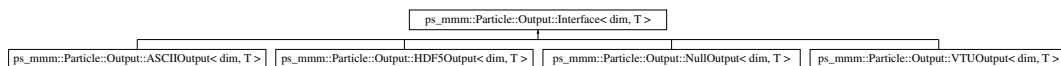
- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/equation\_data.h

## 6.16 ps\_mmm::Particle::Output::Interface&lt; dim, T &gt; Class Template Reference

Abstract base class used for classes that generate particle output.

```
#include <output.h>
```

Inheritance diagram for ps\_mmm::Particle::Output::Interface< dim, T >:



## Public Member Functions

- [Interface](#) (const std::string &output\_directory, const MPI\_Comm [communicator](#))  
*Constructor.*
- virtual [~Interface](#) ()  
*Destructor.*
- virtual std::string [output\\_particle\\_data](#) (const std::multimap< [Levellnd](#), T > &particles, const double &current\_time)=0  
*Write data about the particles specified in the first argument to a file.*
- template<class Archive >  
void [serialize](#) (Archive &ar, const unsigned int version)  
*Read or write the data of this object for serialization.*

## Protected Attributes

- const std::string [output\\_dir](#)  
*Path to directory in which to put particle output files.*
- MPI\_Comm [communicator](#)  
*MPI communicator to be used for output synchronization.*
- unsigned int [file\\_index](#)  
*Internal index of file output number, must be incremented by derived classes when they create a new file.*

## 6.16.1 Detailed Description

```
template<int dim, class T>class ps_mmm::Particle::Output::Interface< dim, T >
```

Abstract base class used for classes that generate particle output.

## 6.16.2 Constructor &amp; Destructor Documentation

```
6.16.2.1 template<int dim, class T> ps_mmm::Particle::Output::Interface< dim, T >::Interface ( const std::string &
output_directory, const MPI_Comm communicator ) [inline]
```

Constructor.

**Parameters**

in	<i>output_directory</i>	The directory into which output files shall be placed.
in	<i>communicator</i>	The MPI communicator that describes this simulation.

6.16.2.2 `template<int dim, class T> virtual ps_mmm::Particle::Output::Interface< dim, T>::~~Interface ( )`  
`[inline], [virtual]`

Destructor.

Made virtual so that derived classes can be created and destroyed through pointers to the base class.

**6.16.3 Member Function Documentation**

6.16.3.1 `template<int dim, class T> virtual std::string ps_mmm::Particle::Output::Interface< dim, T>::output_particle_data ( const std::multimap< LevelInd, T> & particles, const double & current_time )` `[pure virtual]`

Write data about the particles specified in the first argument to a file.

If possible, encode the current simulation time into this file using the data provided in the second argument.

**Parameters**

in	<i>particles</i>	The set of particles to generate a graphical representation for
in	<i>current_time</i>	Current time of the simulation, given as either years or seconds, as selected in the input file. In other words, output writers do not need to know the units in which time is described.

**Returns**

The name of the file that was written, or any other information that describes what output was produced if for example multiple files were created.

Implemented in `ps_mmm::Particle::Output::HDF5Output< dim, T>`, `ps_mmm::Particle::Output::VTUOutput< dim, T>`, `ps_mmm::Particle::Output::ASCIIOutput< dim, T>`, and `ps_mmm::Particle::Output::NullOutput< dim, T>`.

6.16.3.2 `template<int dim, class T> template<class Archive> void ps_mmm::Particle::Output::Interface< dim, T>::serialize ( Archive & ar, const unsigned int version )` `[inline]`

Read or write the data of this object for serialization.

**6.16.4 Member Data Documentation**

6.16.4.1 `template<int dim, class T> MPI_Comm ps_mmm::Particle::Output::Interface< dim, T>::communicator`  
`[protected]`

MPI communicator to be used for output synchronization.

6.16.4.2 `template<int dim, class T> unsigned int ps_mmm::Particle::Output::Interface< dim, T>::file_index`  
`[protected]`

Internal index of file output number, must be incremented by derived classes when they create a new file.

6.16.4.3 `template<int dim, class T> const std::string ps_mmm::Particle::Output::Interface< dim, T>::output_dir`  
`[protected]`

Path to directory in which to put particle output files.

The documentation for this class was generated from the following file:

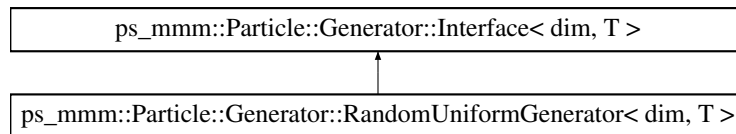
- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/particle/output.h](/Users/miranus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/particle/output.h)

## 6.17 ps\_mmm::Particle::Generator::Interface< dim, T > Class Template Reference

Abstract base class used for classes that generate particles.

```
#include <generator.h>
```

Inheritance diagram for ps\_mmm::Particle::Generator::Interface< dim, T >:



### Public Member Functions

- [Interface](#) ()  
*Constructor.*
- virtual [~Interface](#) ()  
*Destructor.*
- virtual void [generate\\_particles](#) ([Particle::World](#)< dim, T > &world, const double total\_num\_particles)=0  
*Generate a specified number of particles in the specified world using the type of generation function implemented by this [Generator](#).*

#### 6.17.1 Detailed Description

```
template<int dim, class T>class ps_mmm::Particle::Generator::Interface< dim, T >
```

Abstract base class used for classes that generate particles.

#### 6.17.2 Constructor & Destructor Documentation

6.17.2.1 `template<int dim, class T > ps_mmm::Particle::Generator::Interface< dim, T >::Interface ( )`  
[inline]

Constructor.

6.17.2.2 `template<int dim, class T > virtual ps_mmm::Particle::Generator::Interface< dim, T >::~~Interface ( )`  
[inline], [virtual]

Destructor.

Made virtual so that derived classes can be created and destroyed through pointers to the base class.

#### 6.17.3 Member Function Documentation

6.17.3.1 `template<int dim, class T > virtual void ps_mmm::Particle::Generator::Interface< dim, T >::generate_particles ( Particle::World< dim, T > & world, const double total_num_particles )` [pure virtual]

Generate a specified number of particles in the specified world using the type of generation function implemented by this [Generator](#).

## Parameters

in	<i>world</i>	The particle world the particles will exist in
in	<i>total_num_↔ particles</i>	Total number of particles to generate. The actual number of generated particles may differ, for example if the generator reads particles from a file this parameter may be ignored.

Implemented in [ps\\_mmm::Particle::Generator::RandomUniformGenerator< dim, T >](#).

The documentation for this class was generated from the following file:

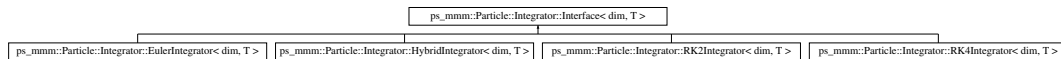
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/[generator.h](#)

## 6.18 ps\_mmm::Particle::Integrator::Interface< dim, T > Class Template Reference

An abstract class defining virtual methods for performing integration of particle paths through the simulation velocity field.

```
#include <integrator.h>
```

Inheritance diagram for ps\_mmm::Particle::Integrator::Interface< dim, T >:



## Public Member Functions

- [Interface](#) (void)  
*Constructor.*
- virtual [~Interface](#) ()  
*Destructor.*
- virtual bool [integrate\\_step](#) ([Particle::World](#)< dim, T > \*world, const double dt)=0  
*Perform an integration step of moving the particles by the specified timestep dt.*
- virtual void [add\\_mpi\\_types](#) (std::vector< [MPIDataInfo](#) > &data\_info)=0  
*Specify the MPI types and data sizes involved in transferring integration related information between processes.*
- virtual unsigned int [data\\_len](#) () const =0  
*Return data length of the integration related data required for communication in terms of number of doubles.*
- virtual unsigned int [read\\_data](#) (const std::vector< double > &data, const unsigned int &pos, const double &id\_num)=0  
*Read integration related data for a particle specified by id\_num from the data vector.*
- virtual void [write\\_data](#) (std::vector< double > &data, const double &id\_num) const =0  
*Write integration related data to a vector for a particle specified by id\_num.*

### 6.18.1 Detailed Description

```
template<int dim, class T>class ps_mmm::Particle::Integrator::Interface< dim, T >
```

An abstract class defining virtual methods for performing integration of particle paths through the simulation velocity field.

### 6.18.2 Constructor & Destructor Documentation

6.18.2.1 `template<int dim, class T> ps_mmm::Particle::Integrator::Interface< dim, T >::Interface ( void )`  
[inline]

Constructor.



6.18.2.2 `template<int dim, class T> virtual ps_mmm::Particle::Integrator::Interface< dim, T >::~~Interface ( )`  
`[inline], [virtual]`

Destructor.

Made virtual so that derived classes can be created and destroyed through pointers to the base class.

### 6.18.3 Member Function Documentation

6.18.3.1 `template<int dim, class T> virtual void ps_mmm::Particle::Integrator::Interface< dim, T >::add_mpi_types (`  
`std::vector< MPIDataInfo > & data_info ) [pure virtual]`

Specify the MPI types and data sizes involved in transferring integration related information between processes.

If the integrator samples velocities at different locations and the particle moves between processes during the integration step, the sampled velocities must be transferred with the particle.

#### Parameters

<code>in, out</code>	<code>data_info</code>	Adds MPI data info to the specified vector indicating the quantity and type of values the integrator needs saved for this particle.
----------------------	------------------------	---

Implemented in `ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >`, `ps_mmm::Particle::Integrator::RK4Integrator< dim, T >`, `ps_mmm::Particle::Integrator::RK2Integrator< dim, T >`, and `ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >`.

6.18.3.2 `template<int dim, class T> virtual unsigned int ps_mmm::Particle::Integrator::Interface< dim, T >::data_len ( ) const [pure virtual]`

Return data length of the integration related data required for communication in terms of number of doubles.

#### Returns

The number of doubles required to store the relevant integrator data.

Implemented in `ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >`, `ps_mmm::Particle::Integrator::RK4Integrator< dim, T >`, `ps_mmm::Particle::Integrator::RK2Integrator< dim, T >`, and `ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >`.

6.18.3.3 `template<int dim, class T> virtual bool ps_mmm::Particle::Integrator::Interface< dim, T >::integrate_step (`  
`Particle::World< dim, T > * world, const double dt ) [pure virtual]`

Perform an integration step of moving the particles by the specified timestep `dt`.

Implementations of this function must update the particle location. If the integrator requires multiple internal steps, this function must return true until all internal steps are finished. Between calls to this function the velocity at the updated particle positions is evaluated and passed to `integrate_step` during the next call.

#### Parameters

<code>in, out</code>	<code>world</code>	The world to integrate particles in. The particle positions will be changed in this function based on the integration scheme.
<code>in</code>	<code>dt</code>	The timestep length to perform the integration.

#### Returns

Whether this function needs to be called again (true) for additional integration steps or if all internal steps are complete (false).

Implemented in `ps_mmm::Particle::Integrator::HybridIntegrator< dim, T >`, `ps_mmm::Particle::Integrator::RK4Integrator< dim, T >`, `ps_mmm::Particle::Integrator::RK2Integrator< dim, T >`, and `ps_mmm::Particle::Integrator::EulerIntegrator< dim, T >`.

6.18.3.4 `template<int dim, class T> virtual unsigned int ps_mmm::Particle::Integrator::Interface< dim, T >::read_data ( const std::vector< double > & data, const unsigned int & pos, const double & id_num ) [pure virtual]`

Read integration related data for a particle specified by `id_num` from the data vector.

## Parameters

in	<i>data</i>	The vector of double data to read from.
in	<i>pos</i>	The position in the data vector to start reading from.
in	<i>id_num</i>	The id number of the particle to read the data for.

## Returns

The position in the vector of the next unread double.

Implemented in [ps\\_mmm::Particle::Integrator::HybridIntegrator< dim, T >](#), [ps\\_mmm::Particle::Integrator::RK4Integrator< dim, T >](#), [ps\\_mmm::Particle::Integrator::RK2Integrator< dim, T >](#), and [ps\\_mmm::Particle::Integrator::EulerIntegrator< dim, T >](#).

6.18.3.5 `template<int dim, class T> virtual void ps_mmm::Particle::Integrator::Interface< dim, T >::write_data ( std::vector< double > & data, const double & id_num ) const` [pure virtual]

Write integration related data to a vector for a particle specified by id\_num.

## Parameters

in, out	<i>data</i>	The vector of doubles to write integrator data into.
in	<i>id_num</i>	The id number of the particle to read the data for.

Implemented in [ps\\_mmm::Particle::Integrator::HybridIntegrator< dim, T >](#), [ps\\_mmm::Particle::Integrator::RK4Integrator< dim, T >](#), [ps\\_mmm::Particle::Integrator::RK2Integrator< dim, T >](#), and [ps\\_mmm::Particle::Integrator::EulerIntegrator< dim, T >](#).

The documentation for this class was generated from the following file:

- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/integrator.h

## 6.19 ps\_mmm::immiscible::Levelset&lt; dim &gt; Class Template Reference

```
#include <levelset.h>
```

## Public Member Functions

- [Levelset](#) (ParameterHandler &)
- [~Levelset](#) ()
- void [initial\\_distribution](#) ()
- void [get\\_parameters](#) ()
- void [advection\\_equation](#) ()
- void [compute\\_normal\\_vector](#) (unsigned int)
- void [reinitialization\\_levelset](#) ()

## 6.19.1 Constructor &amp; Destructor Documentation

6.19.1.1 `template<int dim> ps_mmm::immiscible::Levelset< dim >::Levelset ( ParameterHandler & )`

6.19.1.2 `template<int dim> ps_mmm::immiscible::Levelset< dim >::~~Levelset ( )`

## 6.19.2 Member Function Documentation

6.19.2.1 `template<int dim> void ps_mmm::immiscible::Levelset< dim >::advection_equation ( )`

6.19.2.2 `template<int dim> void ps_mmm::immiscible::Levelset< dim >::compute_normal_vector ( unsigned int current_dimension )`

6.19.2.3 `template<int dim> void ps_mmm::immicible::Levelset< dim >::get_parameters ( )`

6.19.2.4 `template<int dim> void ps_mmm::immicible::Levelset< dim >::initial_distribution ( )`

6.19.2.5 `template<int dim> void ps_mmm::immicible::Levelset< dim >::reinitialization_levelset ( )`

The documentation for this class was generated from the following files:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/phase/levelset.h](#)
- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/solver/phase/levelset.cc](#)

## 6.20 ps\_mmm::Particle::MPIDataInfo Class Reference

```
#include <particle.h>
```

### Public Member Functions

- [MPIDataInfo](#) (std::string [name](#), unsigned int num\_elems)

### Public Attributes

- std::string [name](#)
- unsigned int [n\\_elements](#)

### 6.20.1 Constructor & Destructor Documentation

6.20.1.1 `ps_mmm::Particle::MPIDataInfo::MPIDataInfo ( std::string name, unsigned int num_elems ) [inline]`

### 6.20.2 Member Data Documentation

6.20.2.1 `unsigned int ps_mmm::Particle::MPIDataInfo::n_elements`

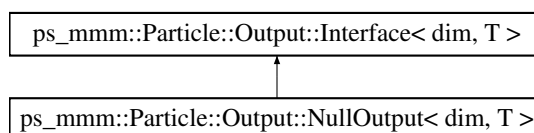
6.20.2.2 `std::string ps_mmm::Particle::MPIDataInfo::name`

The documentation for this class was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/particle/particle.h](#)

## 6.21 ps\_mmm::Particle::Output::NullOutput< dim, T > Class Template Reference

Inheritance diagram for `ps_mmm::Particle::Output::NullOutput< dim, T >`:



### Public Member Functions

- [NullOutput](#) (const std::string &output\_directory, const MPI\_Comm [communicator](#))  
*Constructor.*
- virtual std::string [output\\_particle\\_data](#) (const std::multimap< [LevelInd](#), T > &, const double &)  
*Write data about the particles specified in the first argument to a file.*

## Additional Inherited Members

## 6.21.1 Constructor &amp; Destructor Documentation

6.21.1.1 `template<int dim, class T > ps_mmm::Particle::Output::NullOutput< dim, T >::NullOutput ( const std::string & output_directory, const MPI_Comm communicator ) [inline]`

Constructor.

## Parameters

<i>in</i>	<i>The</i>	directory into which output files shall be placed.
<i>in</i>	<i>The</i>	MPI communicator that describes this simulation.

