

HEWES : Heisenberg-Euler Weak-Field Expansion Simulator

v0.2.2

Generated by Doxygen 1.9.5

1 HEWES: Heisenberg-Euler Weak-Field Expansion Simulator	1
1.1 Contents	1
1.2 Required software	2
1.3 Short user manual	2
1.3.1 Note on simulation settings	3
1.3.2 Note on resource occupation	3
1.3.3 Note on the output analysis	4
1.4 Authors	4
2 Hierarchical Index	5
2.1 Class Hierarchy	5
3 Data Structure Index	7
3.1 Data Structures	7
4 File Index	9
4.1 File List	9
5 Data Structure Documentation	11
5.1 Gauss1D Class Reference	11
5.1.1 Detailed Description	12
5.1.2 Constructor & Destructor Documentation	12
5.1.2.1 Gauss1D()	12
5.1.3 Member Function Documentation	13
5.1.3.1 addToSpace()	13
5.1.4 Field Documentation	14
5.1.4.1 kx	14
5.1.4.2 ky	14
5.1.4.3 kz	14
5.1.4.4 phig	14
5.1.4.5 phix	15
5.1.4.6 phiy	15
5.1.4.7 phiz	15
5.1.4.8 px	15
5.1.4.9 py	16
5.1.4.10 pz	16
5.1.4.11 x0x	16
5.1.4.12 x0y	16
5.1.4.13 x0z	17
5.2 Gauss2D Class Reference	17
5.2.1 Detailed Description	18
5.2.2 Constructor & Destructor Documentation	18
5.2.2.1 Gauss2D()	18
5.2.3 Member Function Documentation	19

5.2.3.1 addToSpace()	19
5.2.4 Field Documentation	20
5.2.4.1 A1	20
5.2.4.2 A2	20
5.2.4.3 Amp	20
5.2.4.4 axis	21
5.2.4.5 dis	21
5.2.4.6 lambda	21
5.2.4.7 Ph0	21
5.2.4.8 PhA	22
5.2.4.9 phip	22
5.2.4.10 w0	22
5.2.4.11 zr	22
5.3 Gauss3D Class Reference	23
5.3.1 Detailed Description	24
5.3.2 Constructor & Destructor Documentation	24
5.3.2.1 Gauss3D()	24
5.3.3 Member Function Documentation	25
5.3.3.1 addToSpace()	25
5.3.4 Field Documentation	25
5.3.4.1 A1	26
5.3.4.2 A2	26
5.3.4.3 Amp	26
5.3.4.4 axis	26
5.3.4.5 dis	27
5.3.4.6 lambda	27
5.3.4.7 Ph0	27
5.3.4.8 PhA	27
5.3.4.9 phip	28
5.3.4.10 w0	28
5.3.4.11 zr	28
5.4 gaussian1D Struct Reference	28
5.4.1 Detailed Description	29
5.4.2 Field Documentation	29
5.4.2.1 k	29
5.4.2.2 p	29
5.4.2.3 phi	29
5.4.2.4 phig	29
5.4.2.5 x0	30
5.5 gaussian2D Struct Reference	30
5.5.1 Detailed Description	30
5.5.2 Field Documentation	30

5.5.2.1 amp	30
5.5.2.2 axis	31
5.5.2.3 ph0	31
5.5.2.4 phA	31
5.5.2.5 phiP	31
5.5.2.6 w0	31
5.5.2.7 x0	32
5.5.2.8 zr	32
5.6 gaussian3D Struct Reference	32
5.6.1 Detailed Description	32
5.6.2 Field Documentation	32
5.6.2.1 amp	33
5.6.2.2 axis	33
5.6.2.3 ph0	33
5.6.2.4 phA	33
5.6.2.5 phiP	33
5.6.2.6 w0	34
5.6.2.7 x0	34
5.6.2.8 zr	34
5.7 ICSetter Class Reference	34
5.7.1 Detailed Description	35
5.7.2 Member Function Documentation	35
5.7.2.1 add()	36
5.7.2.2 addGauss1D()	36
5.7.2.3 addGauss2D()	37
5.7.2.4 addGauss3D()	38
5.7.2.5 addPlaneWave1D()	38
5.7.2.6 addPlaneWave2D()	39
5.7.2.7 addPlaneWave3D()	40
5.7.2.8 eval()	40
5.7.3 Field Documentation	41
5.7.3.1 gauss1Ds	41
5.7.3.2 gauss2Ds	41
5.7.3.3 gauss3Ds	42
5.7.3.4 planeWaves1D	42
5.7.3.5 planeWaves2D	42
5.7.3.6 planeWaves3D	42
5.8 Lattice Class Reference	43
5.8.1 Detailed Description	44
5.8.2 Constructor & Destructor Documentation	44
5.8.2.1 Lattice()	45
5.8.3 Member Function Documentation	45

5.8.3.1 get_dataPointDimension()	45
5.8.3.2 get_dx()	46
5.8.3.3 get_dy()	46
5.8.3.4 get_dz()	46
5.8.3.5 get_ghostLayerWidth()	47
5.8.3.6 get_stencilOrder()	47
5.8.3.7 get_tot_lx()	48
5.8.3.8 get_tot_ly()	49
5.8.3.9 get_tot_lz()	49
5.8.3.10 get_tot_noDP()	50
5.8.3.11 get_tot_noP()	50
5.8.3.12 get_tot_nx()	50
5.8.3.13 get_tot_ny()	50
5.8.3.14 get_tot_nz()	51
5.8.3.15 initializeCommunicator()	51
5.8.3.16 setDiscreteDimensions()	52
5.8.3.17 setPhysicalDimensions()	52
5.8.4 Field Documentation	53
5.8.4.1 comm	53
5.8.4.2 dataPointDimension	53
5.8.4.3 dx	54
5.8.4.4 dy	54
5.8.4.5 dz	54
5.8.4.6 ghostLayerWidth	54
5.8.4.7 my_prc	55
5.8.4.8 n_prc	55
5.8.4.9 statusFlags	55
5.8.4.10 stencilOrder	55
5.8.4.11 sunctx	56
5.8.4.12 tot_lx	56
5.8.4.13 tot_ly	56
5.8.4.14 tot_lz	56
5.8.4.15 tot_noDP	57
5.8.4.16 tot_noP	57
5.8.4.17 tot_nx	57
5.8.4.18 tot_ny	57
5.8.4.19 tot_nz	58
5.9 LatticePatch Class Reference	58
5.9.1 Detailed Description	61
5.9.2 Constructor & Destructor Documentation	61
5.9.2.1 LatticePatch()	62
5.9.2.2 ~LatticePatch()	62

5.9.3 Member Function Documentation	62
5.9.3.1 checkFlag()	63
5.9.3.2 derive()	64
5.9.3.3 derotate()	69
5.9.3.4 discreteSize()	71
5.9.3.5 exchangeGhostCells()	72
5.9.3.6 generateTranslocationLookup()	74
5.9.3.7 getDelta()	76
5.9.3.8 initializeBuffers()	77
5.9.3.9 origin()	78
5.9.3.10 rotateIntoEigen()	79
5.9.3.11 rotateToX()	80
5.9.3.12 rotateToY()	82
5.9.3.13 rotateToZ()	83
5.9.4 Friends And Related Function Documentation	84
5.9.4.1 generatePatchwork	84
5.9.5 Field Documentation	85
5.9.5.1 buffData	85
5.9.5.2 buffX	85
5.9.5.3 buffY	86
5.9.5.4 buffZ	86
5.9.5.5 du	86
5.9.5.6 duData	86
5.9.5.7 duLocal	87
5.9.5.8 dx	87
5.9.5.9 dy	87
5.9.5.10 dz	87
5.9.5.11 envelopeLattice	88
5.9.5.12 gCLData	88
5.9.5.13 gCRData	88
5.9.5.14 ghostCellLeft	88
5.9.5.15 ghostCellLeftToSend	89
5.9.5.16 ghostCellRight	89
5.9.5.17 ghostCellRightToSend	89
5.9.5.18 ghostCells	89
5.9.5.19 ghostCellsToSend	89
5.9.5.20 ID	90
5.9.5.21 lgcTox	90
5.9.5.22 lgcToy	90
5.9.5.23 lgcToz	90
5.9.5.24 Llx	91
5.9.5.25 Lly	91

5.9.5.26 Llz	91
5.9.5.27 lx	91
5.9.5.28 ly	92
5.9.5.29 lz	92
5.9.5.30 nx	92
5.9.5.31 ny	92
5.9.5.32 nz	93
5.9.5.33 rgcTox	93
5.9.5.34 rgcToy	93
5.9.5.35 rgcToz	93
5.9.5.36 statusFlags	94
5.9.5.37 u	94
5.9.5.38 uAux	94
5.9.5.39 uAuxData	94
5.9.5.40 uData	95
5.9.5.41 uLocal	95
5.9.5.42 uTox	95
5.9.5.43 uToy	95
5.9.5.44 uToz	96
5.9.5.45 x0	96
5.9.5.46 xTou	96
5.9.5.47 y0	96
5.9.5.48 yTou	97
5.9.5.49 z0	97
5.9.5.50 zTou	97
5.10 OutputManager Class Reference	97
5.10.1 Detailed Description	98
5.10.2 Constructor & Destructor Documentation	98
5.10.2.1 OutputManager()	99
5.10.3 Member Function Documentation	99
5.10.3.1 generateOutputFolder()	99
5.10.3.2 getSimCode()	100
5.10.3.3 outUState()	101
5.10.3.4 set_outputStyle()	103
5.10.3.5 SimCodeGenerator()	103
5.10.4 Field Documentation	104
5.10.4.1 outputStyle	104
5.10.4.2 Path	104
5.10.4.3 simCode	105
5.11 PlaneWave Class Reference	105
5.11.1 Detailed Description	106
5.11.2 Field Documentation	106