## 6.21.2 Member Function Documentation

6.21.2.1 `template<int dim, class T > virtual std::string ps_mmm::Particle::Output::NullOutput< dim, T >::output_particle_data ( const std::multimap< LevelInd, T > &, const double & ) [inline],[virtual]`

Write data about the particles specified in the first argument to a file.

If possible, encode the current simulation time into this file using the data provided in the second argument.

## Parameters

<i>in</i>	<i>particles</i>	The set of particles to generate a graphical representation for
<i>in</i>	<i>current_time</i>	Current time of the simulation, given as either years or seconds, as selected in the input file. In other words, output writers do not need to know the units in which time is described.

## Returns

The name of the file that was written, or any other information that describes what output was produced if for example multiple files were created.

Implements [ps\\_mmm::Particle::Output::Interface< dim, T >](#).

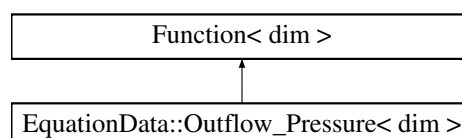
The documentation for this class was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/particle/output.cc](#)

## 6.22 EquationData::Outflow\_Pressure&lt; dim &gt; Class Template Reference

```
#include <equation_data.h>
```

Inheritance diagram for EquationData::Outflow\_Pressure< dim >:



## Public Member Functions

- [Outflow\\_Pressure](#) (double, double)
- virtual double [value](#) (const Point< dim > &p, const unsigned int component=0) const
- virtual void [vector\\_value](#) (const Point< dim > &p, Vector< double > &[value](#)) const

- virtual void [vector\\_value\\_list](#) (const std::vector< Point< dim > > &p, std::vector< Vector< double > > &values) const

#### Public Attributes

- double [inclined\\_angle](#)
- double [Froude\\_number](#)

#### 6.22.1 Constructor & Destructor Documentation

- 6.22.1.1 `template<int dim> EquationData::Outflow_Pressure< dim >::Outflow_Pressure ( double inclined_angle, double Froude_number )`

#### 6.22.2 Member Function Documentation

- 6.22.2.1 `template<int dim> double EquationData::Outflow_Pressure< dim >::value ( const Point< dim > &p, const unsigned int component = 0 ) const` [virtual]
- 6.22.2.2 `template<int dim> void EquationData::Outflow_Pressure< dim >::vector_value ( const Point< dim > &p, Vector< double > &value ) const` [virtual]
- 6.22.2.3 `template<int dim> void EquationData::Outflow_Pressure< dim >::vector_value_list ( const std::vector< Point< dim > > &p, std::vector< Vector< double > > &values ) const` [virtual]

#### 6.22.3 Member Data Documentation

- 6.22.3.1 `template<int dim> double EquationData::Outflow_Pressure< dim >::Froude_number`
- 6.22.3.2 `template<int dim> double EquationData::Outflow_Pressure< dim >::inclined_angle`

The documentation for this class was generated from the following file:

- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/[equation\\_data.h](#)

### 6.23 UBC\_mis\_mixing< dim >::Parameters Struct Reference

```
#include <class.h>
```

#### Public Member Functions

- [Parameters](#) (std::string &parameters\_filename)
- void [parse\\_parameters](#) (ParameterHandler &prm)

#### Static Public Member Functions

- static void [declare\\_parameters](#) (ParameterHandler &prm)

#### Public Attributes

- std::string [input\\_mesh\\_file](#)
- Point< dim > [length\\_of\\_domain](#)
- std::vector< double > [domain\\_size](#)
- std::vector< double > [domain\\_boundary](#)
- unsigned int [num\\_element\\_size](#)

- unsigned int [flow\\_direction](#)
- unsigned int [depth\\_direction](#)
- unsigned int [latitude\\_direction](#)
- unsigned int [num\\_slices\\_domain](#)
- bool [is\\_symmetry\\_boundary](#)
- unsigned int [max\\_grid\\_level](#)
- unsigned int [type\\_adaptivity\\_rule](#)
- double [error\\_threshold](#)
- double [ref\\_crit](#)
- double [coar\\_crit](#)
- unsigned int [no\\_refine\\_period](#)
- unsigned int [intial\\_ratio\\_refinement](#)
- double [stabilization\\_alpha](#)
- double [stabilization\\_beta](#)
- double [stabilization\\_c\\_R](#)
- unsigned int [ist\\_optimization\\_method](#)
- unsigned int [ist\\_projection\\_method](#)
- unsigned int [ist\\_pressure\\_boundary](#)
- unsigned int [ist\\_flow\\_source](#)
- bool [ist\\_uniform\\_flow](#)
- double [coeff\\_relax\\_div\\_velocity](#)
- unsigned int [no\\_steps\\_for\\_buffering](#)
- double [mesh\\_speed](#)
- unsigned int [dir\\_concentration](#)
- unsigned int [which\\_method\\_for\\_c](#)
- unsigned int [which\\_interpl\\_c](#)
- bool [ist\\_add\\_reinit](#)
- double [coeff\\_gamma\\_grad\\_div](#)
- double [coeff\\_arti\\_viscosity](#)
- double [maximum\\_coeff\\_arti\\_viscosity](#)
- bool [exclude\\_depth\\_direction](#)
- bool [is\\_verbal\\_output](#)
- double [CFL\\_number](#)
- double [init\\_sep\\_x](#)
- double [inclined\\_angle](#)
- double [Atwood\\_number](#)
- double [mean\\_velocity\\_inlet](#)
- double [inlet\\_pressure](#)
- double [viscosity\\_ratio](#)
- double [computed\\_time\\_step](#)
- double [Reynolds\\_number](#)
- double [Froude\\_number](#)
- double [reference\\_length](#)
- double [reference\\_time](#)
- double [reference\\_velocity](#)
- bool [is\\_density\\_stable\\_flow](#)
- double [upstream\\_concentr](#)
- double [downstream\\_concentr](#)
- double [mean\\_viscosity](#)
- double [ratio\\_pow\\_law](#)
- double [n\\_pow\\_law](#)
- Point< dim > [inclined\\_angle\\_vector](#)
- double [tau\\_step](#)
- double [eps\\_v\\_concentr](#)
- unsigned int [degree\\_of\\_velocity](#)

- unsigned int [degree\\_of\\_pressure](#)
- unsigned int [degree\\_of\\_concentr](#)
- unsigned int [data\\_id](#)
- unsigned int [output\\_fac\\_vtu](#)
- unsigned int [output\\_fac\\_data](#)
- unsigned int [number\\_slices\\_coarse\\_mesh](#)
- bool [is\\_restart](#)
- unsigned int [save\\_fac\\_period](#)
- unsigned int [index\\_for\\_restart](#)
- unsigned int [restart\\_no\\_timestep](#)
- double [check\\_total\\_time](#)
- double [check\\_total\\_real\\_time](#)
- double [check\\_current\\_time\\_step](#)
- double [check\\_old\\_time\\_step](#)
- double [eps\\_ns](#)
- double [eps\\_c](#)
- unsigned int [kry\\_size](#)
- unsigned int [no\\_test\\_case](#)

### 6.23.1 Constructor & Destructor Documentation

6.23.1.1 `template<int dim> UBC_mis_mixing< dim >::Parameters::Parameters ( std::string & parameters_filename )`

### 6.23.2 Member Function Documentation

6.23.2.1 `template<int dim> void UBC_mis_mixing< dim >::Parameters::declare_parameters ( ParameterHandler & prm )`  
`[static]`

6.23.2.2 `template<int dim> void UBC_mis_mixing< dim >::Parameters::parse_parameters ( ParameterHandler & prm )`

### 6.23.3 Member Data Documentation

6.23.3.1 `template<int dim> double UBC_mis_mixing< dim >::Parameters::Atwood_number`

6.23.3.2 `template<int dim> double UBC_mis_mixing< dim >::Parameters::CFL_number`

6.23.3.3 `template<int dim> double UBC_mis_mixing< dim >::Parameters::check_current_time_step`

6.23.3.4 `template<int dim> double UBC_mis_mixing< dim >::Parameters::check_old_time_step`

6.23.3.5 `template<int dim> double UBC_mis_mixing< dim >::Parameters::check_total_real_time`

6.23.3.6 `template<int dim> double UBC_mis_mixing< dim >::Parameters::check_total_time`

6.23.3.7 `template<int dim> double UBC_mis_mixing< dim >::Parameters::coar_crit`

6.23.3.8 `template<int dim> double UBC_mis_mixing< dim >::Parameters::coeff_arti_viscosity`

6.23.3.9 `template<int dim> double UBC_mis_mixing< dim >::Parameters::coeff_gamma_grad_div`

6.23.3.10 `template<int dim> double UBC_mis_mixing< dim >::Parameters::coeff_relax_div_velocity`

6.23.3.11 `template<int dim> double UBC_mis_mixing< dim >::Parameters::computed_time_step`

6.23.3.12 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::data_id`

6.23.3.13 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::degree_of_concentr`



- 6.23.3.14 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::degree_of_pressure`
- 6.23.3.15 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::degree_of_velocity`
- 6.23.3.16 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::depth_direction`
- 6.23.3.17 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::dir_concentration`
- 6.23.3.18 `template<int dim> std::vector<double> UBC_mis_mixing< dim >::Parameters::domain_boundary`
- 6.23.3.19 `template<int dim> std::vector<double> UBC_mis_mixing< dim >::Parameters::domain_size`
- 6.23.3.20 `template<int dim> double UBC_mis_mixing< dim >::Parameters::downstream_concentr`
- 6.23.3.21 `template<int dim> double UBC_mis_mixing< dim >::Parameters::eps_c`
- 6.23.3.22 `template<int dim> double UBC_mis_mixing< dim >::Parameters::eps_ns`
- 6.23.3.23 `template<int dim> double UBC_mis_mixing< dim >::Parameters::eps_v_concentr`
- 6.23.3.24 `template<int dim> double UBC_mis_mixing< dim >::Parameters::error_threshold`
- 6.23.3.25 `template<int dim> bool UBC_mis_mixing< dim >::Parameters::exclude_depth_direction`
- 6.23.3.26 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::flow_direction`
- 6.23.3.27 `template<int dim> double UBC_mis_mixing< dim >::Parameters::Froude_number`
- 6.23.3.28 `template<int dim> double UBC_mis_mixing< dim >::Parameters::inclined_angle`
- 6.23.3.29 `template<int dim> Point<dim> UBC_mis_mixing< dim >::Parameters::inclined_angle_vector`
- 6.23.3.30 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::index_for_restart`
- 6.23.3.31 `template<int dim> double UBC_mis_mixing< dim >::Parameters::init_sep_x`
- 6.23.3.32 `template<int dim> double UBC_mis_mixing< dim >::Parameters::inlet_pressure`
- 6.23.3.33 `template<int dim> std::string UBC_mis_mixing< dim >::Parameters::input_mesh_file`
- 6.23.3.34 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::intial_ratio_refinement`
- 6.23.3.35 `template<int dim> bool UBC_mis_mixing< dim >::Parameters::is_density_stable_flow`
- 6.23.3.36 `template<int dim> bool UBC_mis_mixing< dim >::Parameters::is_restart`
- 6.23.3.37 `template<int dim> bool UBC_mis_mixing< dim >::Parameters::is_symmetry_boundary`
- 6.23.3.38 `template<int dim> bool UBC_mis_mixing< dim >::Parameters::is_verbal_output`
- 6.23.3.39 `template<int dim> bool UBC_mis_mixing< dim >::Parameters::ist_add_reinit`
- 6.23.3.40 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::ist_flow_source`
- 6.23.3.41 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::ist_optimization_method`
- 6.23.3.42 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::ist_pressure_boundary`
- 6.23.3.43 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::ist_projection_method`
- 6.23.3.44 `template<int dim> bool UBC_mis_mixing< dim >::Parameters::ist_uniform_flow`

- 6.23.3.45 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::kry_size`
- 6.23.3.46 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::latitude_direction`
- 6.23.3.47 `template<int dim> Point<dim> UBC_mis_mixing< dim >::Parameters::length_of_domain`
- 6.23.3.48 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::max_grid_level`
- 6.23.3.49 `template<int dim> double UBC_mis_mixing< dim >::Parameters::maximum_coeff_arti_viscosity`
- 6.23.3.50 `template<int dim> double UBC_mis_mixing< dim >::Parameters::mean_velocity_inlet`
- 6.23.3.51 `template<int dim> double UBC_mis_mixing< dim >::Parameters::mean_viscosity`
- 6.23.3.52 `template<int dim> double UBC_mis_mixing< dim >::Parameters::mesh_speed`
- 6.23.3.53 `template<int dim> double UBC_mis_mixing< dim >::Parameters::n_pow_law`
- 6.23.3.54 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::no_refine_period`
- 6.23.3.55 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::no_steps_for_buffering`
- 6.23.3.56 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::no_test_case`
- 6.23.3.57 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::num_element_size`
- 6.23.3.58 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::num_slices_domain`
- 6.23.3.59 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::number_slices_coarse_mesh`
- 6.23.3.60 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::output_fac_data`
- 6.23.3.61 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::output_fac_vtu`
- 6.23.3.62 `template<int dim> double UBC_mis_mixing< dim >::Parameters::ratio_pow_law`
- 6.23.3.63 `template<int dim> double UBC_mis_mixing< dim >::Parameters::ref_crit`
- 6.23.3.64 `template<int dim> double UBC_mis_mixing< dim >::Parameters::reference_length`
- 6.23.3.65 `template<int dim> double UBC_mis_mixing< dim >::Parameters::reference_time`
- 6.23.3.66 `template<int dim> double UBC_mis_mixing< dim >::Parameters::reference_velocity`
- 6.23.3.67 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::restart_no_timestep`
- 6.23.3.68 `template<int dim> double UBC_mis_mixing< dim >::Parameters::Reynolds_number`
- 6.23.3.69 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::save_fac_period`
- 6.23.3.70 `template<int dim> double UBC_mis_mixing< dim >::Parameters::stabilization_alpha`
- 6.23.3.71 `template<int dim> double UBC_mis_mixing< dim >::Parameters::stabilization_beta`
- 6.23.3.72 `template<int dim> double UBC_mis_mixing< dim >::Parameters::stabilization_c_R`
- 6.23.3.73 `template<int dim> double UBC_mis_mixing< dim >::Parameters::tau_step`
- 6.23.3.74 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::type_adaptivity_rule`
- 6.23.3.75 `template<int dim> double UBC_mis_mixing< dim >::Parameters::upstream_concentr`

6.23.3.76 `template<int dim> double UBC_mis_mixing< dim >::Parameters::viscosity_ratio`

6.23.3.77 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::which_interpl_c`

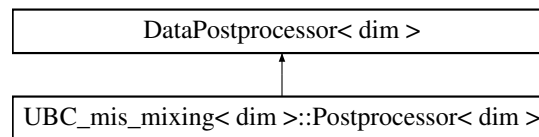
6.23.3.78 `template<int dim> unsigned int UBC_mis_mixing< dim >::Parameters::which_method_for_c`

The documentation for this struct was generated from the following files:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/class.h](#)
- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/parameter.h](#)

## 6.24 UBC\_mis\_mixing< dim >::Postprocessor< dim > Class Template Reference

Inheritance diagram for UBC\_mis\_mixing< dim >::Postprocessor< dim >:



### Public Member Functions

- [Postprocessor](#) (const unsigned int partition)
- virtual void [compute\\_derived\\_quantities\\_vector](#) (const std::vector< Vector< double > > &uh, const std::vector< std::vector< Tensor< 1, dim > > > &duh, const std::vector< std::vector< Tensor< 2, dim > > > &dduh, const std::vector< Point< dim > > &normals, const std::vector< Point< dim > > &evaluation\_points, std::vector< Vector< double > > &computed\_quantities) const
- virtual std::vector< std::string > [get\\_names](#) () const
- virtual std::vector< DataComponentInterpretation::DataComponentInterpretation > [get\\_data\\_component\\_interpretation](#) () const
- virtual UpdateFlags [get\\_needed\\_update\\_flags](#) () const

### 6.24.1 Constructor & Destructor Documentation

6.24.1.1 `template<int dim> template<int dim> UBC_mis_mixing< dim >::Postprocessor< dim >::Postprocessor ( const unsigned int partition )`

### 6.24.2 Member Function Documentation

6.24.2.1 `template<int dim> template<int dim> void UBC_mis_mixing< dim >::Postprocessor< dim >::compute_derived_quantities_vector ( const std::vector< Vector< double > > &uh, const std::vector< std::vector< Tensor< 1, dim > > > &duh, const std::vector< std::vector< Tensor< 2, dim > > > &dduh, const std::vector< Point< dim > > &normals, const std::vector< Point< dim > > &evaluation_points, std::vector< Vector< double > > &computed_quantities ) const [virtual]`

6.24.2.2 `template<int dim> template<int dim> std::vector< DataComponentInterpretation::DataComponentInterpretation > UBC_mis_mixing< dim >::Postprocessor< dim >::get_data_component_interpretation ( ) const [virtual]`

6.24.2.3 `template<int dim> template<int dim> std::vector< std::string > UBC_mis_mixing< dim >::Postprocessor< dim >::get_names ( ) const [virtual]`

6.24.2.4 `template<int dim> template<int dim> UpdateFlags UBC_mis_mixing< dim >::Postprocessor< dim >::get_needed_update_flags ( ) const [virtual]`

The documentation for this class was generated from the following file:

- [/Users/miramus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/post\\_process/post\\_processing.cc](#)

## 6.25 Assembly::Scratch::pressure\_rot\_step< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- [pressure\\_rot\\_step](#) (const FiniteElement< dim > &fe\_pressure, const Mapping< dim > &pressure\_mapping, const Quadrature< dim > &quadrature, const UpdateFlags pressure\_update\_flags, const FiniteElement< dim > &fe\_velocity, const Mapping< dim > &velocity\_mapping, const UpdateFlags velocity\_update\_flags, const FiniteElement< dim > &concentr\_fe, const Mapping< dim > &concentr\_mapping, const UpdateFlags concentr\_update\_flags)
- [pressure\\_rot\\_step](#) (const [pressure\\_rot\\_step](#) &data)

### Public Attributes

- FEValues< dim > [fe\\_pressure\\_values](#)
- FEValues< dim > [fe\\_velocity\\_values](#)
- FEValues< dim > [concentr\\_fe\\_values](#)
- std::vector< double > [phi\\_p](#)
- std::vector< double > [aux\\_sol\\_values](#)
- std::vector< double > [pre\\_sol\\_values](#)
- std::vector< Tensor< 2, dim > > [grad\\_vel\\_sol\\_values](#)
- std::vector< double > [concentr\\_values](#)
- std::vector< SymmetricTensor< 2, dim > > [symm\\_grads\\_vel\\_sol](#)

### 6.25.1 Constructor & Destructor Documentation

6.25.1.1 `template<int dim> Assembly::Scratch::pressure_rot_step< dim >::pressure_rot_step ( const FiniteElement< dim > & fe_pressure, const Mapping< dim > & pressure_mapping, const Quadrature< dim > & quadrature, const UpdateFlags pressure_update_flags, const FiniteElement< dim > & fe_velocity, const Mapping< dim > & velocity_mapping, const UpdateFlags velocity_update_flags, const FiniteElement< dim > & concentr_fe, const Mapping< dim > & concentr_mapping, const UpdateFlags concentr_update_flags )`

6.25.1.2 `template<int dim> Assembly::Scratch::pressure_rot_step< dim >::pressure_rot_step ( const pressure_rot_step< dim > & data )`

### 6.25.2 Member Data Documentation

6.25.2.1 `template<int dim> std::vector<double> Assembly::Scratch::pressure_rot_step< dim >::aux_sol_values`

6.25.2.2 `template<int dim> FEValues<dim> Assembly::Scratch::pressure_rot_step< dim >::concentr_fe_values`

6.25.2.3 `template<int dim> std::vector<double> Assembly::Scratch::pressure_rot_step< dim >::concentr_values`

6.25.2.4 `template<int dim> FEValues<dim> Assembly::Scratch::pressure_rot_step< dim >::fe_pressure_values`

6.25.2.5 `template<int dim> FEValues<dim> Assembly::Scratch::pressure_rot_step< dim >::fe_velocity_values`

6.25.2.6 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::pressure_rot_step< dim >::grad_vel_sol_values`

6.25.2.7 `template<int dim> std::vector<double> Assembly::Scratch::pressure_rot_step< dim >::phi_p`

6.25.2.8 `template<int dim> std::vector<double> Assembly::Scratch::pressure_rot_step< dim >::pre_sol_values`

6.25.2.9 `template<int dim> std::vector<SymmetricTensor<2,dim> > Assembly::Scratch::pressure_rot_step< dim >::symm_grads_vel_sol`

The documentation for this struct was generated from the following file:

- `/Users/miramus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/assembly_copydata.h`

## 6.26 Assembly::CopyData::pressure\_rot\_step< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- `pressure_rot_step` (const FiniteElement< dim > &fe\_pressure)
- `pressure_rot_step` (const `pressure_rot_step` &data)

### Public Attributes

- FullMatrix< double > `local_matrix`
- Vector< double > `local_rhs`
- std::vector< types::global\_dof\_index > `local_dof_indices`

### 6.26.1 Constructor & Destructor Documentation

6.26.1.1 `template<int dim> Assembly::CopyData::pressure_rot_step< dim >::pressure_rot_step ( const FiniteElement< dim > & fe_pressure )`

6.26.1.2 `template<int dim> Assembly::CopyData::pressure_rot_step< dim >::pressure_rot_step ( const pressure_rot_step< dim > & data )`

### 6.26.2 Member Data Documentation

6.26.2.1 `template<int dim> std::vector<types::global_dof_index> Assembly::CopyData::pressure_rot_step< dim >::local_dof_indices`

6.26.2.2 `template<int dim> FullMatrix<double> Assembly::CopyData::pressure_rot_step< dim >::local_matrix`

6.26.2.3 `template<int dim> Vector<double> Assembly::CopyData::pressure_rot_step< dim >::local_rhs`

The documentation for this struct was generated from the following file:

- `/Users/miramus/work/Devs/miscible_mixing_series/miscible_mixing/include/mismix/assembly_copydata.h`

## 6.27 Assembly::CopyData::projection\_step< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- `projection_step` (const FiniteElement< dim > &fe\_auxiliary)
- `projection_step` (const `projection_step` &data)

## Public Attributes

- FullMatrix< double > [local\\_matrix](#)
- Vector< double > [local\\_rhs](#)
- std::vector< types::global\_dof\_index > [local\\_dof\\_indices](#)

## 6.27.1 Constructor &amp; Destructor Documentation

6.27.1.1 `template<int dim> Assembly::CopyData::projection_step< dim >::projection_step ( const FiniteElement< dim > & fe_auxiliary )`

6.27.1.2 `template<int dim> Assembly::CopyData::projection_step< dim >::projection_step ( const projection_step< dim > & data )`

## 6.27.2 Member Data Documentation

6.27.2.1 `template<int dim> std::vector<types::global_dof_index> Assembly::CopyData::projection_step< dim >::local_dof_indices`

6.27.2.2 `template<int dim> FullMatrix<double> Assembly::CopyData::projection_step< dim >::local_matrix`

6.27.2.3 `template<int dim> Vector<double> Assembly::CopyData::projection_step< dim >::local_rhs`

The documentation for this struct was generated from the following file:

- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/[assembly\\_copydata.h](#)

## 6.28 Assembly::Scratch::projection\_step&lt; dim &gt; Struct Template Reference

```
#include <assembly_copydata.h>
```

## Public Member Functions

- [projection\\_step](#) (const FiniteElement< dim > &fe\_auxiliary, const Mapping< dim > &auxiliary\_mapping, const Quadrature< dim > &quadrature, const UpdateFlags auxiliary\_update\_flags, const FiniteElement< dim > &fe\_velocity, const Mapping< dim > &velocity\_mapping, const UpdateFlags velocity\_update\_flags, const FiniteElement< dim > &concentr\_fe, const Mapping< dim > &concentr\_mapping, const UpdateFlags concentr\_update\_flags)
- [projection\\_step](#) (const [projection\\_step](#) &data)

## Public Attributes

- FEValues< dim > [fe\\_auxiliary\\_values](#)
- FEValues< dim > [fe\\_velocity\\_values](#)
- FEValues< dim > [concentr\\_fe\\_values](#)
- std::vector< Tensor< 1, dim > > [grads\\_phi\\_p](#)
- std::vector< double > [phi\\_p](#)
- std::vector< Tensor< 2, dim > > [grad\\_vel\\_n\\_plus\\_1\\_values](#)
- std::vector< double > [concentr\\_values](#)
- std::vector< double > [div\\_vel\\_values](#)

## 6.28.1 Constructor &amp; Destructor Documentation

6.28.1.1 `template<int dim> Assembly::Scratch::projection_step< dim >::projection_step ( const FiniteElement< dim > & fe_auxiliary, const Mapping< dim > & auxiliary_mapping, const Quadrature< dim > & quadrature, const UpdateFlags auxiliary_update_flags, const FiniteElement< dim > & fe_velocity, const Mapping< dim > & velocity_mapping, const UpdateFlags velocity_update_flags, const FiniteElement< dim > & concentr_fe, const Mapping< dim > & concentr_mapping, const UpdateFlags concentr_update_flags )`

6.28.1.2 `template<int dim> Assembly::Scratch::projection_step< dim >::projection_step ( const projection_step< dim > & data )`

## 6.28.2 Member Data Documentation

6.28.2.1 `template<int dim> FEValues<dim> Assembly::Scratch::projection_step< dim >::concentr_fe_values`

6.28.2.2 `template<int dim> std::vector<double> Assembly::Scratch::projection_step< dim >::concentr_values`

6.28.2.3 `template<int dim> std::vector<double> Assembly::Scratch::projection_step< dim >::div_vel_values`

6.28.2.4 `template<int dim> FEValues<dim> Assembly::Scratch::projection_step< dim >::fe_auxiliary_values`

6.28.2.5 `template<int dim> FEValues<dim> Assembly::Scratch::projection_step< dim >::fe_velocity_values`

6.28.2.6 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::projection_step< dim >::grad_vel_n_plus_1_values`

6.28.2.7 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::projection_step< dim >::grads_phi_p`

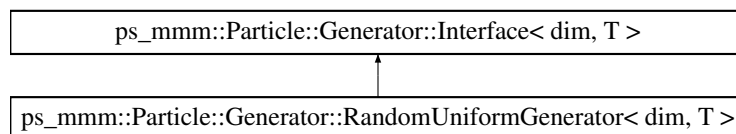
6.28.2.8 `template<int dim> std::vector<double> Assembly::Scratch::projection_step< dim >::phi_p`

The documentation for this struct was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/assembly\\_copydata.h](#)

## 6.29 ps\_mmm::Particle::Generator::RandomUniformGenerator&lt; dim, T &gt; Class Template Reference

Inheritance diagram for ps\_mmm::Particle::Generator::RandomUniformGenerator< dim, T >:



## Public Member Functions

- [RandomUniformGenerator](#) ()  
*Constructor.*
- virtual void [generate\\_particles](#) ([Particle::World](#)< dim, T > &world, const double total\_num\_particles)  
*Generate a uniformly randomly distributed set of particles in the current triangulation.*
- void [uniform\\_random\\_particles\\_in\\_subdomain](#) ([Particle::World](#)< dim, T > &world, const unsigned int num\_particles, const unsigned int start\_id)  
*Generate a set of particles uniformly randomly distributed within the specified triangulation.*

### 6.29.1 Constructor & Destructor Documentation

6.29.1.1 `template<int dim, class T > ps_mmm::Particle::Generator::RandomUniformGenerator< dim, T >::RandomUniformGenerator ( ) [inline]`

Constructor.



## Parameters

in	<i>The</i>	MPI communicator for synchronizing particle generation.
----	------------	---

## 6.29.2 Member Function Documentation

6.29.2.1 `template<int dim, class T> virtual void ps_mmm::Particle::Generator::RandomUniformGenerator< dim, T >::generate_particles ( Particle::World< dim, T > & world, const double total_num_particles ) [inline], [virtual]`

Generate a uniformly randomly distributed set of particles in the current triangulation.

Implements [ps\\_mmm::Particle::Generator::Interface< dim, T >](#).

6.29.2.2 `template<int dim, class T> void ps_mmm::Particle::Generator::RandomUniformGenerator< dim, T >::uniform_random_particles_in_subdomain ( Particle::World< dim, T > & world, const unsigned int num_particles, const unsigned int start_id ) [inline]`

Generate a set of particles uniformly randomly distributed within the specified triangulation.

This is done using "roulette wheel" style selection weighted by cell volume. We do cell-by-cell assignment of particles because the decomposition of the mesh may result in a highly non-rectangular local mesh which makes uniform particle distribution difficult.

## Parameters

in	<i>world</i>	The particle world the particles will exist in
in	<i>num_particles</i>	The number of particles to generate in this subdomain
in	<i>start_id</i>	The starting ID to assign to generated particles

The documentation for this class was generated from the following file:

- [/Users/miramus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/particle/generator.cc](#)

## 6.30 Assembly::CopyData::relaxation\_div\_velocity\_step&lt; dim &gt; Struct Template Reference

```
#include <assembly_copydata.h>
```

## Public Member Functions

- [relaxation\\_div\\_velocity\\_step](#) (const FiniteElement< dim > &fe\_auxiliary)
- [relaxation\\_div\\_velocity\\_step](#) (const [relaxation\\_div\\_velocity\\_step](#) &data)

## Public Attributes

- FullMatrix< double > [local\\_matrix](#)
- Vector< double > [local\\_rhs](#)
- std::vector< types::global\_dof\_index > [local\\_dof\\_indices](#)

## 6.30.1 Constructor &amp; Destructor Documentation

6.30.1.1 `template<int dim> Assembly::CopyData::relaxation_div_velocity_step< dim >::relaxation_div_velocity_step ( const FiniteElement< dim > & fe_auxiliary )`

6.30.1.2 `template<int dim> Assembly::CopyData::relaxation_div_velocity_step< dim >::relaxation_div_velocity_step ( const relaxation_div_velocity_step< dim > & data )`

### 6.30.2 Member Data Documentation

6.30.2.1 `template<int dim> std::vector<types::global_dof_index> Assembly::CopyData::relaxation_div_velocity_step< dim >::local_dof_indices`

6.30.2.2 `template<int dim> FullMatrix<double> Assembly::CopyData::relaxation_div_velocity_step< dim >::local_matrix`

6.30.2.3 `template<int dim> Vector<double> Assembly::CopyData::relaxation_div_velocity_step< dim >::local_rhs`

The documentation for this struct was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/assembly\\_copydata.h](#)

## 6.31 Assembly::Scratch::relaxation\_div\_velocity\_step< dim > Struct Template Reference

```
#include <assembly_copydata.h>
```

### Public Member Functions

- [relaxation\\_div\\_velocity\\_step](#) (const FiniteElement< dim > &fe\_auxiliary, const Mapping< dim > &auxiliary\_mapping, const Quadrature< dim > &quadrature, const UpdateFlags auxiliary\_update\_flags, const FiniteElement< dim > &fe\_velocity, const Mapping< dim > &velocity\_mapping, const UpdateFlags velocity\_update\_flags, const FiniteElement< dim > &concentr\_fe, const Mapping< dim > &concentr\_mapping, const UpdateFlags concentr\_update\_flags)
- [relaxation\\_div\\_velocity\\_step](#) (const [relaxation\\_div\\_velocity\\_step](#) &data)

### Public Attributes

- FEValues< dim > [fe\\_auxiliary\\_values](#)
- FEValues< dim > [fe\\_velocity\\_values](#)
- FEValues< dim > [concentr\\_fe\\_values](#)
- std::vector< Tensor< 1, dim > > [grads\\_phi\\_p](#)
- std::vector< double > [phi\\_p](#)
- std::vector< Tensor< 2, dim > > [grad\\_vel\\_n\\_plus\\_1\\_values](#)
- std::vector< double > [concentr\\_values](#)

### 6.31.1 Constructor & Destructor Documentation

6.31.1.1 `template<int dim> Assembly::Scratch::relaxation_div_velocity_step< dim >::relaxation_div_velocity_step ( const FiniteElement< dim > & fe_auxiliary, const Mapping< dim > & auxiliary_mapping, const Quadrature< dim > & quadrature, const UpdateFlags auxiliary_update_flags, const FiniteElement< dim > & fe_velocity, const Mapping< dim > & velocity_mapping, const UpdateFlags velocity_update_flags, const FiniteElement< dim > & concentr_fe, const Mapping< dim > & concentr_mapping, const UpdateFlags concentr_update_flags )`

6.31.1.2 `template<int dim> Assembly::Scratch::relaxation_div_velocity_step< dim >::relaxation_div_velocity_step ( const relaxation_div_velocity_step< dim > & data )`

### 6.31.2 Member Data Documentation

6.31.2.1 `template<int dim> FEValues<dim> Assembly::Scratch::relaxation_div_velocity_step< dim >::concentr_fe_values`

- 6.31.2.2 `template<int dim> std::vector<double> Assembly::Scratch::relaxation_div_velocity_step< dim >::concentr_values`
- 6.31.2.3 `template<int dim> FEValues<dim> Assembly::Scratch::relaxation_div_velocity_step< dim >::fe_auxiliary_values`
- 6.31.2.4 `template<int dim> FEValues<dim> Assembly::Scratch::relaxation_div_velocity_step< dim >::fe_velocity_values`
- 6.31.2.5 `template<int dim> std::vector<Tensor<2,dim> > Assembly::Scratch::relaxation_div_velocity_step< dim >::grad_vel_n_plus_1_values`
- 6.31.2.6 `template<int dim> std::vector<Tensor<1,dim> > Assembly::Scratch::relaxation_div_velocity_step< dim >::grads_phi_p`
- 6.31.2.7 `template<int dim> std::vector<double> Assembly::Scratch::relaxation_div_velocity_step< dim >::phi_p`

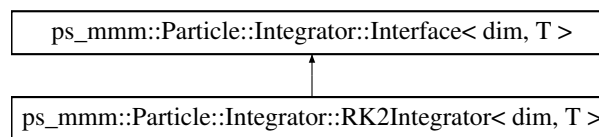
The documentation for this struct was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/include/mismix/assembly\\_copydata.h](#)

## 6.32 ps\_mmm::Particle::Integrator::RK2Integrator< dim, T > Class Template Reference

Runge Kutta second order integrator, where  $y_{n+1} = y_n + dt*v(0.5*k_1)$ ,  $k_1 = dt*v(y_n)$ .