5.11.2.1 kx	106
5.11.2.2 ky	106
5.11.2.3 kz	106
5.11.2.4 phix	107
5.11.2.5 phiy	107
5.11.2.6 phiz	107
5.11.2.7 px	107
5.11.2.8 py	108
5.11.2.9 pz	108
5.12 planewave Struct Reference	108
5.12.1 Detailed Description	108
5.12.2 Field Documentation	109
5.12.2.1 k	109
5.12.2.2 p	109
5.12.2.3 phi	109
5.13 PlaneWave1D Class Reference	110
5.13.1 Detailed Description	110
5.13.2 Constructor & Destructor Documentation	111
5.13.2.1 PlaneWave1D()	111
5.13.3 Member Function Documentation	111
5.13.3.1 addToSpace()	112
5.14 PlaneWave2D Class Reference	112
5.14.1 Detailed Description	113
5.14.2 Constructor & Destructor Documentation	113
5.14.2.1 PlaneWave2D()	113
5.14.3 Member Function Documentation	114
5.14.3.1 addToSpace()	114
5.15 PlaneWave3D Class Reference	115
5.15.1 Detailed Description	116
5.15.2 Constructor & Destructor Documentation	116
5.15.2.1 PlaneWave3D()	116
5.15.3 Member Function Documentation	117
5.15.3.1 addToSpace()	117
5.16 Simulation Class Reference	117
5.16.1 Detailed Description	119
5.16.2 Constructor & Destructor Documentation	119
5.16.2.1 Simulation()	120
5.16.2.2 ~Simulation()	120
5.16.3 Member Function Documentation	121
5.16.3.1 addInitialConditions()	121
5.16.3.2 addPeriodicCLayerInX()	122
5.16.3.3 addPeriodicCLayerInXY()	123

5.16.3.4 advanceToTime()	124
5.16.3.5 checkFlag()	125
5.16.3.6 checkNoFlag()	126
5.16.3.7 get_cart_comm()	127
5.16.3.8 initializeCVODEobject()	128
5.16.3.9 initializePatchwork()	129
5.16.3.10 outAllFieldData()	130
5.16.3.11 setDiscreteDimensionsOfLattice()	131
5.16.3.12 setInitialConditions()	132
5.16.3.13 setPhysicalDimensionsOfLattice()	133
5.16.3.14 start()	134
5.16.4 Field Documentation	135
5.16.4.1 cvode_mem	135
5.16.4.2 icsettings	136
5.16.4.3 lattice	136
5.16.4.4 latticePatch	136
5.16.4.5 NLS	136
5.16.4.6 outputManager	137
5.16.4.7 statusFlags	137
5.16.4.8 t	137
5.17 TimeEvolution Class Reference	137
5.17.1 Detailed Description	138
5.17.2 Member Function Documentation	138
5.17.2.1 f()	138
5.17.3 Field Documentation	139
5.17.3.1 c	139
5.17.3.2 TimeEvolver	139
6 File Documentation	141
6.1 README.md File Reference	141
6.2 src/build/CMakeFiles/3.24.2/CompilerIdC/CMakeCCompilerId.c File Reference	141
6.2.1 Macro Definition Documentation	141
6.2.1.1 __has_include	142
6.2.1.2 ARCHITECTURE_ID	142
6.2.1.3 C_VERSION	142
6.2.1.4 COMPILER_ID	142
6.2.1.5 DEC	142
6.2.1.6 HEX	143
6.2.1.7 PLATFORM_ID	143
6.2.1.8 STRINGIFY	143
6.2.1.9 STRINGIFY_HELPER	143
6.2.2 Function Documentation	143

6.2.2.1 <code>main()</code>	144
6.2.3 Variable Documentation	144
6.2.3.1 <code>info_arch</code>	144
6.2.3.2 <code>info_compiler</code>	144
6.2.3.3 <code>info_language_extensions_default</code>	145
6.2.3.4 <code>info_language_standard_default</code>	145
6.2.3.5 <code>info_platform</code>	145
6.3 CMakeCCompilerId.c	145
6.4 src/build/CMakeFiles/3.24.2/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference	155
6.4.1 Macro Definition Documentation	156
6.4.1.1 <code>__has_include</code>	156
6.4.1.2 <code>ARCHITECTURE_ID</code>	156
6.4.1.3 <code>COMPILER_ID</code>	156
6.4.1.4 <code>CXX_STD</code>	156
6.4.1.5 <code>DEC</code>	156
6.4.1.6 <code>HEX</code>	157
6.4.1.7 <code>PLATFORM_ID</code>	157
6.4.1.8 <code>STRINGIFY</code>	157
6.4.1.9 <code>STRINGIFY_HELPER</code>	157
6.4.2 Function Documentation	157
6.4.2.1 <code>main()</code>	158
6.4.3 Variable Documentation	158
6.4.3.1 <code>info_arch</code>	158
6.4.3.2 <code>info_compiler</code>	158
6.4.3.3 <code>info_language_extensions_default</code>	159
6.4.3.4 <code>info_language_standard_default</code>	159
6.4.3.5 <code>info_platform</code>	159
6.5 CMakeCXXCompilerId.cpp	159
6.6 src/build/CMakeFiles/FindMPI/test_mpi.cpp File Reference	169
6.6.1 Function Documentation	169
6.6.1.1 <code>main()</code>	170
6.7 test_mpi.cpp	170
6.8 src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference	170
6.8.1 Function Documentation	171
6.8.1.1 <code>main()</code>	171
6.8.2 Variable Documentation	171
6.8.2.1 <code>ompver_str</code>	172
6.9 OpenMPCheckVersion.c	172
6.10 src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference	172
6.10.1 Function Documentation	173
6.10.1.1 <code>main()</code>	173
6.10.2 Variable Documentation	173

6.10.2.1 <code>ompver_str</code>	173
6.11 <code>OpenMPCheckVersion.cpp</code>	174
6.12 <code>src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c</code> File Reference	174
6.12.1 Function Documentation	174
6.12.1.1 <code>main()</code>	175
6.13 <code>OpenMPTryFlag.c</code>	175
6.14 <code>src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp</code> File Reference	175
6.14.1 Function Documentation	176
6.14.1.1 <code>main()</code>	176
6.15 <code>OpenMPTryFlag.cpp</code>	176
6.16 <code>src/build/CMakeFiles/hewes.dir/DerivationStencils.cpp.o.d</code> File Reference	176
6.17 <code>DerivationStencils.cpp.o.d</code>	176
6.18 <code>src/build/CMakeFiles/hewes.dir/ICSetters.cpp.o.d</code> File Reference	177
6.19 <code>ICSetters.cpp.o.d</code>	177
6.20 <code>src/build/CMakeFiles/hewes.dir/LatticePatch.cpp.o.d</code> File Reference	183
6.21 <code>LatticePatch.cpp.o.d</code>	183
6.22 <code>src/build/CMakeFiles/hewes.dir/main.cpp.o.d</code> File Reference	191
6.23 <code>main.cpp.o.d</code>	191
6.24 <code>src/build/CMakeFiles/hewes.dir/Outputters.cpp.o.d</code> File Reference	198
6.25 <code>Outputters.cpp.o.d</code>	198
6.26 <code>src/build/CMakeFiles/hewes.dir/SimulationClass.cpp.o.d</code> File Reference	205
6.27 <code>SimulationClass.cpp.o.d</code>	205
6.28 <code>src/build/CMakeFiles/hewes.dir/SimulationFunctions.cpp.o.d</code> File Reference	213
6.29 <code>SimulationFunctions.cpp.o.d</code>	213
6.30 <code>src/build/CMakeFiles/hewes.dir/TimeEvolutionFunctions.cpp.o.d</code> File Reference	220
6.31 <code>TimeEvolutionFunctions.cpp.o.d</code>	220
6.32 <code>src/DerivationStencils.cpp</code> File Reference	228
6.32.1 Detailed Description	228
6.33 <code>DerivationStencils.cpp</code>	229
6.34 <code>src/DerivationStencils.h</code> File Reference	229
6.34.1 Detailed Description	231
6.34.2 Function Documentation	231
6.34.2.1 <code>s10b()</code> [1/2]	231
6.34.2.2 <code>s10b()</code> [2/2]	232
6.34.2.3 <code>s10c()</code> [1/2]	232
6.34.2.4 <code>s10c()</code> [2/2]	233
6.34.2.5 <code>s10f()</code> [1/2]	233
6.34.2.6 <code>s10f()</code> [2/2]	234
6.34.2.7 <code>s11b()</code> [1/2]	235
6.34.2.8 <code>s11b()</code> [2/2]	235
6.34.2.9 <code>s11f()</code> [1/2]	236
6.34.2.10 <code>s11f()</code> [2/2]	236

6.34.2.11 s12b() [1/2]	237
6.34.2.12 s12b() [2/2]	238
6.34.2.13 s12c() [1/2]	238
6.34.2.14 s12c() [2/2]	239
6.34.2.15 s12f() [1/2]	239
6.34.2.16 s12f() [2/2]	240
6.34.2.17 s13b() [1/2]	241
6.34.2.18 s13b() [2/2]	241
6.34.2.19 s13f() [1/2]	242
6.34.2.20 s13f() [2/2]	242
6.34.2.21 s1b() [1/2]	243
6.34.2.22 s1b() [2/2]	244
6.34.2.23 s1f() [1/2]	244
6.34.2.24 s1f() [2/2]	245
6.34.2.25 s2b() [1/2]	245
6.34.2.26 s2b() [2/2]	246
6.34.2.27 s2c() [1/2]	246
6.34.2.28 s2c() [2/2]	247
6.34.2.29 s2f() [1/2]	247
6.34.2.30 s2f() [2/2]	248
6.34.2.31 s3b() [1/2]	248
6.34.2.32 s3b() [2/2]	249
6.34.2.33 s3f() [1/2]	249
6.34.2.34 s3f() [2/2]	250
6.34.2.35 s4b() [1/2]	250
6.34.2.36 s4b() [2/2]	251
6.34.2.37 s4c() [1/2]	251
6.34.2.38 s4c() [2/2]	252
6.34.2.39 s4f() [1/2]	252
6.34.2.40 s4f() [2/2]	253
6.34.2.41 s5b() [1/2]	253
6.34.2.42 s5b() [2/2]	254
6.34.2.43 s5f() [1/2]	254
6.34.2.44 s5f() [2/2]	255
6.34.2.45 s6b() [1/2]	255
6.34.2.46 s6b() [2/2]	256
6.34.2.47 s6c() [1/2]	256
6.34.2.48 s6c() [2/2]	257
6.34.2.49 s6f() [1/2]	257
6.34.2.50 s6f() [2/2]	258
6.34.2.51 s7b() [1/2]	258
6.34.2.52 s7b() [2/2]	259

6.34.2.53 s7f() [1/2]	259
6.34.2.54 s7f() [2/2]	260
6.34.2.55 s8b() [1/2]	260
6.34.2.56 s8b() [2/2]	261
6.34.2.57 s8c() [1/2]	261
6.34.2.58 s8c() [2/2]	262
6.34.2.59 s8f() [1/2]	262
6.34.2.60 s8f() [2/2]	263
6.34.2.61 s9b() [1/2]	263
6.34.2.62 s9b() [2/2]	264
6.34.2.63 s9f() [1/2]	264
6.34.2.64 s9f() [2/2]	265
6.35 DerivationStencils.h	265
6.36 src/ICSetters.cpp File Reference	269
6.36.1 Detailed Description	269
6.37 ICSetters.cpp	269
6.38 src/ICSetters.h File Reference	274
6.38.1 Detailed Description	275
6.39 ICSetters.h	275
6.40 src/LatticePatch.cpp File Reference	279
6.40.1 Detailed Description	279
6.40.2 Function Documentation	279
6.40.2.1 check_error()	280
6.40.2.2 check_retval()	280
6.40.2.3 errorKill()	281
6.40.2.4 generatePatchwork()	282
6.41 LatticePatch.cpp	284
6.42 src/LatticePatch.h File Reference	295
6.42.1 Detailed Description	297
6.42.2 Function Documentation	297
6.42.2.1 check_error()	297
6.42.2.2 check_retval()	298
6.42.2.3 errorKill()	299
6.42.3 Variable Documentation	300
6.42.3.1 BuffersInitialized	300
6.42.3.2 FLatticeDimensionSet	300
6.42.3.3 FLatticePatchSetUp	300
6.42.3.4 GhostLayersInitialized	300
6.42.3.5 TranslocationLookupSetUp	301
6.43 LatticePatch.h	301
6.44 src/main.cpp File Reference	304
6.44.1 Detailed Description	304

6.44.2 Function Documentation	304
6.44.2.1 main()	305
6.45 main.cpp	308
6.46 src/Outputters.cpp File Reference	311
6.46.1 Detailed Description	311
6.47 Outputters.cpp	312
6.48 src/Outputters.h File Reference	313
6.48.1 Detailed Description	314
6.49 Outputters.h	315
6.50 src/SimulationClass.cpp File Reference	315
6.50.1 Detailed Description	316
6.51 SimulationClass.cpp	316
6.52 src/SimulationClass.h File Reference	319
6.52.1 Detailed Description	320
6.52.2 Variable Documentation	320
6.52.2.1 CvodeObjectSetUp	321
6.52.2.2 LatticeDiscreteSetUp	321
6.52.2.3 LatticePatchworkSetUp	321
6.52.2.4 LatticePhysicalSetUp	321
6.52.2.5 SimulationStarted	322
6.53 SimulationClass.h	322
6.54 src/SimulationFunctions.cpp File Reference	323
6.54.1 Detailed Description	324
6.54.2 Function Documentation	324
6.54.2.1 Sim1D()	324
6.54.2.2 Sim2D()	326
6.54.2.3 Sim3D()	329
6.54.2.4 timer()	331
6.55 SimulationFunctions.cpp	332
6.56 src/SimulationFunctions.h File Reference	335
6.56.1 Detailed Description	337
6.56.2 Function Documentation	337
6.56.2.1 Sim1D()	337
6.56.2.2 Sim2D()	339
6.56.2.3 Sim3D()	342
6.56.2.4 timer()	344
6.57 SimulationFunctions.h	345
6.58 src/TimeEvolutionFunctions.cpp File Reference	346
6.58.1 Detailed Description	347
6.58.2 Function Documentation	347
6.58.2.1 linear1DProp()	347
6.58.2.2 linear2DProp()	349

6.58.2.3 linear3DProp()	351
6.58.2.4 nonlinear1DProp()	353
6.58.2.5 nonlinear2DProp()	358
6.58.2.6 nonlinear3DProp()	362
6.59 TimeEvolutionFunctions.cpp	366
6.60 src/TimeEvolutionFunctions.h File Reference	374
6.60.1 Detailed Description	376
6.60.2 Function Documentation	376
6.60.2.1 linear1DProp()	376
6.60.2.2 linear2DProp()	377
6.60.2.3 linear3DProp()	379
6.60.2.4 nonlinear1DProp()	381
6.60.2.5 nonlinear2DProp()	386
6.60.2.6 nonlinear3DProp()	390
6.61 TimeEvolutionFunctions.h	394
Index	395

Chapter 1

HEWES: Heisenberg-Euler Weak-Field Expansion Simulator

The Heisenberg-Euler Weak-Field Expansion Simulator is a solver for the all-optical QED vacuum. It solves the equations of motion for electromagnetic waves in the Heisenberg-Euler effective QED theory in the weak-field expansion with up to six-photon processes.

There is a [paper](#) that introduces the algorithm and shows remarkable results and a [Mendeley Data repository](#) with extra and supplementary materials.

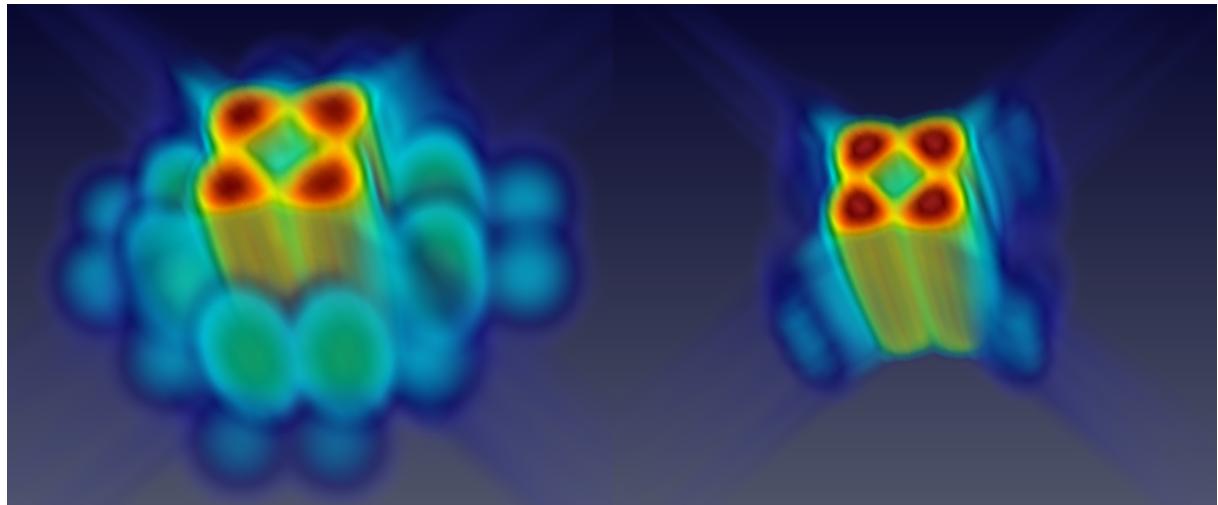


Figure 1.1 Harmonic Generation 3D

1.1 Contents

- Required software
- Short user manual
 - Note on simulation settings
 - Note on resource occupation
 - Note on the output analysis
- Authors

1.2 Required software

CMake is used for building and a recent C++ compiler version is required since features up to the C++20 standard are used. An *MPI* implementation is required. *OpenMP* is optional to enforce more vectorization and enable multi-threading. The latter is useful for performance only when a very large number of compute nodes is used.

The *CVODE* solver is installed on the fly through *CMake*. If *CVODE* (or the whole *SUNDIALS* package) is installed manually: Version 6 is required, the code is presumably compliant with the upcoming version 7. Enable *MPI* (and *OpenMP*). For optimal performance the `CMAKE_BUILD_TYPE` should be "Release". Edit the `SUNDIALS_DIR` in the `CMakeLists.txt`.

1.3 Short user manual

You have full control over all high-level simulation settings via the `main.cpp` file. (In an upcoming version the parameters will be passed via the command line at the detriment of missing compile-time checks.)