Inheritance diagram for ps\_mmm::Particle::Integrator::RK2Integrator< dim, T >:



### Public Member Functions

- [RK2Integrator](#) (void)
- virtual bool [integrate\\_step](#) ([Particle::World< dim, T > \\*world](#), const double dt)  
*Perform an integration step of moving the particles by the specified timestep dt.*
- virtual void [add\\_mpi\\_types](#) (std::vector< [MPIDataInfo](#) > &data\_info)  
*Secify the MPI types and data sizes involved in transferring integration related information between processes.*
- virtual unsigned int [data\\_len](#) () const  
*Return data length of the integration related data required for communication in terms of number of doubles.*
- virtual unsigned int [read\\_data](#) (const std::vector< double > &data, const unsigned int &pos, const double &id\_num)  
*Read integration related data for a particle specified by id\_num from the data vector.*
- virtual void [write\\_data](#) (std::vector< double > &data, const double &id\_num) const  
*Write integration related data to a vector for a particle specified by id\_num.*

### 6.32.1 Detailed Description

```
template<int dim, class T>class ps_mmm::Particle::Integrator::RK2Integrator< dim, T >
```

Runge Kutta second order integrator, where  $y_{n+1} = y_n + dt*v(0.5*k_1)$ ,  $k_1 = dt*v(y_n)$ .

This scheme requires storing the original location, and the read/write\_data functions reflect this.

### 6.32.2 Constructor & Destructor Documentation

6.32.2.1 `template<int dim, class T> ps_mmm::Particle::Integrator::RK2Integrator< dim, T>::RK2Integrator ( void ) [inline]`

### 6.32.3 Member Function Documentation

6.32.3.1 `template<int dim, class T> virtual void ps_mmm::Particle::Integrator::RK2Integrator< dim, T>::add_mpi_types ( std::vector< MPIDataInfo> & data_info ) [inline], [virtual]`

Specify the MPI types and data sizes involved in transferring integration related information between processes.

If the integrator samples velocities at different locations and the particle moves between processes during the integration step, the sampled velocities must be transferred with the particle.

#### Parameters

<i>in, out</i>	<i>data_info</i>	Adds MPI data info to the specified vector indicating the quantity and type of values the integrator needs saved for this particle.
----------------	------------------	---

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T>](#).

6.32.3.2 `template<int dim, class T> virtual unsigned int ps_mmm::Particle::Integrator::RK2Integrator< dim, T>::data_len ( ) const [inline], [virtual]`

Return data length of the integration related data required for communication in terms of number of doubles.

#### Returns

The number of doubles required to store the relevant integrator data.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T>](#).

6.32.3.3 `template<int dim, class T> virtual bool ps_mmm::Particle::Integrator::RK2Integrator< dim, T>::integrate_step ( Particle::World< dim, T> * world, const double dt ) [inline], [virtual]`

Perform an integration step of moving the particles by the specified timestep dt.

Implementations of this function must update the particle location. If the integrator requires multiple internal steps, this function must return true until all internal steps are finished. Between calls to this function the velocity at the updated particle positions is evaluated and passed to `integrate_step` during the next call.

#### Parameters

<i>in, out</i>	<i>world</i>	The world to integrate particles in. The particle positions will be changed in this function based on the integration scheme.
<i>in</i>	<i>dt</i>	The timestep length to perform the integration.

#### Returns

Whether this function needs to be called again (true) for additional integration steps or if all internal steps are complete (false).

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T>](#).

6.32.3.4 `template<int dim, class T> virtual unsigned int ps_mmm::Particle::Integrator::RK2Integrator< dim, T>::read_data ( const std::vector< double> & data, const unsigned int & pos, const double & id_num ) [inline], [virtual]`

Read integration related data for a particle specified by `id_num` from the data vector.

## Parameters

in	<i>data</i>	The vector of double data to read from.
in	<i>pos</i>	The position in the data vector to start reading from.
in	<i>id_num</i>	The id number of the particle to read the data for.

## Returns

The position in the vector of the next unread double.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

6.32.3.5 `template<int dim, class T> virtual void ps_mmm::Particle::Integrator::RK2Integrator< dim, T >::write_data ( std::vector< double > & data, const double & id_num ) const [inline], [virtual]`

Write integration related data to a vector for a particle specified by id\_num.

## Parameters

in, out	<i>data</i>	The vector of doubles to write integrator data into.
in	<i>id_num</i>	The id number of the particle to read the data for.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

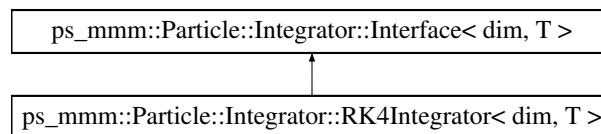
The documentation for this class was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/particle/integrator.cc](#)

## 6.33 ps\_mmm::Particle::Integrator::RK4Integrator&lt; dim, T &gt; Class Template Reference

Runge Kutta fourth order integrator, where  $y_{n+1} = y_n + (1/6)*k_1 + (1/3)*k_2 + (1/3)*k_3 + (1/6)*k_4$  and  $k_1, k_2, k_3, k_4$  are defined as usual.

Inheritance diagram for `ps_mmm::Particle::Integrator::RK4Integrator< dim, T >`:



## Public Member Functions

- [RK4Integrator](#) (void)
- virtual bool [integrate\\_step](#) ([Particle::World](#)< dim, T > \*world, const double dt)  
*Perform an integration step of moving the particles by the specified timestep dt.*
- virtual void [add\\_mpi\\_types](#) (std::vector< [MPIDataInfo](#) > &data\_info)  
*Secify the MPI types and data sizes involved in transferring integration related information between processes.*
- virtual unsigned int [data\\_len](#) () const  
*Return data length of the integration related data required for communication in terms of number of doubles.*
- virtual unsigned int [read\\_data](#) (const std::vector< double > &data, const unsigned int &pos, const double &id\_num)  
*Read integration related data for a particle specified by id\_num from the data vector.*
- virtual void [write\\_data](#) (std::vector< double > &data, const double &id\_num) const  
*Write integration related data to a vector for a particle specified by id\_num.*

### 6.33.1 Detailed Description

```
template<int dim, class T>class ps_mmm::Particle::Integrator::RK4Integrator< dim, T >
```

Runge Kutta fourth order integrator, where  $y_{n+1} = y_n + (1/6)*k_1 + (1/3)*k_2 + (1/3)*k_3 + (1/6)*k_4$  and  $k_1, k_2, k_3, k_4$  are defined as usual.

This scheme requires storing the original location and intermediate  $k_1, k_2, k_3$  values, so the read/write\_data functions reflect this.

### 6.33.2 Constructor & Destructor Documentation

```
6.33.2.1 template<int dim, class T > ps_mmm::Particle::Integrator::RK4Integrator< dim, T >::RK4Integrator (
        void ) [inline]
```

### 6.33.3 Member Function Documentation

```
6.33.3.1 template<int dim, class T > virtual void ps_mmm::Particle::Integrator::RK4Integrator< dim, T
        >::add_mpi_types ( std::vector< MPIDataInfo > & data_info ) [inline],[virtual]
```

Specify the MPI types and data sizes involved in transferring integration related information between processes.

If the integrator samples velocities at different locations and the particle moves between processes during the integration step, the sampled velocities must be transferred with the particle.

#### Parameters

<i>in, out</i>	<i>data_info</i>	Adds MPI data info to the specified vector indicating the quantity and type of values the integrator needs saved for this particle.
----------------	------------------	---

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

```
6.33.3.2 template<int dim, class T > virtual unsigned int ps_mmm::Particle::Integrator::RK4Integrator< dim, T
        >::data_len ( ) const [inline],[virtual]
```

Return data length of the integration related data required for communication in terms of number of doubles.

#### Returns

The number of doubles required to store the relevant integrator data.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

```
6.33.3.3 template<int dim, class T > virtual bool ps_mmm::Particle::Integrator::RK4Integrator< dim, T
        >::integrate_step ( Particle::World< dim, T > * world, const double dt ) [inline],[virtual]
```

Perform an integration step of moving the particles by the specified timestep  $dt$ .

Implementations of this function must update the particle location. If the integrator requires multiple internal steps, this function must return true until all internal steps are finished. Between calls to this function the velocity at the updated particle positions is evaluated and passed to `integrate_step` during the next call.

#### Parameters

<i>in, out</i>	<i>world</i>	The world to integrate particles in. The particle positions will be changed in this function based on the integration scheme.
<i>in</i>	<i>dt</i>	The timestep length to perform the integration.

#### Returns

Whether this function needs to be called again (true) for additional integration steps or if all internal steps are complete (false).

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

6.33.3.4 `template<int dim, class T > virtual unsigned int ps_mmm::Particle::Integrator::RK4Integrator< dim, T >::read_data ( const std::vector< double > & data, const unsigned int & pos, const double & id_num ) [inline],[virtual]`

Read integration related data for a particle specified by id\_num from the data vector.

#### Parameters

in	<i>data</i>	The vector of double data to read from.
in	<i>pos</i>	The position in the data vector to start reading from.
in	<i>id_num</i>	The id number of the particle to read the data for.

#### Returns

The position in the vector of the next unread double.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

6.33.3.5 `template<int dim, class T > virtual void ps_mmm::Particle::Integrator::RK4Integrator< dim, T >::write_data ( std::vector< double > & data, const double & id_num ) const [inline],[virtual]`

Write integration related data to a vector for a particle specified by id\_num.

#### Parameters

in, out	<i>data</i>	The vector of doubles to write integrator data into.
in	<i>id_num</i>	The id number of the particle to read the data for.

Implements [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#).

The documentation for this class was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/particle/integrator.cc](#)

## 6.34 UBC\_mis\_mixing< dim > Class Template Reference

```
#include <class.h>
```

#### Classes

- struct [Parameters](#)
- class [Postprocessor](#)

#### Public Member Functions

- [UBC\\_mis\\_mixing](#) ([Parameters](#) &parameters)
- void [run](#) ()

#### 6.34.1 Constructor & Destructor Documentation

6.34.1.1 `template<int dim> UBC_mis_mixing< dim >::UBC_mis_mixing ( Parameters & parameters )`

#### 6.34.2 Member Function Documentation

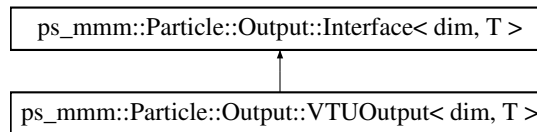
6.34.2.1 `template<int dim> void UBC_mis_mixing< dim >::run ( )`

The documentation for this class was generated from the following files:

- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/class.h
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/amr/amr.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/constitutive\_model/consti\_model.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/main/run.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/post\_process/extract\_data.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/post\_process/post\_processing.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/constructor.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/mesh\_in.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/read\_and\_write.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/setup\_dofs.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/navier\_stokes/control.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/navier\_stokes/projection\_for\_div\_velocity.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/navier\_stokes/solve\_ns\_equation.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/phase/solve\_hyperbolic\_equation.cc
- /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/support/utilities.cc

### 6.35 ps\_mmm::Particle::Output::VTUOutput< dim, T > Class Template Reference

Inheritance diagram for ps\_mmm::Particle::Output::VTUOutput< dim, T >:



#### Public Member Functions

- **VTUOutput** (const std::string &output\_directory, const MPI\_Comm communicator)  
*Constructor.*
- virtual std::string **output\_particle\_data** (const std::multimap< LevelInd, T > &particles, const double &current\_time)  
*Write data about the particles specified in the first argument to a file.*

#### Additional Inherited Members

##### 6.35.1 Constructor & Destructor Documentation

6.35.1.1 `template<int dim, class T > ps_mmm::Particle::Output::VTUOutput< dim, T >::VTUOutput ( const std::string & output_directory, const MPI_Comm communicator ) [inline]`

Constructor.

Parameters

---



in	<i>The</i>	directory into which output files shall be placed.
in	<i>The</i>	MPI communicator that describes this simulation.

### 6.35.2 Member Function Documentation

6.35.2.1 `template<int dim, class T > virtual std::string ps_mmm::Particle::Output::VTUOutput< dim, T >::output_particle_data ( const std::multimap< LevelInd, T > & particles, const double & current_time ) [inline],[virtual]`

Write data about the particles specified in the first argument to a file.

If possible, encode the current simulation time into this file using the data provided in the second argument.

#### Parameters

in	<i>particles</i>	The set of particles to generate a graphical representation for
in	<i>current_time</i>	Current time of the simulation, given as either years or seconds, as selected in the input file. In other words, output writers do not need to know the units in which time is described.

#### Returns

The name of the file that was written, or any other information that describes what output was produced if for example multiple files were created.

Implements [ps\\_mmm::Particle::Output::Interface< dim, T >](#).