- First, specify the path you want the output data to go via the variable `outputDirectory`.
- Second, decide if you want to simulate in 1D, 2D, or 3D and uncomment only that full section.
You can then specify
 - the relative and absolute integration tolerances of the *CVODE* solver.
Recommended values are between 1e-12 and 1e-16.
 - the order of accuracy of the numerical scheme via the stencil order.
You can choose an integer in the range 1-13.
 - the physical side lengths of the grid in meters.
 - the number of lattice points per dimension.
 - the slicing of the lattice into patches (only for 2D and 3D simulations, automatic in 1D) – this determines the number of patches and therefore the required distinct processing units for *MPI*.
The total number of processes is given by the product of patches in any dimension.
Note: In the 3D case patches are required to be cubic in terms of lattice points. This is decisive for computational efficiency and checked at compile-time.
 - whether to have periodic or vanishing boundary values (currently has to be chosen periodic).
 - whether you want to simulate on top of the linear vacuum only 4-photon processes (1), 6-photon processes (2), both (3), or none (0) – the linear Maxwell case.
 - the total time of the simulation in units $c=1$, i.e., the distance propagated by the light waves in meters.
 - the number of time steps that will be solved stepwise by *CVODE*.
In order to keep interpolation errors small do not choose this number too small.
 - the multiple of steps at which you want the data to be written to disk.
 - the output format. It can be 'c' for comma separated values (csv), or 'b' for binary. For csv format the name of the files written to the output directory is of the form `{step_number}_{process_number}.csv`. For binary output all data per step are written into one file and the step number is the name of the file.
 - which electromagnetic waveform(s) you want to propagate.
You can choose between a plane wave (not much physical content, but useful for checks) and implementations of Gaussians in 1D, 2D, and 3D. Their parameters can be tuned.
A description of the wave implementations is given in [ref.pdf](#). Note that the 3D Gaussians, as they are implemented up to now, should be propagated in the xy-plane. More waveform implementations will follow in subsequent versions of the code.

A doxygen-generated complete code reference is provided with [ref.pdf](#).

- Third, in the src directory execute `cmake -S . -Bbuild` and then `cmake --build build`.
- Forth, run the simulation.

Make sure to use `src` as working directory as the code uses a relative path to log the configuration in `main.cpp`.

Note that in 2D and 3D simulations the number of *MPI* processes has to coincide with the actual number of patches, as described above.

Here, the simulation would be executed distributed over four processes:

```
mpirun -np 4 ./build/hewes
```

When the program was built with *OpenMP*, the environment variable `OMP_NUM_THREADS` needs to be set.

- Monitor `stdout` and `stderr`. The unique simulation identifier number (starting timestep = name of data directory), the process steps, and the used wall times per step are printed on `stdout`. Errors are printed on `stderr`.

Note: Convergence of the employed *CVODE* solver cannot be guaranteed and issues of this kind can hardly be predicted. On top, they are even system dependent. Piece of advice: Only pass decimal numbers for the grid settings and initial conditions.

CVODE warnings and errors are reported on `stdout` and `stderr`.

A `config.txt` file containing the configuration part of `main.cpp` is written to the output directory in order to log the simulation settings of each particular run.

1.3.1 Note on simulation settings

You may want to start with two Gaussian pulses in 1D colliding head-on in a pump-probe setup. For this event, specify a high-frequency probe pulse with a low amplitude and a low-frequency pump pulse with a high frequency. Both frequencies should be chosen to be below a forth of the Nyquist frequency to minimize nonphysical dispersion effects on the lattice. The wavelengths should neither be chosen too large (bulky wave) on a fine patchwork of narrow patches. Their communication might be problematic with too small halo layer depths. You would observe a blurring over time. The amplitudes need be below 1 – the critical field strength – for the weak-field expansion to be valid.

You can then investigate the arising of higher harmonics in frequency space via a Fourier analysis. The signals from the higher harmonics can be highlighted by subtracting the results of the same simulation in the linear Maxwell vacuum. You will be left with the nonlinear effects.

Choosing the probe pulse to be polarized with an angle to the polarization of the pump you may observe a fractional polarization flip of the probe due to their nonlinear interaction.

Decide beforehand which steps you need to be written to disk for your analysis.

Example scenarios of colliding Gaussians are preconfigured for any dimension.

1.3.2 Note on resource occupation

The computational load depends mostly on the grid size and resolution. The order of accuracy of the numerical scheme and *CVODE* are rather secondary except for simulations running on many processing units, as the communication load is dependent on the stencil order.

Simulations in 1D are relatively cheap and can easily be run on a modern laptop within seconds. The output size per step is usually less than a megabyte.

Simulations in 2D with about one million grid points are still feasible for a personal machine but might take a couple of minutes or longer to finish. The output size per step is in the range of some dozen megabytes.

Sensible simulations in 3D require large memory resources and therefore need to be run on distributed systems. Hundreds of cores can be kept busy for many hours or days. The output size quickly amounts to dozens of gigabytes for just a single state.

1.3.3 Note on the output analysis

The field data are either written in csv format to one file per MPI process, the ending of which (after an underscore) corresponds to the process number, as described above. This is the simplest solution for smaller simulations and a portable way that also works fast and is straightforward to analyze.

Or, the option strictly recommended for larger write operations, in binary format with a single file per output step. Raw bytes are written to the files as they are in memory. This option is more performant and achieved with MPI IO. However, there is no guarantee of portability; postprocessing/conversion is required. The step number is the file name.

A `SimResults` folder is created in the chosen output directory if it does not exist and therein a folder named after the starting timestep of the simulation (in the form `yy-mm-dd_hh-MM-ss`) is created. This is where the output files are written into.

There are six columns in the csv files, corresponding to the six components of the electromagnetic field: `E_x, E_y, E_z, B_x, B_y, B_z. Each row corresponds to one lattice point.`

Postprocessing is required to read-in the files in order. A [Python module](#) taking care of this is provided.

Likewise, [another Python module](#) is provided to read the binary data of a selected field component into a numpy array – its portability, however, cannot be guaranteed.

The process numbers first align along dimension 1 until the number of patches is that direction is reached, then continue on dimension two and finally fill dimension 3. For example, for a 3D simulation on 4x4x4=64 cores, the field data is divided over the patches as follows:

<code>z=1</code>	<code>z=2</code>	<code>z=3</code>	<code>z=4</code>
<code>x</code>	<code>x</code>	<code>x</code>	<code>x</code>
<code>^</code>	<code>^</code>	<code>^</code>	<code>^</code>
1 0 4 8 12	1 16 20 24 28	1 16 20 24 28	1 16 20 24 28
2 1 5 9 13	2 17 21 25 29	2 17 21 25 29	2 17 21 25 29
3 2 6 10 14	3 18 22 26 30	3 18 22 26 30	3 18 22 26 30
4 3 7 11 15	4 19 23 27 31	4 19 23 27 31	4 19 23 27 31
----->	----->	----->	----->
1 2 3 4 y	1 2 3 4 y	1 2 3 4 y	1 2 3 4 y

The axes denote the physical dimensions that are each divided into 4 sectors in this example. The numbers inside the 4x4 squares indicate the process number, which is the number of the patch and also the number at the end of the corresponding output csv file. The ordering of the array within a patch follows the standard C convention and can be reshaped in 2D and 3D to the actual size of the path.

More information describing settings and analysis procedures used for actual scientific results are given in an open-access [paper](#) and a collection of corresponding analysis notebooks are uploaded to a [Mendeley Data repository](#). Some example Python analysis scripts can be found in the examples. The [first steps](#) demonstrate how the simulated data is accurately read-in from disk to numpy arrays using the provided [get field data module](#). [Harmonic generation](#) in various forms is sketched as one application showing nonlinear quantum vacuum effects. Analyses of 3D simulations are more involved due to large volumes of data. Visualization requires tools like Paraview; examples are shown [here](#). There is however *no simulation data provided* as it would make the repository size unnecessarily large.

1.4 Authors

- Arnau Pons Domenech
- Hartmut Ruhl (hartmut.ruhl@physik.uni-muenchen.de)
- Andreas Lindner (and.lindner@physik.uni-muenchen.de)
- Baris Ölmez (b.oelmez@physik.uni-muenchen.de)

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Gauss1D	11
Gauss2D	17
Gauss3D	23
gaussian1D	28
gaussian2D	30
gaussian3D	32
ICSetter	34
Lattice	43
LatticePatch	58
OutputManager	97
PlaneWave	105
PlaneWave1D	110
PlaneWave2D	112
PlaneWave3D	115
planewave	108
Simulation	117
TimeEvolution	137

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Gauss1D	Class for Gaussian pulses in 1D	11
Gauss2D	Class for Gaussian pulses in 2D	17
Gauss3D	Class for Gaussian pulses in 3D	23
gaussian1D	1D Gaussian wave structure	28
gaussian2D	2D Gaussian wave structure	30
gaussian3D	3D Gaussian wave structure	32
ICSetter	ICSetter class to initialize wave types with default parameters	34
Lattice	Lattice class for the construction of the enveloping discrete simulation space	43
LatticePatch	LatticePatch class for the construction of the patches in the enveloping lattice	58
OutputManager	Output Manager class to generate and coordinate output writing to disk	97
PlaneWave	Super-class for plane waves	105
planewave	Plane wave structure	108
PlaneWave1D	Class for plane waves in 1D	110
PlaneWave2D	Class for plane waves in 2D	112
PlaneWave3D	Class for plane waves in 3D	115
Simulation	Simulation class to instantiate the whole walkthrough of a Simulation	117
TimeEvolution	Monostate TimeEvolution class to propagate the field data in time in a given order of the HE weak-field expansion	137