The documentation for this class was generated from the following file:

- [/Users/miranus/work/Devs/miscible\\_mixing\\_series/miscible\\_mixing/source/particle/output.cc](#)

## 6.36 ps\_mmm::Particle::World< dim, T > Class Template Reference

```
#include <world.h>
```

### Public Member Functions

- [World \(\)](#)  
*Default [World](#) constructor.*
- [~World \(\)](#)  
*Default [World](#) destructor, deallocates all relevant arrays and structures.*
- `const Mapping< dim > * get\_mapping () const`  
*Get the deal.II Mapping associated with this particle world.*
- `void set\_mapping (const Mapping< dim > *new_mapping)`  
*Set the deal.II Mapping associated with this particle world.*
- `void set\_triangulation (const parallel::distributed::Triangulation< dim > *new_tria)`  
*Set the deal.II Triangulation associated with this particle world and connects relevant listener for mesh changes.*
- `const parallel::distributed::Triangulation< dim > * get\_triangulation ()`  
*Get the deal.II Triangulation associated with this particle world.*
- `const DoFHandler< dim > * get\_dof\_handler () const`  
*Get the deal.II DoFHandler associated with this particle world.*
- `void set\_dof\_handler (const DoFHandler< dim > *new_dh)`  
*Set the deal.II DoFHandler associated with this particle world.*
- `const TrilinosWrappers::MPI::Vector * get\_solution () const`

- Get the deal.II BlockVector solution associated with this particle world.*

  - void [set\\_solution](#) (const TrilinosWrappers::MPI::Vector \*new\_solution)

*Set the deal.II BlockVector solution associated with this particle world.*

- void [set\\_integrator](#) ([Integrator::Interface](#)< dim, T > \*new\_integrator)

*Set the particle [Integrator](#) scheme for this particle world.*

- void [set\\_mpi\\_comm](#) (const MPI\_Comm new\_comm\_world)

*Set the MPI communicator for this world.*

- MPI\_Comm [mpi\\_comm](#) ()

*Get the MPI communicator associated with this particle world.*

- void [finished\\_adding\\_particles](#) ()

*All processes must call this function when finished adding particles to the world.*

- void [add\\_particle](#) (const T &particle, const [LevelInd](#) &cell)

*Add a particle to this world.*

- std::multimap< [LevelInd](#), T > & [get\\_particles](#) ()

*Access to particles in this world.*

- const std::multimap< [LevelInd](#), T > & [get\\_particles](#) () const

*Const access to particles in this world.*

- void [init](#) ()

*Initialize the particle world by creating appropriate MPI data types for transferring particles, and allocating memory for MPI related functions.*

- void [find\\_all\\_cells](#) ()

*Calculate the cells containing each particle for all particles.*

- void [advance\\_timestep](#) (const double timestep, const TrilinosWrappers::MPI::Vector &solution)

*Advance particles by the specified timestep using the current integration scheme.*

- void [move\\_particles\\_back\\_in\\_mesh](#) ()
- void [mark\\_particles\\_for\\_check](#) ()

*Mark all particles to be checked for velocity at their current position.*

- [LevelInd](#) [find\\_cell](#) (T &particle, const [LevelInd](#) &cur\_cell)

*Finds the cell the particle is contained in and returns the appropriate cell level/index.*

- void [send\\_rcv\\_particles](#) ()

*Transfer particles that have crossed subdomain boundaries to other processors.*

- void [get\\_particle\\_velocities](#) (const TrilinosWrappers::MPI::Vector &solution)

*Calculates the velocities for each particle at its location given the input solution velocity field.*

- unsigned int [get\\_global\\_particle\\_count](#) ()

*Calculates the global sum of particles over all processes.*

- void [check\\_particle\\_count](#) ()

*Checks that the number of particles in the simulation has not unexpectedly changed.*

- template<class Archive >
- void [serialize](#) (Archive &ar, const unsigned int version)

*Read or write the data of this object for serialization.*

### 6.36.1 Constructor & Destructor Documentation

6.36.1.1 `template<int dim, class T> ps_mmm::Particle::World< dim, T >::World ( ) [inline]`

Default [World](#) constructor.

6.36.1.2 `template<int dim, class T> ps_mmm::Particle::World< dim, T >::~~World ( ) [inline]`

Default [World](#) destructor, deallocates all relevant arrays and structures.

## 6.36.2 Member Function Documentation

6.36.2.1 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::add_particle ( const T & particle, const LevelInd & cell ) [inline]`

Add a particle to this world.

If the specified cell does not exist in the local subdomain an exception will be thrown.

6.36.2.2 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::advance_timestep ( const double timestep, const TrilinosWrappers::MPI::Vector & solution ) [inline]`

Advance particles by the specified timestep using the current integration scheme.

## Parameters

in	<i>timestep</i>	Length of timestep to integrate particle movement
in	<i>solution</i>	Current Aspect solution vector

6.36.2.3 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::check_particle_count ( ) [inline]`

Checks that the number of particles in the simulation has not unexpectedly changed.

If the particle count changes then the simulation will be aborted.

6.36.2.4 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::find_all_cells ( ) [inline]`

Calculate the cells containing each particle for all particles.

6.36.2.5 `template<int dim, class T> LevelInd ps_mmm::Particle::World< dim, T >::find_cell ( T & particle, const LevelInd & cur_cell ) [inline]`

Finds the cell the particle is contained in and returns the appropriate cell level/index.

## Parameters

in, out	<i>particle</i>	The particle to find the cell for. This particle will be updated to indicate whether it is in the local subdomain or not.
in	<i>cur_cell</i>	The current cell (level and index) being checked.

## Returns

The level and index of the active cell the particle is in. If no cell was found to contain the particle, return the level/index (-1, -1)

6.36.2.6 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::finished_adding_particles ( ) [inline]`

All processes must call this function when finished adding particles to the world.

This function will determine the total number of particles.

6.36.2.7 `template<int dim, class T> const DoFHandler<dim>* ps_mmm::Particle::World< dim, T >::get_dof_handler ( ) const [inline]`

Get the deal.II DoFHandler associated with this particle world.

## Returns

The DoFHandler for this world.

**6.36.2.8** `template<int dim, class T> unsigned int ps_mmm::Particle::World< dim, T >::get_global_particle_count ( )`  
`[inline]`

Calculates the global sum of particles over all processes.

This is done to ensure no particles have fallen out of the simulation domain.

**Returns**

Total number of particles in simulation.

**6.36.2.9** `template<int dim, class T> const Mapping<dim>* ps_mmm::Particle::World< dim, T >::get_mapping ( )`  
`const [inline]`

Get the deal.II Mapping associated with this particle world.

**Returns**

The Mapping for this world.

**6.36.2.10** `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::get_particle_velocities ( const`  
`TrilinosWrappers::MPI::Vector & solution ) [inline]`

Calculates the velocities for each particle at its location given the input solution velocity field.

The calculated velocities are stored in the [Particle](#) objects for this world.

**Parameters**

<code>in</code>	<code>solution</code>	The current solution vector for this simulation.
-----------------	-----------------------	--

**6.36.2.11** `template<int dim, class T> std::multimap<LevelInd, T>& ps_mmm::Particle::World< dim, T`  
`>::get_particles ( ) [inline]`

Access to particles in this world.

**6.36.2.12** `template<int dim, class T> const std::multimap<LevelInd, T>& ps_mmm::Particle::World< dim, T`  
`>::get_particles ( ) const [inline]`

Const access to particles in this world.

**6.36.2.13** `template<int dim, class T> const TrilinosWrappers::MPI::Vector* ps_mmm::Particle::World< dim, T`  
`>::get_solution ( ) const [inline]`

Get the deal.II BlockVector solution associated with this particle world.

**Returns**

The BlockVector solution for this world.

**6.36.2.14** `template<int dim, class T> const parallel::distributed::Triangulation<dim>* ps_mmm::Particle::World< dim,`  
`T >::get_triangulation ( ) [inline]`

Get the deal.II Triangulation associated with this particle world.

**Returns**

const pointer to associated Triangulation

**6.36.2.15** `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::init ( ) [inline]`

Initialize the particle world by creating appropriate MPI data types for transferring particles, and allocating memory for MPI related functions.

6.36.2.16 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::mark_particles_for_check ( )`  
`[inline]`

Mark all particles to be checked for velocity at their current position.

6.36.2.17 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::move_particles_back_in_mesh ( )`  
`[inline]`

6.36.2.18 `template<int dim, class T> MPI_Comm ps_mmm::Particle::World< dim, T >::mpi_comm ( )` `[inline]`

Get the MPI communicator associated with this particle world.

#### Returns

associated MPI\_Comm

6.36.2.19 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::send_rcv_particles ( )` `[inline]`

Transfer particles that have crossed subdomain boundaries to other processors.

Because subdomains can change drastically during mesh refinement, particle transfer occurs as follows: - Each subdomain finds the particles it owns which have fallen outside it - For each particle outside the subdomain, send the particle to all subdomains and let them determine which one owns it. This assumes there is no overlap between subdomains. - Each process determines which of the received particles is in its subdomain, keeps these and deletes the others - TODO: handle particles outside any domain - TODO: if we know the domain of a particle (e.g. bordering domains), send it only to that domain

6.36.2.20 `template<int dim, class T> template<class Archive > void ps_mmm::Particle::World< dim, T >::serialize ( Archive & ar, const unsigned int version )` `[inline]`

Read or write the data of this object for serialization.

6.36.2.21 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::set_dof_handler ( const DoFHandler< dim > * new_dh )` `[inline]`

Set the deal.II DoFHandler associated with this particle world.

#### Parameters

<code>in</code>	<code>new_dh</code>	The new DoFHandler for this world.
-----------------	---------------------	------------------------------------

6.36.2.22 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::set_integrator ( Integrator::Interface< dim, T > * new_integrator )` `[inline]`

Set the particle [Integrator](#) scheme for this particle world.

#### Parameters

<code>in</code>	<code>new_integrator</code>	The new <a href="#">Integrator</a> scheme for this world.
-----------------	-----------------------------	---

6.36.2.23 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::set_mapping ( const Mapping< dim > * new_mapping )` `[inline]`

Set the deal.II Mapping associated with this particle world.

#### Parameters

<code>in</code>	<code>new_mapping</code>	The new Mapping for this world.
-----------------	--------------------------	---------------------------------

6.36.2.24 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::set_mpi_comm ( const MPI_Comm  
new_comm_world ) [inline]`

Set the MPI communicator for this world.

## Parameters

in	<i>new_comm</i> ↔ <i>world</i>	The new MPI_Comm object for this world.
----	-----------------------------------	---

6.36.2.25 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::set_solution ( const TrilinosWrappers::MPI::Vector * new_solution ) [inline]`

Set the deal.II BlockVector solution associated with this particle world.

## Parameters

in	<i>new_solution</i>	The new TrilinosWrappers::MPI::Vector solution for this world.
----	---------------------	--

6.36.2.26 `template<int dim, class T> void ps_mmm::Particle::World< dim, T >::set_triangulation ( const parallel::distributed::Triangulation< dim > * new_tria ) [inline]`

Set the deal.II Triangulation associated with this particle world and connects relevant listener for mesh changes.

## Parameters

in	<i>new_tria</i>	The new Triangulation for this world.
----	-----------------	---------------------------------------

The documentation for this class was generated from the following file:

- /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/[world.h](#)

## 7 File Documentation

### 7.1 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/assembly\_↔ copydata.h File Reference

```
#include "include.h"
```

## Classes

- struct [Assembly::Scratch::diffusion\\_step< dim >](#)
- struct [Assembly::Scratch::relaxation\\_div\\_velocity\\_step< dim >](#)
- struct [Assembly::Scratch::projection\\_step< dim >](#)
- struct [Assembly::Scratch::pressure\\_rot\\_step< dim >](#)
- struct [Assembly::Scratch::concentrMatrix< dim >](#)
- struct [Assembly::Scratch::concentrRHS< dim >](#)
- struct [Assembly::CopyData::diffusion\\_step< dim >](#)
- struct [Assembly::CopyData::relaxation\\_div\\_velocity\\_step< dim >](#)
- struct [Assembly::CopyData::projection\\_step< dim >](#)
- struct [Assembly::CopyData::pressure\\_rot\\_step< dim >](#)
- struct [Assembly::CopyData::concentrMatrix< dim >](#)
- struct [Assembly::CopyData::concentrRHS< dim >](#)

## Namespaces

- [Assembly](#)
- [Assembly::Scratch](#)
- [Assembly::CopyData](#)

## 7.2 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/class.h File Reference

```
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include <limits>
#include <locale>
#include "include.h"
#include "equation_data.h"
```

### Classes

- class [UBC\\_mis\\_mixing< dim >](#)
- struct [UBC\\_mis\\_mixing< dim >::Parameters](#)

## 7.3 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/equation\_data.h File Reference

```
#include <deal.II/base/function.h>
```

### Classes

- class [EquationData::Inflow\\_Velocity< dim >](#)
- class [EquationData::Outflow\\_Pressure< dim >](#)
- class [EquationData::concentrInletValues< dim >](#)
- class [EquationData::concentrInitialValues< dim >](#)

### Namespaces

- [EquationData](#)

### Variables

- const double [EquationData::pipe\\_diameter](#) = 19.05
- const double [EquationData::gravity\\_accelation](#) = 9800
- const double [EquationData::upstream\\_concentr](#) = 0.0
- const double [EquationData::downstream\\_concentr](#) = 1.0
- const double [EquationData::kinematic\\_viscosity](#) = 1.0

## 7.4 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/include.h File Reference

```
#include <deal.II/base/quadrature_lib.h>
```



```
#include <deal.II/base/logstream.h>
#include <deal.II/base/function.h>
#include <deal.II/base/utilities.h>
#include <deal.II/base/conditional_ostream.h>
#include <deal.II/base/work_stream.h>
#include <deal.II/base/timer.h>
#include <deal.II/base/parameter_handler.h>
#include <deal.II/lac/full_matrix.h>
#include <deal.II/lac/solver_gmres.h>
#include <deal.II/lac/solver_bicgstab.h>
#include <deal.II/lac/solver_cg.h>
#include <deal.II/lac/constraint_matrix.h>
#include <deal.II/lac/block_sparsity_pattern.h>
#include <deal.II/lac/trilinos_block_vector.h>
#include <deal.II/lac/trilinos_sparse_matrix.h>
#include <deal.II/lac/trilinos_block_sparse_matrix.h>
#include <deal.II/lac/trilinos_precondition.h>
#include <deal.II/lac/trilinos_vector_base.h>
#include <deal.II/lac/trilinos_vector.h>
#include <deal.II/lac/trilinos_solver.h>
#include <deal.II/grid/tria.h>
#include <deal.II/grid/grid_in.h>
#include <deal.II/grid/grid_out.h>
#include <deal.II/grid/grid_generator.h>
#include <deal.II/grid/tria_accessor.h>
#include <deal.II/grid/tria_iterator.h>
#include <deal.II/grid/tria_boundary_lib.h>
#include <deal.II/grid/grid_tools.h>
#include <deal.II/grid/grid_refinement.h>
#include <deal.II/grid/filtered_iterator.h>
#include <deal.II/dofs/dof_handler.h>
#include <deal.II/dofs/dof_renumbering.h>
#include <deal.II/dofs/dof_accessor.h>
#include <deal.II/dofs/dof_tools.h>
#include <deal.II/fe/fe_q.h>
#include <deal.II/fe/fe_dgq.h>
#include <deal.II/fe/fe_dgp.h>
#include <deal.II/fe/fe_system.h>
#include <deal.II/fe/fe_values.h>
#include <deal.II/fe/mapping_q.h>
#include <deal.II/numerics/vector_tools.h>
#include <deal.II/numerics/matrix_tools.h>
#include <deal.II/numerics/data_out.h>
#include <deal.II/numerics/error_estimator.h>
#include <deal.II/numerics/solution_transfer.h>
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include <limits>
#include <locale>
#include <deal.II/distributed/solution_transfer.h>
#include <deal.II/base/index_set.h>
#include <deal.II/distributed/tria.h>
#include <deal.II/distributed/grid_refinement.h>
```

## 7.5 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/parameter.h File Reference

```
#include <fstream>
#include <iostream>
#include <sstream>
#include <string>
#include <limits>
#include <locale>
#include "include.h"
#include "equation_data.h"
#include "class.h"
```

## 7.6 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/generator.h File Reference

```
#include <mismix/particle/particle.h>
#include <mismix/particle/world.h>
```

### Classes

- class [ps\\_mmm::Particle::Generator::Interface< dim, T >](#)  
*Abstract base class used for classes that generate particles.*

### Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::Particle](#)
- [ps\\_mmm::Particle::Generator](#)

### Functions

- `template<int dim, class T >`  
`Interface< dim, T > * ps\_mmm::Particle::Generator::create\_generator\_object (const std::string &generator↵  
_type)`  
*Create a generator object.*
- `std::string ps\_mmm::Particle::Generator::generator\_object\_names ()`  
*Return a list of names (separated by '|') of possible particle generators.*

## 7.7 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/integrator.h File Reference

```
#include <mismix/particle/world.h>
#include <deal.II/numerics/fe_field_function.h>
```

### Classes

- class [ps\\_mmm::Particle::Integrator::Interface< dim, T >](#)  
*An abstract class defining virtual methods for performing integration of particle paths through the simulation velocity field.*

## Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::Particle](#)
- [ps\\_mmm::Particle::Integrator](#)

## Functions

- `template<int dim, class T >`  
`Interface< dim, T > * ps\_mmm::Particle::Integrator::create\_integrator\_object (const std::string &integrator_  
_name)`  
*Create an integrator object.*
- `std::string ps\_mmm::Particle::Integrator::integrator\_object\_names ()`  
*Return a list of names (separated by '\') of possible integrator classes for particles.*

## 7.8 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/output.h File Reference

```
#include <deal.II/base/mpi.h>  
#include <mismix/particle/particle.h>
```

## Classes

- class [ps\\_mmm::Particle::Output::Interface< dim, T >](#)  
*Abstract base class used for classes that generate particle output.*

## Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::Particle](#)
- [ps\\_mmm::Particle::Output](#)