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

src/DerivationStencils.cpp	
Empty. All definitions in the header	228
src/DerivationStencils.h	
Definition of derivation stencils from order 1 to 13	229
src/ICSetters.cpp	
Implementation of the plane wave and Gaussian wave packets	269
src/ICSetters.h	
Declaration of the plane wave and Gaussian wave packets	274
src/LatticePatch.cpp	
Construction of the overall envelope lattice and the lattice patches	279
src/LatticePatch.h	
Declaration of the lattice and lattice patches	295
src/main.cpp	
Main function to configure the user's simulation settings	304
src/Outputters.cpp	
Generation of output writing to disk	311
src/Outputters.h	
OutputManager class to outstream simulation data	313
src/SimulationClass.cpp	
Interface to the whole Simulation procedure: from wave settings over lattice construction, time evolution and outputs (also all relevant CVODE steps are performed here)	315
src/SimulationClass.h	
Class for the Simulation object calling all functionality: from wave settings over lattice construction, time evolution and outputs initialization of the CVode object	319
src/SimulationFunctions.cpp	
Implementation of the complete simulation functions for 1D, 2D, and 3D, as called in the main function	323
src/SimulationFunctions.h	
Full simulation functions for 1D, 2D, and 3D used in main.cpp	335
src/TimeEvolutionFunctions.cpp	
Implementation of functions to propagate data vectors in time according to Maxwell's equations, and various orders in the HE weak-field expansion	346
src/TimeEvolutionFunctions.h	
Functions to propagate data vectors in time according to Maxwell's equations, and various orders in the HE weak-field expansion	374

src/build/CMakeFiles/3.24.2/CompilerIdC/CMakeCCCompilerId.c	141
src/build/CMakeFiles/3.24.2/CompilerIdCXX/CMakeCXXCompilerId.cpp	155
src/build/CMakeFiles/FindMPI/test_mpi.cpp	169
src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c	170
src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp	172
src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c	174
src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp	175
src/build/CMakeFiles/hewes.dir/DerivationStencils.cpp.o.d	176
src/build/CMakeFiles/hewes.dir/ICSetters.cpp.o.d	177
src/build/CMakeFiles/hewes.dir/LatticePatch.cpp.o.d	183
src/build/CMakeFiles/hewes.dir/main.cpp.o.d	191
src/build/CMakeFiles/hewes.dir/Outputters.cpp.o.d	198
src/build/CMakeFiles/hewes.dir/SimulationClass.cpp.o.d	205
src/build/CMakeFiles/hewes.dir/SimulationFunctions.cpp.o.d	213
src/build/CMakeFiles/hewes.dir/TimeEvolutionFunctions.cpp.o.d	220

Chapter 5

Data Structure Documentation

5.1 Gauss1D Class Reference

class for Gaussian pulses in 1D

```
#include <src/ICSetters.h>
```

Public Member Functions

- `Gauss1D` (std::array< surrealtype, 3 > k={1, 0, 0}, std::array< surrealtype, 3 > p={0, 0, 1}, std::array< surrealtype, 3 > xo={0, 0, 0}, surrealtype phig_=1.0, std::array< surrealtype, 3 > phi={0, 0, 0})
construction with default parameters
- void `addToSpace` (surrealtype x, surrealtype y, surrealtype z, surrealtype *pTo6Space) const
function for the actual implementation in space

Private Attributes

- surrealtype `kx`
wavenumber k_x
- surrealtype `ky`
wavenumber k_y
- surrealtype `kz`
wavenumber k_z
- surrealtype `px`
polarization & amplitude in x-direction, p_x
- surrealtype `py`
polarization & amplitude in y-direction, p_y
- surrealtype `pz`
polarization & amplitude in z-direction, p_z
- surrealtype `phix`
phase shift in x-direction, ϕ_x
- surrealtype `phiy`
phase shift in y-direction, ϕ_y
- surrealtype `phiz`

- sunrealtype `x0z`
center of pulse in z-direction, ϕ_z
- sunrealtype `x0x`
center of pulse in x-direction, x_0
- sunrealtype `x0y`
center of pulse in y-direction, y_0
- sunrealtype `x0z`
center of pulse in z-direction, z_0
- sunrealtype `phig`
pulse width Φ_g

5.1.1 Detailed Description

class for Gaussian pulses in 1D

They are given in the form $\vec{E} = \vec{p} \exp\left(-(\vec{x} - \vec{x}_0)^2/\Phi_g^2\right) \cos(\vec{k} \cdot \vec{x})$

Definition at line 83 of file [ICSetters.h](#).

5.1.2 Constructor & Destructor Documentation

5.1.2.1 Gauss1D()

```
Gauss1D::Gauss1D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > xo = {0, 0, 0},
    sunrealtype phig_ = 1.0,
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
```

construction with default parameters

[Gauss1D](#) construction with

- wavevectors k_x
- k_y
- k_z normalized to $1/\lambda$
- amplitude (polarization) in x-direction
- amplitude (polarization) in y-direction
- amplitude (polarization) in z-direction
- phase shift in x-direction
- phase shift in y-direction
- phase shift in z-direction
- width

- shift from origin in x-direction
- shift from origin in y-direction
- shift from origin in z-direction

Definition at line 125 of file [ICSetters.cpp](#).

```
00127
00128   kx = k[0];    /** - wavevectors \f$ k_x \f$ */
00129   ky = k[1];    /** - \f$ k_y \f$ */
00130   kz = k[2];    /** - \f$ k_z \f$ normalized to \f$ 1/\lambda \f$ */
00131   px = p[0];    /** - amplitude (polarization) in x-direction */
00132   py = p[1];    /** - amplitude (polarization) in y-direction */
00133   pz = p[2];    /** - amplitude (polarization) in z-direction */
00134   phix = phi[0]; /* - phase shift in x-direction */
00135   phiy = phi[1]; /* - phase shift in y-direction */
00136   phiz = phi[2]; /* - phase shift in z-direction */
00137   phig = phig_; /* - width */
00138   x0x = xo[0];  /* - shift from origin in x-direction*/
00139   x0y = xo[1];  /* - shift from origin in y-direction*/
00140   x0z = xo[2];  /* - shift from origin in z-direction*/
00141 }
```

References [kx](#), [ky](#), [kz](#), [phig](#), [phix](#), [phiy](#), [phiz](#), [px](#), [py](#), [pz](#), [x0x](#), [x0y](#), and [x0z](#).

5.1.3 Member Function Documentation

5.1.3.1 addToSpace()

```
void Gauss1D::addToSpace (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space ) const
```

function for the actual implementation in space

[Gauss1D](#) implementation in space

Definition at line 144 of file [ICSetters.cpp](#).

```
00145
00146   const sunrealtype wavelength =
00147     sqrt(kx * kx + ky * ky + kz * kz); /* \f$ 1/\lambda \f$ */
00148   x = x - x0x; /* x-coordinate minus shift from origin */
00149   y = y - x0y; /* y-coordinate minus shift from origin */
00150   z = z - x0z; /* z-coordinate minus shift from origin */
00151   const sunrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 *
00152     std::numbers::pi; /* \f$ 2\pi \f$ */
00153   const sunrealtype envelopeAmp =
00154     exp(-(x * x + y * y + z * z) / phig / phig); /* enveloping Gauss shape */
00155   // Gaussian wave definition
00156   const std::array<sunrealtype, 3> E{                                /* E-field vector */
00157     px * cos(kScalarX - phix) * envelopeAmp, /* \f$ E_x \f$ */
00158     py * cos(kScalarX - phiy) * envelopeAmp, /* \f$ E_y \f$ */
00159     pz * cos(kScalarX - phiz) * envelopeAmp}; /* \f$ E_z \f$ */
00160   // Put E-field into space
00161   pTo6Space[0] += E[0];
00162   pTo6Space[1] += E[1];
00163   pTo6Space[2] += E[2];
00164   // and B-field
00165   pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
00166   pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00167   pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00168
00169 }
```

References [kx](#), [ky](#), [kz](#), [phig](#), [phix](#), [phiy](#), [phiz](#), [px](#), [py](#), [pz](#), [x0x](#), [x0y](#), and [x0z](#).

5.1.4 Field Documentation

5.1.4.1 kx

```
sunrealtype Gauss1D::kx [private]
```

wavenumber k_x

Definition at line 86 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.2 ky

```
sunrealtype Gauss1D::ky [private]
```

wavenumber k_y

Definition at line 88 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.3 kz

```
sunrealtype Gauss1D::kz [private]
```

wavenumber k_z

Definition at line 90 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.4 phig

```
sunrealtype Gauss1D::phig [private]
```

pulse width Φ_g

Definition at line 110 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.5 phix

```
sunrealtype Gauss1D::phix [private]
```

phase shift in x-direction, ϕ_x

Definition at line 98 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.6 phiy

```
sunrealtype Gauss1D::phiy [private]
```

phase shift in y-direction, ϕ_y

Definition at line 100 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.7 phiz

```
sunrealtype Gauss1D::phiz [private]
```

phase shift in z-direction, ϕ_z

Definition at line 102 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.8 px

```
sunrealtype Gauss1D::px [private]
```

polarization & amplitude in x-direction, p_x

Definition at line 92 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.9 **py**

`sunrealtype Gauss1D::py [private]`

polarization & amplitude in y-direction, p_y

Definition at line 94 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.10 **pz**

`sunrealtype Gauss1D::pz [private]`

polarization & amplitude in z-direction, p_z

Definition at line 96 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.11 **x0x**

`sunrealtype Gauss1D::x0x [private]`

center of pulse in x-direction, x_0

Definition at line 104 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.12 **x0y**

`sunrealtype Gauss1D::x0y [private]`

center of pulse in y-direction, y_0

Definition at line 106 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

5.1.4.13 x0z

`sunrealtype Gauss1D::x0z [private]`

center of pulse in z-direction, z_0

Definition at line 108 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss1D\(\)](#).