## Functions

- `template<int dim, class T >`  
`Interface< dim, T > * ps\_mmm::Particle::Output::create\_output\_object (const std::string &data_format_  
name, const std::string &output_directory, const MPI_Comm communicator)`  
*Create an output object.*
- `std::string ps\_mmm::Particle::Output::output\_object\_names ()`  
*Return a list of names (separated by '\') of possible writers of graphical output formats for particle data.*

## 7.9 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/particle.h File Reference

```
#include <mismix/include.h>
```

## Classes

- class `ps_mmm::Particle::MPIDataInfo`
- class `ps_mmm::Particle::BaseParticle< dim >`  
*Base class of particles - represents a particle with position, velocity, and an ID number.*
- class `ps_mmm::Particle::DataParticle< dim, data_dim >`  
*DataParticle provides an example of how to extend the BaseParticle class to include related particle data.*

## Namespaces

- `ps_mmm`
- `ps_mmm::Particle`

## Typedefs

- typedef `std::pair< int, int >` `ps_mmm::Particle::LevelInd`  
*Typedef of cell level/index pair.*

## 7.10 /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/particle/world.h

### File Reference

```
#include <deal.II/numerics/fe_field_function.h>
#include <mismix/particle/particle.h>
#include <mismix/include.h>
```

## Classes

- class `ps_mmm::Particle::Integrator::Interface< dim, T >`  
*An abstract class defining virtual methods for performing integration of particle paths through the simulation velocity field.*
- class `ps_mmm::Particle::World< dim, T >`

## Namespaces

- `ps_mmm`
- `ps_mmm::Particle`
- `ps_mmm::Particle::Integrator`

## Variables

- const int `ps_mmm::Particle::PARTICLE_XFER_TAG` = 382  
*MPI tag for particle transfers.*

## 7.11 /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/include/mismix/phase/levelset.h

### File Reference

```
#include <mismix/include.h>
```

## Classes

- class [ps\\_mmm::immicable::Levelset< dim >](#)

## Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::immicable](#)

## 7.12 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/amr/amr.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

## 7.13 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/constitutive\_model/consti↵\_model.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

## 7.14 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/main/main.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

## Functions

- int [main](#) (int argc, char \*argv[ ])

### 7.14.1 Function Documentation

#### 7.14.1.1 int main ( int argc, char \* argv[ ] )

## 7.15 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/main/run.cc File Reference

```
#include <mismix/include.h>
```

```
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

## 7.16 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/particle/generator.cc File Reference

```
#include <mismix/particle/generator.h>
#include <boost/random.hpp>
```

### Classes

- class [ps\\_mmm::Particle::Generator::RandomUniformGenerator](#)< dim, T >

### Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::Particle](#)
- [ps\\_mmm::Particle::Generator](#)

### Functions

- template<int dim, class T >  
Interface< dim, T > \* [ps\\_mmm::Particle::Generator::create\\_generator\\_object](#) (const std::string &generator↔\_type)  
*Create a generator object.*
- std::string [ps\\_mmm::Particle::Generator::generator\\_object\\_names](#) ()  
*Return a list of names (separated by '\n') of possible particle generators.*

## 7.17 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/particle/integrator.cc File Reference

```
#include <mismix/particle/integrator.h>
```

### Classes

- class [ps\\_mmm::Particle::Integrator::EulerIntegrator](#)< dim, T >  
*Euler scheme integrator, where  $y_{n+1} = y_n + dt * v(y_n)$ .*
- class [ps\\_mmm::Particle::Integrator::RK2Integrator](#)< dim, T >  
*Runge Kutta second order integrator, where  $y_{n+1} = y_n + dt * v(0.5 * k_1)$ ,  $k_1 = dt * v(y_n)$ .*
- class [ps\\_mmm::Particle::Integrator::RK4Integrator](#)< dim, T >  
*Runge Kutta fourth order integrator, where  $y_{n+1} = y_n + (1/6) * k_1 + (1/3) * k_2 + (1/3) * k_3 + (1/6) * k_4$  and  $k_1, k_2, k_3, k_4$  are defined as usual.*
- class [ps\\_mmm::Particle::Integrator::HybridIntegrator](#)< dim, T >  
*[Integrator](#) which chooses Euler, RK2 or RK4 depending on characteristics of the cell a particle is in.*

## Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::Particle](#)
- [ps\\_mmm::Particle::Integrator](#)

## Functions

- `template<int dim, class T >`  
`Interface< dim, T > * ps\_mmm::Particle::Integrator::create\_integrator\_object (const std::string &integrator_↵  
_name)  
Create an integrator object.`
- `std::string ps\_mmm::Particle::Integrator::integrator\_object\_names ()`  
*Return a list of names (separated by '|') of possible integrator classes for particles.*

## 7.18 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/particle/output.cc File Reference

```
#include <mismix/particle/output.h>
#include <mismix/particle/particle.h>
#include <deal.II/numerics/data_out.h>
```

## Classes

- class [ps\\_mmm::Particle::Output::NullOutput< dim, T >](#)
- class [ps\\_mmm::Particle::Output::ASCIIOutput< dim, T >](#)
- class [ps\\_mmm::Particle::Output::VTUOutput< dim, T >](#)
- class [ps\\_mmm::Particle::Output::HDF5Output< dim, T >](#)

## Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::Particle](#)
- [ps\\_mmm::Particle::Output](#)

## Functions

- `template<int dim, class T >`  
`Interface< dim, T > * ps\_mmm::Particle::Output::create\_output\_object (const std::string &data_format_↵  
name, const std::string &output_directory, const MPI_Comm communicator)  
Create an output object.`
- `std::string ps\_mmm::Particle::Output::output\_object\_names ()`  
*Return a list of names (separated by '|') of possible writers of graphical output formats for particle data.*

## 7.19 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/particle/particle.cc File Reference

```
#include <mismix/particle/particle.h>
```

## Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::Particle](#)

### 7.20 /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/post\_process/extract\_↵ data.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

### 7.21 /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/post\_process/post\_↵ processing.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

## Classes

- class [UBC\\_mis\\_mixing< dim >::Postprocessor< dim >](#)

### 7.22 /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/constructor.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

### 7.23 /Users/miramus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/mesh\_in.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```



## 7.24 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/read\_and\_write.cc File Reference

75

### 7.24 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/read\_and\_write.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
#include <boost/iostreams/tee.hpp>
#include <boost/iostreams/stream.hpp>
#include <boost/archive/binary_oarchive.hpp>
#include <boost/archive/binary_iarchive.hpp>
```

### 7.25 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/pre\_process/setup\_dofs.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
#include <deal.II/base/types.h>
```

### 7.26 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/navier\_stokes/control.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

### 7.27 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/navier\_stokes/projection\_for\_div\_velocity.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

### 7.28 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/navier\_stokes/solve\_ns\_equation.cc File Reference

```
#include <mismix/include.h>
#include <mismix/equation_data.h>
#include <mismix/assembly_copydata.h>
#include <mismix/class.h>
#include <mismix/parameter.h>
```

### 7.29 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/phase/levelset.cc File Reference

```
#include <mismix/phase/levelset.h>
```

#### Namespaces

- [ps\\_mmm](#)
- [ps\\_mmm::immicable](#)

### 7.30 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/solver/phase/solve\_↔ hyperbolic\_equation.cc File Reference

```
#include <mismix/include.h>  
#include <mismix/equation_data.h>  
#include <mismix/assembly_copydata.h>  
#include <mismix/class.h>  
#include <mismix/parameter.h>
```

### 7.31 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible\_mixing/source/support/utilities.cc File Reference