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

- [src/ICSetters.h](#)
- [src/ICSetters.cpp](#)

5.2 Gauss2D Class Reference

class for Gaussian pulses in 2D

```
#include <src/ICSetters.h>
```

Public Member Functions

- [Gauss2D \(std::array< unrealtype, 3 > dis_= {0, 0, 0}, std::array< unrealtype, 3 > axis_= {1, 0, 0}, unrealtype Amp_= 1.0, unrealtype phip_= 0, unrealtype w0_= 1e-5, unrealtype zr_= 4e-5, unrealtype Ph0_= 2e-5, unrealtype PhA_= 0.45e-5\)](#)
construction with default parameters
- void [addToSpace \(unrealtype x, unrealtype y, unrealtype z, unrealtype *pTo6Space\) const](#)
function for the actual implementation in space

Private Attributes

- `std::array< unrealtype, 3 > dis`
distance maximum to origin
- `std::array< unrealtype, 3 > axis`
normalized propagation axis
- `unrealtype Amp`
amplitude A
- `unrealtype phip`
polarization rotation from TE-mode around propagation direction
- `unrealtype w0`
taille ω_0
- `unrealtype zr`
Rayleigh length $z_R = \pi\omega_0^2/\lambda$
- `unrealtype Ph0`
center of beam Φ_0
- `unrealtype PhA`
length of beam Φ_A
- `unrealtype A1`
amplitude projection on TE-mode
- `unrealtype A2`
amplitude projection on xy-plane
- `unrealtype lambda`
wavelength λ

5.2.1 Detailed Description

class for Gaussian pulses in 2D

They are given in the form $\vec{E} = A \vec{\epsilon} \sqrt{\frac{\omega_0}{\omega(z)}} \exp(-r/\omega(z))^2 \exp\left(-((z_g - \Phi_0)/\Phi_A)^2\right) \cos\left(\frac{k r^2}{2R(z)} + g(z) - k z_g\right)$ with

- propagation direction (subtracted distance to origin) z_g
- radial distance to propagation axis $r = \sqrt{\vec{x}^2 - z_g^2}$
- $k = 2\pi/\lambda$
- waist at position z , $\omega(z) = w_0 \sqrt{1 + (z_g/z_R)^2}$
- Gouy phase $g(z) = \tan^{-1}(z_g/z_r)$
- beam curvature $R(z) = z_g (1 + (z_r/z_g)^2)$ obtained via the chosen parameters

Definition at line 139 of file [ICSetters.h](#).

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Gauss2D()

```
Gauss2D::Gauss2D (
    std::array< sunrealtype, 3 > dis_ = {0, 0, 0},
    std::array< sunrealtype, 3 > axis_ = {1, 0, 0},
    sunrealtype Amp_ = 1.0,
    sunrealtype phip_ = 0,
    sunrealtype w0_ = 1e-5,
    sunrealtype zr_ = 4e-5,
    sunrealtype Ph0_ = 2e-5,
    sunrealtype PhA_ = 0.45e-5 )
```

construction with default parameters

[Gauss2D](#) construction with

- center it approaches
- direction from where it comes
- amplitude
- polarization rotation from TE-mode
- taille
- Rayleigh length
- beam center
- beam length

Definition at line 172 of file [ICSetters.cpp](#).

```

00175
00176     dis = dis_;           /** - center it approaches */
00177     axis = axis_;         /** - direction form where it comes */
00178     Amp = Amp_;          /** - amplitude */
00179     phip = phip_;        /** - polarization rotation from TE-mode */
00180     w0 = w0_;            /** - taille */
00181     zr = zr_;            /** - Rayleigh length */
00182     Ph0 = Ph0_;          /** - beam center */
00183     PhA = PhA_;          /** - beam length */
00184     A1 = Amp * cos(phip); // amplitude in z-direction
00185     A2 = Amp * sin(phip); // amplitude on xy-plane
00186     lambda = std::numbers::pi * w0 * w0 / zr; // formula for wavelength
00187 }
```

References [A1](#), [A2](#), [Amp](#), [axis](#), [dis](#), [lambda](#), [Ph0](#), [PhA](#), [phip](#), [w0](#), and [zr](#).

5.2.3 Member Function Documentation

5.2.3.1 addToSpace()

```

void Gauss2D::addToSpace (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space ) const
```

function for the actual implementation in space

Definition at line 189 of file [ICSetters.cpp](#).

```

00190
00191     //\f$ \vec{x} = \vec{x}_0-\vec{dis} \f$ // coordinates minus distance to
00192     //origin
00193     x -= dis[0];
00194     y -= dis[1];
00195     // z-=dis[2];
00196     z = nan("0x12345"); // unused parameter
00197     // \f$ z_g = \vec{x}\cdot\vec{e}_g \f$ projection on propagation axis
00198     const sunrealtype zg =
00199         x * axis[0] + y * axis[1]; // +z*axis[2]; // =z-z0 -> propagation
00200         //direction, minus origin
00201     // \f$ r = \sqrt{\vec{x}^2 - z_g^2} \f$ -> pythagoras of radius minus
00202     // projection on prop axis
00203     const sunrealtype r = sqrt((x * x + y * y) /*+z*z*/)
00204         zg * zg); // radial distance to propagation axis
00205     // \f$ w(z) = w_0\sqrt{1+(z_g/z_R)^2} \f$-
00206     // waist at position z
00207     const sunrealtype wz = w0 * sqrt(1 + (zg * zg / zr / zr));
00208     // \f$ g(z) = \arctan(z_g/z_r) \f$-
00209     const sunrealtype gz = atan(zg / zr); // Gouy phase
00210     // \f$ R(z) = z_g*(1+(z_r/z_g)^2) \f$-
00211     sunrealtype Rz = nan("0x12345"); // beam curvature
00212     if (abs(gz) > 1e-15)
00213         Rz = zg * (1 + (zr * zr / zg / zg));
00214     else
00215         Rz = 1e308;
00216     // wavenumber \f$ k = 2\pi/\lambda \f$-
00217     const sunrealtype k = 2 * std::numbers::pi / lambda;
00218     // \f$ \Phi_F = kr^2/(2*R(z))+g(z)-kz_g \f$-
00219     const sunrealtype PhF =
00220         -k * r * r / (2 * Rz) + gz - k * zg; // to be inserted into cosine
00221     // \f$ G = \sqrt{w_0/w_z}e^{-(r/w(z))^2}e^{-(zg-\Phi_0)^2/PhA^2}\cos(\Phi_F) \f$-
00222     // CVode is a diva, no chance to remove the square in the second exponential
00223     // -> h too small
00224     const sunrealtype G2D = sqrt(w0 / wz) * exp(-r * r / wz / wz) *
00225         exp(-(zg - Ph0) * (zg - Ph0) / PhA / PhA) *
00226         cos(PhF); // gauss shape
00227     // \f$ c_\alpha = \vec{e}_x\cdot\vec{e}_z \f$-
00228     // projection components; do like this for CVode convergence -> otherwise
00229     // results in machine error values for non-existant field components if
00230     // axis[0] and axis[1] are given
```

```

00231 const sunrealtype ca =
00232     axis[0]; // x-component of propagation axis which is given as parameter
00233 // no z-component for 2D propagation
00234 const sunrealtype sa = sqrt(1 - ca * ca);
00235 // E-field to space: polarization in xy-plane (A2) is projection of
00236 // z-polarization (A1) on x- and y-directions
00237 pTo6Space[0] += sa * (G2D * A2);
00238 pTo6Space[1] += -ca * (G2D * A2);
00239 pTo6Space[2] += G2D * A1;
00240 // B-field -> negative derivative wrt polarization shift of E-field
00241 pTo6Space[3] += -sa * (G2D * A1);
00242 pTo6Space[4] += ca * (G2D * A1);
00243 pTo6Space[5] += G2D * A2;
00244 }

```

References [A1](#), [A2](#), [axis](#), [dis](#), [lambda](#), [Ph0](#), [PhA](#), [w0](#), and [zr](#).

5.2.4 Field Documentation

5.2.4.1 A1

`sunrealtype Gauss2D::A1 [private]`

amplitude projection on TE-mode

Definition at line [159](#) of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.2 A2

`sunrealtype Gauss2D::A2 [private]`

amplitude projection on xy-plane

Definition at line [161](#) of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.3 Amp

`sunrealtype Gauss2D::Amp [private]`

amplitude A

Definition at line [146](#) of file [ICSetters.h](#).

Referenced by [Gauss2D\(\)](#).

5.2.4.4 axis

```
std::array<sunrealtype, 3> Gauss2D::axis [private]
```

normalized propagation axis

Definition at line 144 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.5 dis

```
std::array<sunrealtype, 3> Gauss2D::dis [private]
```

distance maximum to origin

Definition at line 142 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.6 lambda

```
sunrealtype Gauss2D::lambda [private]
```

wavelength λ

Definition at line 163 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.7 Ph0

```
sunrealtype Gauss2D::Ph0 [private]
```

center of beam Φ_0

Definition at line 155 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.8 PhA

`sunrealtype Gauss2D::PhA [private]`

length of beam Φ_A

Definition at line 157 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.9 phip

`sunrealtype Gauss2D::phip [private]`

polarization rotation from TE-mode around propagation direction

Definition at line 149 of file [ICSetters.h](#).

Referenced by [Gauss2D\(\)](#).

5.2.4.10 w0

`sunrealtype Gauss2D::w0 [private]`

taille ω_0

Definition at line 151 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

5.2.4.11 zr

`sunrealtype Gauss2D::zr [private]`

Rayleigh length $z_R = \pi\omega_0^2/\lambda$.

Definition at line 153 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss2D\(\)](#).