```
#include <mismix/include.h>  
#include <mismix/equation_data.h>  
#include <mismix/assembly_copydata.h>  
#include <mismix/class.h>  
#include <mismix/parameter.h>
```

## Index

/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ write.cc, 75  
\_mixing/include/mismix/assembly\_copydata.↔ /Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
h, 65 \_mixing/source/pre\_process/setup\_dofs.cc,  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ 75  
\_mixing/include/mismix/class.h, 66 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ \_mixing/source/solver/navier\_stokes/control.↔  
\_mixing/include/mismix/equation\_data.h, 66 cc, 75  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ /Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
\_mixing/include/mismix/include.h, 66 \_mixing/source/solver/navier\_stokes/projection↔  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ \_for\_div\_velocity.cc, 75  
\_mixing/include/mismix/parameter.h, 68 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ \_mixing/source/solver/navier\_stokes/solve\_↔  
\_mixing/include/mismix/particle/generator.h, ns\_equation.cc, 75  
68 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ \_mixing/source/solver/phase/levelset.cc, 76  
\_mixing/include/mismix/particle/integrator.h, /Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
68 \_mixing/source/solver/phase/solve\_hyperbolic↔  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ \_equation.cc, 76  
\_mixing/include/mismix/particle/output.h, 69 /Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ \_mixing/source/support/utilities.cc, 76  
\_mixing/include/mismix/particle/particle.h, 69 ~BaseParticle  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::Particle::BaseParticle, 12  
\_mixing/include/mismix/particle/world.h, 70 ~Interface  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::Particle::Generator::Interface, 33  
\_mixing/include/mismix/phase/levelset.h, 70 ps\_mmm::Particle::Integrator::Interface, 34  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::Particle::Output::Interface, 32  
\_mixing/source/amr/amr.cc, 71 ~Levelset  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::immicable::Levelset, 37  
\_mixing/source/constitutive\_model/consti\_↔ ~World  
model.cc, 71 ps\_mmm::Particle::World, 60  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔  
\_mixing/source/main/main.cc, 71 ASCIIOutput  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::Particle::Output::ASCIIOutput, 10  
\_mixing/source/main/run.cc, 71 add\_mpi\_types  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::Particle::BaseParticle, 13  
\_mixing/source/particle/generator.cc, 72 ps\_mmm::Particle::Integrator::EulerIntegrator, 25  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::Particle::Integrator::HybridIntegrator, 28  
\_mixing/source/particle/integrator.cc, 72 ps\_mmm::Particle::Integrator::Interface, 35  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::Particle::Integrator::RK2Integrator, 54  
\_mixing/source/particle/output.cc, 73 ps\_mmm::Particle::Integrator::RK4Integrator, 56  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ add\_particle  
\_mixing/source/particle/particle.cc, 73 ps\_mmm::Particle::World, 61  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ advance\_timestep  
\_mixing/source/post\_process/extract\_data.cc, ps\_mmm::Particle::World, 61  
74 advection\_equation  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ ps\_mmm::immicable::Levelset, 37  
\_mixing/source/post\_process/post\_processing.↔ Assembly, 5  
cc, 74 Assembly::CopyData, 6  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ Assembly::CopyData::concentrMatrix  
\_mixing/source/pre\_process/constructor.cc, concentrMatrix, 19  
74 local\_dof\_indices, 19  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ local\_mass\_matrix, 19  
\_mixing/source/pre\_process/mesh\_in.cc, 74 local\_stiffness\_matrix, 19  
/Users/miranus/work/Devs/miscible\_mixing\_series/miscible↔ Assembly::CopyData::concentrMatrix < dim >, 19  
\_mixing/source/pre\_process/read\_and\_↔ Assembly::CopyData::concentrRHS

- concentrRHS, 21
- local\_dof\_indices, 21
- local\_rhs, 21
- matrix\_for\_bc, 21
- Assembly::CopyData::concentrRHS< dim >, 21
- Assembly::CopyData::diffusion\_step
  - diffusion\_step, 24
  - local\_dof\_indices, 24
  - local\_matrix, 24
  - local\_rhs, 24
- Assembly::CopyData::diffusion\_step< dim >, 24
- Assembly::CopyData::pressure\_rot\_step
  - local\_dof\_indices, 47
  - local\_matrix, 47
  - local\_rhs, 47
  - pressure\_rot\_step, 47
- Assembly::CopyData::pressure\_rot\_step< dim >, 47
- Assembly::CopyData::projection\_step
  - local\_dof\_indices, 48
  - local\_matrix, 48
  - local\_rhs, 48
  - projection\_step, 48
- Assembly::CopyData::projection\_step< dim >, 47
- Assembly::CopyData::relaxation\_div\_velocity\_step
  - local\_dof\_indices, 52
  - local\_matrix, 52
  - local\_rhs, 52
  - relaxation\_div\_velocity\_step, 51
- Assembly::CopyData::relaxation\_div\_velocity\_step< dim >, 51
- Assembly::Scratch, 6
- Assembly::Scratch::concentrMatrix
  - concentr\_fe\_values, 18
  - concentrMatrix, 18
  - grad\_phi\_T, 18
  - phi\_T, 18
- Assembly::Scratch::concentrMatrix< dim >, 18
- Assembly::Scratch::concentrRHS
  - concentr\_fe\_values, 20
  - concentrRHS, 20
  - fe\_velocity\_values, 20
  - grad\_phi\_T, 20
  - old\_concentr\_grads, 20
  - old\_concentr\_laplacians, 20
  - old\_concentr\_values, 20
  - old\_old\_concentr\_grads, 20
  - old\_old\_concentr\_laplacians, 20
  - old\_old\_concentr\_values, 20
  - old\_old\_strain\_rates, 20
  - old\_old\_velocity\_values, 20
  - old\_strain\_rates, 20
  - old\_velocity\_values, 20
  - phi\_T, 20
- Assembly::Scratch::concentrRHS< dim >, 19
- Assembly::Scratch::diffusion\_step
  - aux\_n\_minus\_1\_values, 23
  - aux\_n\_values, 23
  - concentr\_fe\_values, 23
  - concentr\_values, 23
  - diffusion\_step, 23
  - divergence\_phi\_u, 23
  - fe\_pressure\_values, 23
  - fe\_velocity\_values, 23
  - grad\_aux\_n\_minus\_1\_values, 23
  - grad\_aux\_n\_values, 23
  - grad\_grad\_aux\_n\_minus\_1\_values, 23
  - grad\_grad\_aux\_n\_values, 23
  - grad\_grad\_pre\_n\_values, 23
  - grad\_pre\_n\_values, 23
  - grad\_vel\_star\_values, 23
  - grads\_phi\_u, 23
  - laplacian\_vel\_star\_values, 23
  - phi\_u, 23
  - pre\_n\_values, 23
  - symm\_grads\_phi\_u, 23
  - symm\_grads\_vel\_star, 24
  - vel\_n\_minus\_1\_values, 24
  - vel\_n\_values, 24
  - vel\_star\_values, 24
- Assembly::Scratch::diffusion\_step< dim >, 22
- Assembly::Scratch::pressure\_rot\_step
  - aux\_sol\_values, 46
  - concentr\_fe\_values, 46
  - concentr\_values, 46
  - fe\_pressure\_values, 46
  - fe\_velocity\_values, 46
  - grad\_vel\_sol\_values, 46
  - phi\_p, 46
  - pre\_sol\_values, 46
  - pressure\_rot\_step, 46
  - symm\_grads\_vel\_sol, 47
- Assembly::Scratch::pressure\_rot\_step< dim >, 46
- Assembly::Scratch::projection\_step
  - concentr\_fe\_values, 49
  - concentr\_values, 49
  - div\_vel\_values, 49
  - fe\_auxiliary\_values, 49
  - fe\_velocity\_values, 49
  - grad\_vel\_n\_plus\_1\_values, 49
  - grads\_phi\_p, 49
  - phi\_p, 49
  - projection\_step, 49
- Assembly::Scratch::projection\_step< dim >, 48
- Assembly::Scratch::relaxation\_div\_velocity\_step
  - concentr\_fe\_values, 52
  - concentr\_values, 52
  - fe\_auxiliary\_values, 53
  - fe\_velocity\_values, 53
  - grad\_vel\_n\_plus\_1\_values, 53
  - grads\_phi\_p, 53
  - phi\_p, 53
  - relaxation\_div\_velocity\_step, 52
- Assembly::Scratch::relaxation\_div\_velocity\_step< dim >, 52
- Atwood\_number
  - UBC\_mis\_mixing::Parameters, 42

- aux\_n\_minus\_1\_values
  - Assembly::Scratch::diffusion\_step, 23
- aux\_n\_values
  - Assembly::Scratch::diffusion\_step, 23
- aux\_sol\_values
  - Assembly::Scratch::pressure\_rot\_step, 46
- BaseParticle
  - ps\_mmm::Particle::BaseParticle, 12
- CFL\_number
  - UBC\_mis\_mixing::Parameters, 42
- check\_current\_time\_step
  - UBC\_mis\_mixing::Parameters, 42
- check\_old\_time\_step
  - UBC\_mis\_mixing::Parameters, 42
- check\_particle\_count
  - ps\_mmm::Particle::World, 61
- check\_total\_real\_time
  - UBC\_mis\_mixing::Parameters, 42
- check\_total\_time
  - UBC\_mis\_mixing::Parameters, 42
- coar\_crit
  - UBC\_mis\_mixing::Parameters, 42
- coeff\_arti\_viscosity
  - UBC\_mis\_mixing::Parameters, 42
- coeff\_gamma\_grad\_div
  - UBC\_mis\_mixing::Parameters, 42
- coeff\_relax\_div\_velocity
  - UBC\_mis\_mixing::Parameters, 42
- communicator
  - ps\_mmm::Particle::Output::Interface, 32
- compute\_derived\_quantities\_vector
  - UBC\_mis\_mixing::Postprocessor, 45
- compute\_normal\_vector
  - ps\_mmm::immiscible::Levelset, 37
- computed\_time\_step
  - UBC\_mis\_mixing::Parameters, 42
- concentr\_fe\_values
  - Assembly::Scratch::concentrMatrix, 18
  - Assembly::Scratch::concentrRHS, 20
  - Assembly::Scratch::diffusion\_step, 23
  - Assembly::Scratch::pressure\_rot\_step, 46
  - Assembly::Scratch::projection\_step, 49
  - Assembly::Scratch::relaxation\_div\_velocity\_step, 52
- concentr\_values
  - Assembly::Scratch::diffusion\_step, 23
  - Assembly::Scratch::pressure\_rot\_step, 46
  - Assembly::Scratch::projection\_step, 49
  - Assembly::Scratch::relaxation\_div\_velocity\_step, 52
- concentrInitialValues
  - EquationData::concentrInitialValues, 17
- concentrInletValues
  - EquationData::concentrInletValues, 18
- concentrMatrix
  - Assembly::CopyData::concentrMatrix, 19
  - Assembly::Scratch::concentrMatrix, 18
- concentrRHS
  - Assembly::CopyData::concentrRHS, 21
  - Assembly::Scratch::concentrRHS, 20
- create\_generator\_object
  - ps\_mmm::Particle::Generator, 8
- create\_integrator\_object
  - ps\_mmm::Particle::Integrator, 9
- create\_output\_object
  - ps\_mmm::Particle::Output, 9
- data\_id
  - UBC\_mis\_mixing::Parameters, 42
- data\_len
  - ps\_mmm::Particle::BaseParticle, 14
  - ps\_mmm::Particle::Integrator::EulerIntegrator, 25
  - ps\_mmm::Particle::Integrator::HybridIntegrator, 29
  - ps\_mmm::Particle::Integrator::Interface, 35
  - ps\_mmm::Particle::Integrator::RK2Integrator, 54
  - ps\_mmm::Particle::Integrator::RK4Integrator, 56
- declare\_parameters
  - UBC\_mis\_mixing::Parameters, 42
- degree\_of\_concentr
  - UBC\_mis\_mixing::Parameters, 42
- degree\_of\_pressure
  - UBC\_mis\_mixing::Parameters, 42
- degree\_of\_velocity
  - UBC\_mis\_mixing::Parameters, 43
- depth\_direction
  - UBC\_mis\_mixing::Parameters, 43
- diffusion\_step
  - Assembly::CopyData::diffusion\_step, 24
  - Assembly::Scratch::diffusion\_step, 23
- dir\_concentration
  - UBC\_mis\_mixing::Parameters, 43
- div\_vel\_values
  - Assembly::Scratch::projection\_step, 49
- divergence\_phi\_u
  - Assembly::Scratch::diffusion\_step, 23
- domain\_boundary
  - UBC\_mis\_mixing::Parameters, 43
- domain\_size
  - UBC\_mis\_mixing::Parameters, 43
- downstream\_concentr
  - EquationData, 6
  - UBC\_mis\_mixing::Parameters, 43
- eps\_c
  - UBC\_mis\_mixing::Parameters, 43
- eps\_ns
  - UBC\_mis\_mixing::Parameters, 43
- eps\_v\_concentr
  - UBC\_mis\_mixing::Parameters, 43
- EquationData, 6
  - downstream\_concentr, 6
  - gravity\_accelation, 6
  - kinematic\_viscosity, 6
  - pipe\_diameter, 6
  - upstream\_concentr, 6
- EquationData::Inflow\_Velocity

- Inflow\_Velocity, 30
- init\_mean\_vel, 30
- value, 30
- vector\_value, 30
- vector\_value\_list, 30
- which\_inflow\_type, 30
- EquationData::Inflow\_Velocity< dim >, 30
- EquationData::Outflow\_Pressure
  - Froude\_number, 40
  - inclined\_angle, 40
  - Outflow\_Pressure, 40
  - value, 40
  - vector\_value, 40
  - vector\_value\_list, 40
- EquationData::Outflow\_Pressure< dim >, 39
- EquationData::concentrInitialValues
  - concentrInitialValues, 17
  - value, 17
  - vector\_value, 17
  - vector\_value\_list, 17
  - x, 17
- EquationData::concentrInitialValues< dim >, 16
- EquationData::concentrInletValues
  - concentrInletValues, 18
  - value, 18
  - vector\_value, 18
- EquationData::concentrInletValues< dim >, 17
- error\_threshold
  - UBC\_mis\_mixing::Parameters, 43
- exclude\_depth\_direction
  - UBC\_mis\_mixing::Parameters, 43
- fe\_auxiliary\_values
  - Assembly::Scratch::projection\_step, 49
  - Assembly::Scratch::relaxation\_div\_velocity\_step, 53
- fe\_pressure\_values
  - Assembly::Scratch::diffusion\_step, 23
  - Assembly::Scratch::pressure\_rot\_step, 46
- fe\_velocity\_values
  - Assembly::Scratch::concentrRHS, 20
  - Assembly::Scratch::diffusion\_step, 23
  - Assembly::Scratch::pressure\_rot\_step, 46
  - Assembly::Scratch::projection\_step, 49
  - Assembly::Scratch::relaxation\_div\_velocity\_step, 53
- file\_index
  - ps\_mmm::Particle::Output::Interface, 32
- find\_all\_cells
  - ps\_mmm::Particle::World, 61
- find\_cell
  - ps\_mmm::Particle::World, 61
- finished\_adding\_particles
  - ps\_mmm::Particle::World, 61
- flow\_direction
  - UBC\_mis\_mixing::Parameters, 43
- Froude\_number
  - EquationData::Outflow\_Pressure, 40
  - UBC\_mis\_mixing::Parameters, 43
- generate\_particles
  - ps\_mmm::Particle::Generator::Interface, 33
  - ps\_mmm::Particle::Generator::RandomUniform↔Generator, 51
- generator\_object\_names
  - ps\_mmm::Particle::Generator, 8
- get\_data\_component\_interpretation
  - UBC\_mis\_mixing::Postprocessor, 45
- get\_dof\_handler
  - ps\_mmm::Particle::World, 61
- get\_global\_particle\_count
  - ps\_mmm::Particle::World, 61
- get\_id
  - ps\_mmm::Particle::BaseParticle, 14
- get\_location
  - ps\_mmm::Particle::BaseParticle, 14
- get\_mapping
  - ps\_mmm::Particle::World, 62
- get\_names
  - UBC\_mis\_mixing::Postprocessor, 45
- get\_needed\_update\_flags
  - UBC\_mis\_mixing::Postprocessor, 45
- get\_parameters
  - ps\_mmm::immiscible::Levelset, 37
- get\_particle\_velocities
  - ps\_mmm::Particle::World, 62
- get\_particles
  - ps\_mmm::Particle::World, 62
- get\_solution
  - ps\_mmm::Particle::World, 62
- get\_triangulation
  - ps\_mmm::Particle::World, 62
- get\_velocity
  - ps\_mmm::Particle::BaseParticle, 14
- grad\_aux\_n\_minus\_1\_values
  - Assembly::Scratch::diffusion\_step, 23
- grad\_aux\_n\_values
  - Assembly::Scratch::diffusion\_step, 23
- grad\_grad\_aux\_n\_minus\_1\_values
  - Assembly::Scratch::diffusion\_step, 23
- grad\_grad\_aux\_n\_values
  - Assembly::Scratch::diffusion\_step, 23
- grad\_grad\_pre\_n\_values
  - Assembly::Scratch::diffusion\_step, 23
- grad\_phi\_T
  - Assembly::Scratch::concentrMatrix, 18
  - Assembly::Scratch::concentrRHS, 20
- grad\_pre\_n\_values
  - Assembly::Scratch::diffusion\_step, 23
- grad\_vel\_n\_plus\_1\_values
  - Assembly::Scratch::projection\_step, 49
  - Assembly::Scratch::relaxation\_div\_velocity\_step, 53
- grad\_vel\_sol\_values
  - Assembly::Scratch::pressure\_rot\_step, 46
- grad\_vel\_star\_values
  - Assembly::Scratch::diffusion\_step, 23
- grads\_phi\_p

- Assembly::Scratch::projection\_step, 49
- Assembly::Scratch::relaxation\_div\_velocity\_step, 53
- grads\_phi\_u
  - Assembly::Scratch::diffusion\_step, 23
- gravitiy\_accelation
  - EquationData, 6
- HDF5Output
  - ps\_mmm::Particle::Output::HDF5Output, 27
- HybridIntegrator
  - ps\_mmm::Particle::Integrator::HybridIntegrator, 28
- inclined\_angle
  - EquationData::Outflow\_Pressure, 40
  - UBC\_mis\_mixing::Parameters, 43
- inclined\_angle\_vector
  - UBC\_mis\_mixing::Parameters, 43
- index\_for\_restart
  - UBC\_mis\_mixing::Parameters, 43
- Inflow\_Velocity
  - EquationData::Inflow\_Velocity, 30
- init
  - ps\_mmm::Particle::World, 62
- init\_mean\_vel
  - EquationData::Inflow\_Velocity, 30
- init\_sep\_x
  - UBC\_mis\_mixing::Parameters, 43
- initial\_distribution
  - ps\_mmm::immiscible::Levelset, 38
- inlet\_pressure
  - UBC\_mis\_mixing::Parameters, 43
- input\_mesh\_file
  - UBC\_mis\_mixing::Parameters, 43
- integrate\_step
  - ps\_mmm::Particle::Integrator::EulerIntegrator, 26
  - ps\_mmm::Particle::Integrator::HybridIntegrator, 29
  - ps\_mmm::Particle::Integrator::Interface, 35
  - ps\_mmm::Particle::Integrator::RK2Integrator, 54
  - ps\_mmm::Particle::Integrator::RK4Integrator, 56
- integrator\_object\_names
  - ps\_mmm::Particle::Integrator, 9
- Interface
  - ps\_mmm::Particle::Generator::Interface, 33
  - ps\_mmm::Particle::Integrator::Interface, 34
  - ps\_mmm::Particle::Output::Interface, 31