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

- [src/ICSetters.h](#)
- [src/ICSetters.cpp](#)

5.3 Gauss3D Class Reference

class for Gaussian pulses in 3D

```
#include <src/ICSetters.h>
```

Public Member Functions

- `Gauss3D` (`std::array<sunrealtype, 3> dis_={0, 0, 0}, std::array<sunrealtype, 3> axis_={1, 0, 0}, unrealtype Amp_=1.0, unrealtype phip_=0, unrealtype w0_=1e-5, unrealtype zr_=4e-5, unrealtype Ph0_=2e-5, unrealtype PhA_=0.45e-5)`
construction with default parameters
- void `addToSpace` (`unrealtype x, unrealtype y, unrealtype z, unrealtype *pTo6Space`) const
function for the actual implementation in space

Private Attributes

- `std::array<unrealtype, 3> dis`
distance maximum to origin
- `std::array<unrealtype, 3> axis`
normalized propagation axis
- `unrealtype Amp`
amplitude A
- `unrealtype phip`
polarization rotation from TE-mode around propagation direction
- `unrealtype w0`
taille ω_0
- `unrealtype zr`
Rayleigh length $z_R = \pi\omega_0^2/\lambda$.
- `unrealtype Ph0`
center of beam Φ_0
- `unrealtype PhA`
length of beam Φ_A
- `unrealtype A1`
amplitude projection on TE-mode (z-axis)
- `unrealtype A2`
amplitude projection on xy-plane
- `unrealtype lambda`
wavelength λ

5.3.1 Detailed Description

class for Gaussian pulses in 3D

They are given in the form $\vec{E} = A \vec{\epsilon} \frac{\omega_0}{\omega(z)} \exp(-r/\omega(z))^2 \exp\left(-((z_g - \Phi_0)/\Phi_A)^2\right) \cos\left(\frac{k r^2}{2R(z)} + g(z) - k z_g\right)$ with

- propagation direction (subtracted distance to origin) z_g
- radial distance to propagation axis $r = \sqrt{\vec{x}^2 - z_g^2}$
- $k = 2\pi/\lambda$
- waist at position z , $\omega(z) = w_0 \sqrt{1 + (z_g/z_R)^2}$
- Gouy phase $g(z) = \tan^{-1}(z_g/z_r)$
- beam curvature $R(z) = z_g (1 + (z_r/z_g)^2)$ obtained via the chosen parameters

Definition at line 193 of file [ICSetters.h](#).

5.3.2 Constructor & Destructor Documentation

5.3.2.1 Gauss3D()

```
Gauss3D::Gauss3D (
    std::array< sunrealtype, 3 > dis_ = {0, 0, 0},
    std::array< sunrealtype, 3 > axis_ = {1, 0, 0},
    sunrealtype Amp_ = 1.0,
    sunrealtype phip_ = 0,
    sunrealtype w0_ = 1e-5,
    sunrealtype zr_ = 4e-5,
    sunrealtype Ph0_ = 2e-5,
    sunrealtype PhA_ = 0.45e-5 )
```

construction with default parameters

[Gauss3D](#) construction with

- center it approaches
- direction from where it comes
- amplitude
- polarization rotation form TE-mode
- taille
- Rayleigh length
- beam center
- beam length

Definition at line 247 of file [ICSetters.cpp](#).

```

00252                                     {
00253     dis = dis_;    /** - center it approaches */
00254     axis = axis_; /** - direction from where it comes */
00255     Amp = Amp_;   /** - amplitude */
00256     // pol=pol_;
00257     phip = phip_; /** - polarization rotation form TE-mode */
00258     w0 = w0_;      /** - taille */
00259     zr = zr_;       /** - Rayleigh length */
00260     Ph0 = Ph0_;    /** - beam center */
00261     PhA = PhA_;   /** - beam length */
00262     lambda = std::numbers::pi * w0 * w0 / zr;
00263     A1 = Amp * cos(phip);
00264     A2 = Amp * sin(phip);
00265 }
```

References [A1](#), [A2](#), [Amp](#), [axis](#), [dis](#), [lambda](#), [Ph0](#), [PhA](#), [phip](#), [w0](#), and [zr](#).

5.3.3 Member Function Documentation

5.3.3.1 addToSpace()

```

void Gauss3D::addToSpace (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space ) const
```

function for the actual implementation in space

[Gauss3D](#) implementation in space

Definition at line 268 of file [ICSetters.cpp](#).

```

00269                                     {
00270     x -= dis[0];
00271     y -= dis[1];
00272     z -= dis[2];
00273     const sunrealtype zg = x * axis[0] + y * axis[1] + z * axis[2];
00274     const sunrealtype r = sqrt((x * x + y * y + z * z) - zg * zg);
00275     const sunrealtype wz = w0 * sqrt(1 + (zg * zg / zr / zr));
00276     const sunrealtype gz = atan(zg / zr);
00277     sunrealtype Rz = nan("0x12345");
00278     if (abs(zg) > 1e-15)
00279         Rz = zg * (1 + (zr * zr / zg / zg));
00280     else
00281         Rz = 1e308;
00282     const sunrealtype k = 2 * std::numbers::pi / lambda;
00283     const sunrealtype PhF = -k * r * r / (2 * Rz) + gz - k * zg;
00284     const sunrealtype G3D = (w0 / wz) * exp(-r * r / wz / wz) *
00285         exp(-(zg - Ph0) * (zg - Ph0) / PhA / PhA) * cos(PhF);
00286     const sunrealtype ca = axis[0];
00287     const sunrealtype sa = sqrt(1 - ca * ca);
00288     pTo6Space[0] += sa * (G3D * A2);
00289     pTo6Space[1] += -ca * (G3D * A2);
00290     pTo6Space[2] += G3D * A1;
00291     pTo6Space[3] += -sa * (G3D * A1);
00292     pTo6Space[4] += ca * (G3D * A1);
00293     pTo6Space[5] += G3D * A2;
00294 }
```

References [A1](#), [A2](#), [axis](#), [dis](#), [lambda](#), [Ph0](#), [PhA](#), [w0](#), and [zr](#).

5.3.4 Field Documentation

5.3.4.1 A1

`sunrealtype Gauss3D::A1 [private]`

amplitude projection on TE-mode (z-axis)

Definition at line 215 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.2 A2

`sunrealtype Gauss3D::A2 [private]`

amplitude projection on xy-plane

Definition at line 217 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.3 Amp

`sunrealtype Gauss3D::Amp [private]`

amplitude A

Definition at line 200 of file [ICSetters.h](#).

Referenced by [Gauss3D\(\)](#).

5.3.4.4 axis

`std::array<sunrealtype, 3> Gauss3D::axis [private]`

normalized propagation axis

Definition at line 198 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.5 dis

```
std::array<sunrealtype, 3> Gauss3D::dis [private]
```

distance maximum to origin

Definition at line 196 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.6 lambda

```
sunrealtype Gauss3D::lambda [private]
```

wavelength λ

Definition at line 219 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.7 Ph0

```
sunrealtype Gauss3D::Ph0 [private]
```

center of beam Φ_0

Definition at line 211 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.8 PhA

```
sunrealtype Gauss3D::PhA [private]
```

length of beam Φ_A

Definition at line 213 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.9 `phip`

`sunrealtype Gauss3D::phip [private]`

polarization rotation from TE-mode around propagation direction

Definition at line 203 of file [ICSetters.h](#).

Referenced by [Gauss3D\(\)](#).

5.3.4.10 `w0`

`sunrealtype Gauss3D::w0 [private]`

taille ω_0

Definition at line 207 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

5.3.4.11 `zr`

`sunrealtype Gauss3D::zr [private]`

Rayleigh length $z_R = \pi\omega_0^2/\lambda$.

Definition at line 209 of file [ICSetters.h](#).

Referenced by [addToSpace\(\)](#), and [Gauss3D\(\)](#).

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

- `src/ICSetters.h`
- `src/ICSetters.cpp`

5.4 gaussian1D Struct Reference

1D Gaussian wave structure

```
#include <src/SimulationFunctions.h>
```

Data Fields

- `std::array<sunrealtype, 3> k`
- `std::array<sunrealtype, 3> p`
- `std::array<sunrealtype, 3> x0`
- `sunrealtype phig`
- `std::array<sunrealtype, 3> phi`

5.4.1 Detailed Description

1D Gaussian wave structure

Definition at line 26 of file [SimulationFunctions.h](#).

5.4.2 Field Documentation

5.4.2.1 k

```
std::array<sunrealtype, 3> gaussian1D::k  
wavevector (normalized to  $1/\lambda$ )
```

Definition at line 27 of file [SimulationFunctions.h](#).

5.4.2.2 p

```
std::array<sunrealtype, 3> gaussian1D::p  
amplitude & polarization vector
```

Definition at line 28 of file [SimulationFunctions.h](#).

5.4.2.3 phi

```
std::array<sunrealtype, 3> gaussian1D::phi  
phase shift
```

Definition at line 31 of file [SimulationFunctions.h](#).

5.4.2.4 phig

```
sunrealtype gaussian1D::phig  
width
```

Definition at line 30 of file [SimulationFunctions.h](#).

5.4.2.5 x0

```
std::array<sunrealtype, 3> gaussian1D::x0
```

shift from origin

Definition at line 29 of file [SimulationFunctions.h](#).

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

- [src/SimulationFunctions.h](#)

5.5 gaussian2D Struct Reference

2D Gaussian wave structure

```
#include <src/SimulationFunctions.h>
```

Data Fields

- `std::array<sunrealtype, 3> x0`
- `std::array<sunrealtype, 3> axis`
- `sunrealtype amp`
- `sunrealtype phip`
- `sunrealtype w0`
- `sunrealtype zr`
- `sunrealtype ph0`
- `sunrealtype phA`

5.5.1 Detailed Description

2D Gaussian wave structure

Definition at line 35 of file [SimulationFunctions.h](#).

5.5.2 Field Documentation

5.5.2.1 amp

```
sunrealtype gaussian2D::amp
```

amplitude

Definition at line 38 of file [SimulationFunctions.h](#).

5.5.2.2 axis

std::array<sunrealtype, 3> gaussian2D::axis
direction from where it comes

Definition at line 37 of file [SimulationFunctions.h](#).

5.5.2.3 ph0

sunrealtype gaussian2D::ph0
beam center

Definition at line 42 of file [SimulationFunctions.h](#).

5.5.2.4 phA

sunrealtype gaussian2D::phA
beam length

Definition at line 43 of file [SimulationFunctions.h](#).

5.5.2.5 phip

sunrealtype gaussian2D::phip
polarization rotation

Definition at line 39 of file [SimulationFunctions.h](#).