- intial\_ratio\_refinement
  - UBC\_mis\_mixing::Parameters, 43
- is\_density\_stable\_flow
  - UBC\_mis\_mixing::Parameters, 43
- is\_restart
  - UBC\_mis\_mixing::Parameters, 43
- is\_symmetry\_boundary
  - UBC\_mis\_mixing::Parameters, 43
- is\_verbal\_output
  - UBC\_mis\_mixing::Parameters, 43
- ist\_add\_reinit
  - UBC\_mis\_mixing::Parameters, 43
- ist\_flow\_source
  - UBC\_mis\_mixing::Parameters, 43
- ist\_optimization\_method
  - UBC\_mis\_mixing::Parameters, 43
- ist\_pressure\_boundary
  - UBC\_mis\_mixing::Parameters, 43
- ist\_projection\_method
  - UBC\_mis\_mixing::Parameters, 43
- ist\_uniform\_flow
  - UBC\_mis\_mixing::Parameters, 43
- kinematic\_viscosity
  - EquationData, 6
- kry\_size
  - UBC\_mis\_mixing::Parameters, 43
- laplacian\_vel\_star\_values
  - Assembly::Scratch::diffusion\_step, 23
- latitude\_direction
  - UBC\_mis\_mixing::Parameters, 44
- length\_of\_domain
  - UBC\_mis\_mixing::Parameters, 44
- LevelInd
  - ps\_mmm::Particle, 7
- Levelset
  - ps\_mmm::immiscible::Levelset, 37
- local
  - ps\_mmm::Particle::BaseParticle, 14
- local\_dof\_indices
  - Assembly::CopyData::concentrMatrix, 19
  - Assembly::CopyData::concentrRHS, 21
  - Assembly::CopyData::diffusion\_step, 24
  - Assembly::CopyData::pressure\_rot\_step, 47
  - Assembly::CopyData::projection\_step, 48
  - Assembly::CopyData::relaxation\_div\_velocity\_↔ step, 52
- local\_mass\_matrix
  - Assembly::CopyData::concentrMatrix, 19
- local\_matrix
  - Assembly::CopyData::diffusion\_step, 24
  - Assembly::CopyData::pressure\_rot\_step, 47
  - Assembly::CopyData::projection\_step, 48
  - Assembly::CopyData::relaxation\_div\_velocity\_↔ step, 52
- local\_rhs
  - Assembly::CopyData::concentrRHS, 21
  - Assembly::CopyData::diffusion\_step, 24
  - Assembly::CopyData::pressure\_rot\_step, 47
  - Assembly::CopyData::projection\_step, 48
  - Assembly::CopyData::relaxation\_div\_velocity\_↔ step, 52
- local\_stiffness\_matrix
  - Assembly::CopyData::concentrMatrix, 19
- MPIDataInfo
  - ps\_mmm::Particle::MPIDataInfo, 38
- main
  - main.cc, 71
- main.cc
  - main, 71



- mark\_particles\_for\_check
  - ps\_mmm::Particle::World, 62
- matrix\_for\_bc
  - Assembly::CopyData::concentrRHS, 21
- max\_grid\_level
  - UBC\_mis\_mixing::Parameters, 44
- maximum\_coeff\_arti\_viscosity
  - UBC\_mis\_mixing::Parameters, 44
- mean\_velocity\_inlet
  - UBC\_mis\_mixing::Parameters, 44
- mean\_viscosity
  - UBC\_mis\_mixing::Parameters, 44
- mesh\_speed
  - UBC\_mis\_mixing::Parameters, 44
- move\_particles\_back\_in\_mesh
  - ps\_mmm::Particle::World, 63
- mpi\_comm
  - ps\_mmm::Particle::World, 63
- n\_elements
  - ps\_mmm::Particle::MPIDataInfo, 38
- n\_pow\_law
  - UBC\_mis\_mixing::Parameters, 44
- name
  - ps\_mmm::Particle::MPIDataInfo, 38
- no\_refine\_period
  - UBC\_mis\_mixing::Parameters, 44
- no\_steps\_for\_buffering
  - UBC\_mis\_mixing::Parameters, 44
- no\_test\_case
  - UBC\_mis\_mixing::Parameters, 44
- NullOutput
  - ps\_mmm::Particle::Output::NullOutput, 39
- num\_element\_size
  - UBC\_mis\_mixing::Parameters, 44
- num\_slices\_domain
  - UBC\_mis\_mixing::Parameters, 44
- number\_slices\_coarse\_mesh
  - UBC\_mis\_mixing::Parameters, 44
- old\_concentr\_grads
  - Assembly::Scratch::concentrRHS, 20
- old\_concentr\_laplacians
  - Assembly::Scratch::concentrRHS, 20
- old\_concentr\_values
  - Assembly::Scratch::concentrRHS, 20
- old\_old\_concentr\_grads
  - Assembly::Scratch::concentrRHS, 20
- old\_old\_concentr\_laplacians
  - Assembly::Scratch::concentrRHS, 20
- old\_old\_concentr\_values
  - Assembly::Scratch::concentrRHS, 20
- old\_old\_strain\_rates
  - Assembly::Scratch::concentrRHS, 20
- old\_old\_velocity\_values
  - Assembly::Scratch::concentrRHS, 20
- old\_strain\_rates
  - Assembly::Scratch::concentrRHS, 20
- old\_velocity\_values
  - Assembly::Scratch::concentrRHS, 20
- Outflow\_Pressure
  - EquationData::Outflow\_Pressure, 40
- output\_dir
  - ps\_mmm::Particle::Output::Interface, 32
- output\_fac\_data
  - UBC\_mis\_mixing::Parameters, 44
- output\_fac\_vtu
  - UBC\_mis\_mixing::Parameters, 44
- output\_object\_names
  - ps\_mmm::Particle::Output, 10
- output\_particle\_data
  - ps\_mmm::Particle::Output::ASCIIOutput, 10
  - ps\_mmm::Particle::Output::HDF5Output, 27
  - ps\_mmm::Particle::Output::Interface, 32
  - ps\_mmm::Particle::Output::NullOutput, 39
  - ps\_mmm::Particle::Output::VTUOutput, 59
- PARTICLE\_XFER\_TAG
  - ps\_mmm::Particle, 7
- Parameters
  - UBC\_mis\_mixing::Parameters, 42
- parse\_parameters
  - UBC\_mis\_mixing::Parameters, 42
- phi\_T
  - Assembly::Scratch::concentrMatrix, 18
  - Assembly::Scratch::concentrRHS, 20
- phi\_p
  - Assembly::Scratch::pressure\_rot\_step, 46
  - Assembly::Scratch::projection\_step, 49
  - Assembly::Scratch::relaxation\_div\_velocity\_step, 53
- phi\_u
  - Assembly::Scratch::diffusion\_step, 23
- pipe\_diameter
  - EquationData, 6
- Postprocessor
  - UBC\_mis\_mixing::Postprocessor, 45
- pre\_n\_values
  - Assembly::Scratch::diffusion\_step, 23
- pre\_sol\_values
  - Assembly::Scratch::pressure\_rot\_step, 46
- pressure\_rot\_step
  - Assembly::CopyData::pressure\_rot\_step, 47
  - Assembly::Scratch::pressure\_rot\_step, 46
- projection\_step
  - Assembly::CopyData::projection\_step, 48
  - Assembly::Scratch::projection\_step, 49
- ps\_mmm, 7
- ps\_mmm::Particle, 7
  - Levellnd, 7
  - PARTICLE\_XFER\_TAG, 7
- ps\_mmm::Particle::BaseParticle
  - ~BaseParticle, 12
  - add\_mpi\_types, 13
  - BaseParticle, 12
  - data\_len, 14
  - get\_id, 14
  - get\_location, 14



- get\_velocity, 14
- local, 14
- read\_data, 14
- set\_local, 14
- set\_location, 16
- set\_vel\_check, 16
- set\_velocity, 16
- vel\_check, 16
- write\_data, 16
- ps\_mmm::Particle::BaseParticle< dim >, 11
- ps\_mmm::Particle::DataParticle< dim, data\_dim >, 21
- ps\_mmm::Particle::Generator, 8
  - create\_generator\_object, 8
  - generator\_object\_names, 8
- ps\_mmm::Particle::Generator::Interface
  - ~Interface, 33
  - generate\_particles, 33
  - Interface, 33
- ps\_mmm::Particle::Generator::Interface< dim, T >, 33
- ps\_mmm::Particle::Generator::RandomUniform←Generator
  - generate\_particles, 51
  - RandomUniformGenerator, 50
  - uniform\_random\_particles\_in\_subdomain, 51
- ps\_mmm::Particle::Generator::RandomUniform←Generator< dim, T >, 49
- ps\_mmm::Particle::Integrator, 8
  - create\_integrator\_object, 9
  - integrator\_object\_names, 9
- ps\_mmm::Particle::Integrator::EulerIntegrator
  - add\_mpi\_types, 25
  - data\_len, 25
  - integrate\_step, 26
  - read\_data, 26
  - write\_data, 26
- ps\_mmm::Particle::Integrator::EulerIntegrator< dim, T >, 25
- ps\_mmm::Particle::Integrator::HybridIntegrator
  - add\_mpi\_types, 28
  - data\_len, 29
  - HybridIntegrator, 28
  - integrate\_step, 29
  - read\_data, 29
  - write\_data, 29
- ps\_mmm::Particle::Integrator::HybridIntegrator< dim, T >, 28
- ps\_mmm::Particle::Integrator::Interface
  - ~Interface, 34
  - add\_mpi\_types, 35
  - data\_len, 35
  - integrate\_step, 35
  - Interface, 34
  - read\_data, 35
  - write\_data, 37
- ps\_mmm::Particle::Integrator::Interface< dim, T >, 34
- ps\_mmm::Particle::Integrator::RK2Integrator
  - add\_mpi\_types, 54
  - data\_len, 54
  - integrate\_step, 54
  - RK2Integrator, 54
  - read\_data, 54
  - write\_data, 55
- ps\_mmm::Particle::Integrator::RK2Integrator< dim, T >, 53
- ps\_mmm::Particle::Integrator::RK4Integrator
  - add\_mpi\_types, 56
  - data\_len, 56
  - integrate\_step, 56
  - RK4Integrator, 56
  - read\_data, 57
  - write\_data, 57
- ps\_mmm::Particle::Integrator::RK4Integrator< dim, T >, 55
- ps\_mmm::Particle::MPIDataInfo, 38
  - MPIDataInfo, 38
  - n\_elements, 38
  - name, 38
- ps\_mmm::Particle::Output, 9
  - create\_output\_object, 9
  - output\_object\_names, 10
- ps\_mmm::Particle::Output::ASCIIOutput
  - ASCIIOutput, 10
  - output\_particle\_data, 10
- ps\_mmm::Particle::Output::ASCIIOutput< dim, T >, 10
- ps\_mmm::Particle::Output::HDF5Output
  - HDF5Output, 27
  - output\_particle\_data, 27
- ps\_mmm::Particle::Output::HDF5Output< dim, T >, 27
- ps\_mmm::Particle::Output::Interface
  - ~Interface, 32
  - communicator, 32
  - file\_index, 32
  - Interface, 31
  - output\_dir, 32
  - output\_particle\_data, 32
  - serialize, 32
- ps\_mmm::Particle::Output::Interface< dim, T >, 31
- ps\_mmm::Particle::Output::NullOutput
  - NullOutput, 39
  - output\_particle\_data, 39
- ps\_mmm::Particle::Output::NullOutput< dim, T >, 38
- ps\_mmm::Particle::Output::VTUOutput
  - output\_particle\_data, 59
  - VTUOutput, 58
- ps\_mmm::Particle::Output::VTUOutput< dim, T >, 58
- ps\_mmm::Particle::World
  - ~World, 60
  - add\_particle, 61
  - advance\_timestep, 61
  - check\_particle\_count, 61
  - find\_all\_cells, 61
  - find\_cell, 61
  - finished\_adding\_particles, 61
  - get\_dof\_handler, 61
  - get\_global\_particle\_count, 61
  - get\_mapping, 62

- get\_particle\_velocities, 62
- get\_particles, 62
- get\_solution, 62
- get\_triangulation, 62
- init, 62
- mark\_particles\_for\_check, 62
- move\_particles\_back\_in\_mesh, 63
- mpi\_comm, 63
- send\_rcv\_particles, 63
- serialize, 63
- set\_dof\_handler, 63
- set\_integrator, 63
- set\_mapping, 63
- set\_mpi\_comm, 63
- set\_solution, 65
- set\_triangulation, 65
- World, 60
- ps\_mmm::Particle::World< dim, T >, 59
- ps\_mmm::immicable, 7
- ps\_mmm::immicable::Levelset
  - ~Levelset, 37
  - advection\_equation, 37
  - compute\_normal\_vector, 37
  - get\_parameters, 37
  - initial\_distribution, 38
  - Levelset, 37
  - reinitialization\_levelset, 38
- ps\_mmm::immicable::Levelset< dim >, 37
- RK2Integrator
  - ps\_mmm::Particle::Integrator::RK2Integrator, 54
- RK4Integrator
  - ps\_mmm::Particle::Integrator::RK4Integrator, 56
- RandomUniformGenerator
  - ps\_mmm::Particle::Generator::RandomUniformGenerator, 50
- ratio\_pow\_law
  - UBC\_mis\_mixing::Parameters, 44
- read\_data
  - ps\_mmm::Particle::BaseParticle, 14
  - ps\_mmm::Particle::Integrator::EulerIntegrator, 26
  - ps\_mmm::Particle::Integrator::HybridIntegrator, 29
  - ps\_mmm::Particle::Integrator::Interface, 35
  - ps\_mmm::Particle::Integrator::RK2Integrator, 54
  - ps\_mmm::Particle::Integrator::RK4Integrator, 57
- ref\_crit
  - UBC\_mis\_mixing::Parameters, 44
- reference\_length
  - UBC\_mis\_mixing::Parameters, 44
- reference\_time
  - UBC\_mis\_mixing::Parameters, 44
- reference\_velocity
  - UBC\_mis\_mixing::Parameters, 44
- reinitialization\_levelset
  - ps\_mmm::immicable::Levelset, 38
- relaxation\_div\_velocity\_step
  - Assembly::CopyData::relaxation\_div\_velocity\_step, 51
  - Assembly::Scratch::relaxation\_div\_velocity\_step, 52
- restart\_no\_timestep
  - UBC\_mis\_mixing::Parameters, 44
- Reynolds\_number
  - UBC\_mis\_mixing::Parameters, 44
- run
  - UBC\_mis\_mixing, 57
- save\_fac\_period
  - UBC\_mis\_mixing::Parameters, 44
- send\_rcv\_particles
  - ps\_mmm::Particle::World, 63
- serialize
  - ps\_mmm::Particle::Output::Interface, 32
  - ps\_mmm::Particle::World, 63
- set\_dof\_handler
  - ps\_mmm::Particle::World, 63
- set\_integrator
  - ps\_mmm::Particle::World, 63
- set\_local
  - ps\_mmm::Particle::BaseParticle, 14
- set\_location
  - ps\_mmm::Particle::BaseParticle, 16
- set\_mapping
  - ps\_mmm::Particle::World, 63
- set\_mpi\_comm
  - ps\_mmm::Particle::World, 63
- set\_solution
  - ps\_mmm::Particle::World, 65
- set\_triangulation
  - ps\_mmm::Particle::World, 65
- set\_vel\_check
  - ps\_mmm::Particle::BaseParticle, 16
- set\_velocity
  - ps\_mmm::Particle::BaseParticle, 16
- stabilization\_alpha
  - UBC\_mis\_mixing::Parameters, 44
- stabilization\_beta
  - UBC\_mis\_mixing::Parameters, 44
- stabilization\_c\_R
  - UBC\_mis\_mixing::Parameters, 44
- symm\_grads\_phi\_u
  - Assembly::Scratch::diffusion\_step, 23
- symm\_grads\_vel\_sol
  - Assembly::Scratch::pressure\_rot\_step, 47
- symm\_grads\_vel\_star
  - Assembly::Scratch::diffusion\_step, 24
- tau\_step
  - UBC\_mis\_mixing::Parameters, 44
- type\_adaptivity\_rule
  - UBC\_mis\_mixing::Parameters, 44
- UBC\_mis\_mixing
  - run, 57
  - UBC\_mis\_mixing, 57
- UBC\_mis\_mixing< dim >, 57
- UBC\_mis\_mixing< dim >::Parameters, 40

- UBC\_mis\_mixing< dim >::Postprocessor< dim >, 45
- UBC\_mis\_mixing::Parameters
  - Atwood\_number, 42
  - CFL\_number, 42
  - check\_current\_time\_step, 42
  - check\_old\_time\_step, 42
  - check\_total\_real\_time, 42
  - check\_total\_time, 42
  - coar\_crit, 42
  - coeff\_arti\_viscosity, 42
  - coeff\_gamma\_grad\_div, 42
  - coeff\_relax\_div\_velocity, 42
  - computed\_time\_step, 42
  - data\_id, 42
  - declare\_parameters, 42
  - degree\_of\_concentr, 42
  - degree\_of\_pressure, 42
  - degree\_of\_velocity, 43
  - depth\_direction, 43
  - dir\_concentration, 43
  - domain\_boundary, 43
  - domain\_size, 43
  - downstream\_concentr, 43
  - eps\_c, 43
  - eps\_ns, 43
  - eps\_v\_concentr, 43
  - error\_threshold, 43
  - exclude\_depth\_direction, 43
  - flow\_direction, 43
  - Froude\_number, 43
  - inclined\_angle, 43
  - inclined\_angle\_vector, 43
  - index\_for\_restart, 43
  - init\_sep\_x, 43
  - inlet\_pressure, 43
  - input\_mesh\_file, 43
  - intial\_ratio\_refinement, 43
  - is\_density\_stable\_flow, 43
  - is\_restart, 43
  - is\_symmetry\_boundary, 43
  - is\_verbal\_output, 43
  - ist\_add\_reinit, 43
  - ist\_flow\_source, 43
  - ist\_optimization\_method, 43
  - ist\_pressure\_boundary, 43
  - ist\_projection\_method, 43
  - ist\_uniform\_flow, 43
  - kry\_size, 43
  - latitude\_direction, 44
  - length\_of\_domain, 44
  - max\_grid\_level, 44
  - maximum\_coeff\_arti\_viscosity, 44
  - mean\_velocity\_inlet, 44
  - mean\_viscosity, 44
  - mesh\_speed, 44
  - n\_pow\_law, 44
  - no\_refine\_period, 44
  - no\_steps\_for\_buffering, 44
  - no\_test\_case, 44
  - num\_element\_size, 44
  - num\_slices\_domain, 44
  - number\_slices\_coarse\_mesh, 44
  - output\_fac\_data, 44
  - output\_fac\_vtu, 44
  - Parameters, 42
  - parse\_parameters, 42
  - ratio\_pow\_law, 44
  - ref\_crit, 44
  - reference\_length, 44
  - reference\_time, 44
  - reference\_velocity, 44
  - restart\_no\_timestep, 44
  - Reynolds\_number, 44
  - save\_fac\_period, 44
  - stabilization\_alpha, 44
  - stabilization\_beta, 44
  - stabilization\_c\_R, 44
  - tau\_step, 44
  - type\_adaptivity\_rule, 44
  - upstream\_concentr, 44
  - viscosity\_ratio, 44
  - which\_interpl\_c, 45
  - which\_method\_for\_c, 45
- UBC\_mis\_mixing::Postprocessor
  - compute\_derived\_quantities\_vector, 45
  - get\_data\_component\_interpretation, 45
  - get\_names, 45
  - get\_needed\_update\_flags, 45
  - Postprocessor, 45
- uniform\_random\_particles\_in\_subdomain
  - ps\_mmm::Particle::Generator::RandomUniformGenerator, 51
- upstream\_concentr
  - EquationData, 6
  - UBC\_mis\_mixing::Parameters, 44
- VTUOutput
  - ps\_mmm::Particle::Output::VTUOutput, 58
- value
  - EquationData::Inflow\_Velocity, 30
  - EquationData::Outflow\_Pressure, 40
  - EquationData::concentrInitialValues, 17
  - EquationData::concentrInletValues, 18
- vector\_value
  - EquationData::Inflow\_Velocity, 30
  - EquationData::Outflow\_Pressure, 40
  - EquationData::concentrInitialValues, 17
  - EquationData::concentrInletValues, 18
- vector\_value\_list
  - EquationData::Inflow\_Velocity, 30
  - EquationData::Outflow\_Pressure, 40
  - EquationData::concentrInitialValues, 17
- vel\_check
  - ps\_mmm::Particle::BaseParticle, 16
- vel\_n\_minus\_1\_values
  - Assembly::Scratch::diffusion\_step, 24
- vel\_n\_values

- Assembly::Scratch::diffusion\_step, [24](#)
- vel\_star\_values
  - Assembly::Scratch::diffusion\_step, [24](#)
- viscosity\_ratio
  - UBC\_mis\_mixing::Parameters, [44](#)
- which\_inflow\_type
  - EquationData::Inflow\_Velocity, [30](#)
- which\_interpl\_c
  - UBC\_mis\_mixing::Parameters, [45](#)
- which\_method\_for\_c
  - UBC\_mis\_mixing::Parameters, [45](#)
- World
  - ps\_mmm::Particle::World, [60](#)
- write\_data
  - ps\_mmm::Particle::BaseParticle, [16](#)
  - ps\_mmm::Particle::Integrator::EulerIntegrator, [26](#)
  - ps\_mmm::Particle::Integrator::HybridIntegrator, [29](#)
  - ps\_mmm::Particle::Integrator::Interface, [37](#)
  - ps\_mmm::Particle::Integrator::RK2Integrator, [55](#)
  - ps\_mmm::Particle::Integrator::RK4Integrator, [57](#)
- x
  - EquationData::concentrInitialValues, [17](#)