5.5.2.6 w0

sunrealtype gaussian2D::w0
taille

Definition at line 40 of file [SimulationFunctions.h](#).

5.5.2.7 x0

```
std::array<sunrealtype, 3> gaussian2D::x0
```

center

Definition at line 36 of file [SimulationFunctions.h](#).

5.5.2.8 zr

```
sunrealtype gaussian2D::zr
```

Rayleigh length

Definition at line 41 of file [SimulationFunctions.h](#).

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

- src/[SimulationFunctions.h](#)

5.6 gaussian3D Struct Reference

3D Gaussian wave structure

```
#include <src/SimulationFunctions.h>
```

Data Fields

- std::array<sunrealtype, 3> x0
- std::array<sunrealtype, 3> axis
- unrealtype amp
- unrealtype phiP
- unrealtype w0
- unrealtype zr
- unrealtype ph0
- unrealtype phA

5.6.1 Detailed Description

3D Gaussian wave structure

Definition at line 47 of file [SimulationFunctions.h](#).

5.6.2 Field Documentation

5.6.2.1 amp

`sunrealtype gaussian3D::amp`

amplitude

Definition at line 50 of file [SimulationFunctions.h](#).

5.6.2.2 axis

`std::array<sunrealtype, 3> gaussian3D::axis`

direction from where it comes

Definition at line 49 of file [SimulationFunctions.h](#).

5.6.2.3 ph0

`sunrealtype gaussian3D::ph0`

beam center

Definition at line 54 of file [SimulationFunctions.h](#).

5.6.2.4 phA

`sunrealtype gaussian3D::phA`

beam length

Definition at line 55 of file [SimulationFunctions.h](#).

5.6.2.5 phip

`sunrealtype gaussian3D::phip`

polarization rotation

Definition at line 51 of file [SimulationFunctions.h](#).

5.6.2.6 w0

```
sunrealtype gaussian3D::w0
```

taille

Definition at line 52 of file [SimulationFunctions.h](#).

5.6.2.7 x0

```
std::array<sunrealtype, 3> gaussian3D::x0
```

center

Definition at line 48 of file [SimulationFunctions.h](#).

5.6.2.8 zr

```
sunrealtype gaussian3D::zr
```

Rayleigh length

Definition at line 53 of file [SimulationFunctions.h](#).

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

- src/[SimulationFunctions.h](#)

5.7 ICSetter Class Reference

[ICSetter](#) class to initialize wave types with default parameters.

```
#include <src/ICSetters.h>
```

Public Member Functions

- void [eval](#) (sunrealtype x, unrealtype y, unrealtype z, unrealtype *pTo6Space)
function to set all coordinates to zero and then add the field values
- void [add](#) (unrealtype x, unrealtype y, unrealtype z, unrealtype *pTo6Space)
function to fill the lattice space with initial field values
- void [addPlaneWave1D](#) (std::array< unrealtype, 3 > k={1, 0, 0}, std::array< unrealtype, 3 > p={0, 0, 1}, std::array< unrealtype, 3 > phi={0, 0, 0})
function to add plane waves in 1D to their container vector
- void [addPlaneWave2D](#) (std::array< unrealtype, 3 > k={1, 0, 0}, std::array< unrealtype, 3 > p={0, 0, 1}, std::array< unrealtype, 3 > phi={0, 0, 0})
function to add plane waves in 2D to their container vector
- void [addPlaneWave3D](#) (std::array< unrealtype, 3 > k={1, 0, 0}, std::array< unrealtype, 3 > p={0, 0, 1}, std::array< unrealtype, 3 > phi={0, 0, 0})
function to add plane waves in 3D to their container vector
- void [addGauss1D](#) (std::array< unrealtype, 3 > k={1, 0, 0}, std::array< unrealtype, 3 > p={0, 0, 1}, std::array< unrealtype, 3 > xo={0, 0, 0}, unrealtype phig_=1.0, std::array< unrealtype, 3 > phi={0, 0, 0})
function to add Gaussian wave packets in 1D to their container vector
- void [addGauss2D](#) (std::array< unrealtype, 3 > dis_={0, 0, 0}, std::array< unrealtype, 3 > axis_={1, 0, 0}, unrealtype Amp_=1.0, unrealtype phip_=0, unrealtype w0_=1e-5, unrealtype zr_=4e-5, unrealtype Ph0_=2e-5, unrealtype PhA_=0.45e-5)
function to add Gaussian wave packets in 2D to their container vector
- void [addGauss3D](#) (std::array< unrealtype, 3 > dis_={0, 0, 0}, std::array< unrealtype, 3 > axis_={1, 0, 0}, unrealtype Amp_=1.0, unrealtype phip_=0, unrealtype w0_=1e-5, unrealtype zr_=4e-5, unrealtype Ph0_=2e-5, unrealtype PhA_=0.45e-5)
function to add Gaussian wave packets in 3D to their container vector

Private Attributes

- std::vector< [PlaneWave1D](#) > [planeWaves1D](#)
container vector for plane waves in 1D
- std::vector< [PlaneWave2D](#) > [planeWaves2D](#)
container vector for plane waves in 2D
- std::vector< [PlaneWave3D](#) > [planeWaves3D](#)
container vector for plane waves in 3D
- std::vector< [Gauss1D](#) > [gauss1Ds](#)
container vector for Gaussian wave packets in 1D
- std::vector< [Gauss2D](#) > [gauss2Ds](#)
container vector for Gaussian wave packets in 2D
- std::vector< [Gauss3D](#) > [gauss3Ds](#)
container vector for Gaussian wave packets in 3D

5.7.1 Detailed Description

[ICSetter](#) class to initialize wave types with default parameters.

Definition at line 238 of file [ICSetters.h](#).

5.7.2 Member Function Documentation

5.7.2.1 add()

```
void ICSetter::add (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space )
```

function to fill the lattice space with initial field values

Add all initial field values to the lattice space

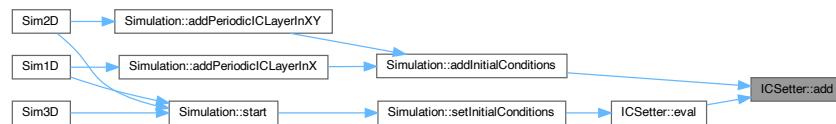
Definition at line 309 of file [ICSetters.cpp](#).

```
00310   for (const auto &wave : planeWaves1D)
00311     wave.addToSpace(x, y, z, pTo6Space);
00312   for (const auto &wave : planeWaves2D)
00313     wave.addToSpace(x, y, z, pTo6Space);
00314   for (const auto &wave : planeWaves3D)
00315     wave.addToSpace(x, y, z, pTo6Space);
00316   for (const auto &wave : gauss1Ds)
00317     wave.addToSpace(x, y, z, pTo6Space);
00318   for (const auto &wave : gauss2Ds)
00319     wave.addToSpace(x, y, z, pTo6Space);
00320   for (const auto &wave : gauss3Ds)
00321     wave.addToSpace(x, y, z, pTo6Space);
00322 }
00323 }
```

References [gauss1Ds](#), [gauss2Ds](#), [gauss3Ds](#), [planeWaves1D](#), [planeWaves2D](#), and [planeWaves3D](#).

Referenced by [Simulation::addInitialConditions\(\)](#), and [eval\(\)](#).

Here is the caller graph for this function:



5.7.2.2 addGauss1D()

```
void ICSetter::addGauss1D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > xo = {0, 0, 0},
    sunrealtype phig_ = 1.0,
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
```

function to add Gaussian wave packets in 1D to their container vector

Add Gaussian waves in 1D to their container vector

Definition at line 347 of file [ICSetters.cpp](#).

```
00350   {
00351   gauss1Ds.emplace_back(Gauss1D(k, p, xo, phig_, phi));
```

```
00352 }
```

References [gauss1Ds](#).

Referenced by [Sim1D\(\)](#).

Here is the caller graph for this function:



5.7.2.3 addGauss2D()

```
void ICSetter::addGauss2D (
    std::array< sunrealtype, 3 > dis_ = {0, 0, 0},
    std::array< sunrealtype, 3 > axis_ = {1, 0, 0},
    sunrealtype Amp_ = 1.0,
    sunrealtype phip_ = 0,
    sunrealtype w0_ = 1e-5,
    sunrealtype zr_ = 4e-5,
    sunrealtype Ph0_ = 2e-5,
    sunrealtype PhA_ = 0.45e-5 )
```

function to add Gaussian wave packets in 2D to their container vector

Add Gaussian waves in 2D to their container vector

Definition at line 355 of file [ICSetters.cpp](#).

```
00359 {
00360     gauss2Ds.emplace_back(
00361         Gauss2D(dis_, axis_, Amp_, phip_, w0_, zr_, Ph0_, PhA_));
00362 }
```

References [gauss2Ds](#).

Referenced by [Sim2D\(\)](#).

Here is the caller graph for this function:



5.7.2.4 addGauss3D()

```
void ICSetter::addGauss3D (
    std::array< sunrealtype, 3 > dis_ = {0, 0, 0},
    std::array< sunrealtype, 3 > axis_ = {1, 0, 0},
    sunrealtype Amp_ = 1.0,
    sunrealtype phip_ = 0,
    sunrealtype w0_ = 1e-5,
    sunrealtype zr_ = 4e-5,
    sunrealtype Ph0_ = 2e-5,
    sunrealtype PhA_ = 0.45e-5 )
```

function to add Gaussian wave packets in 3D to their container vector

Add Gaussian waves in 3D to their container vector

Definition at line 365 of file [ICSetters.cpp](#).

```
00369 {
00370     gauss3Ds.emplace_back(
00371         Gauss3D(dis_, axis_, Amp_, phip_, w0_, zr_, Ph0_, PhA_));
00372 }
```

References [gauss3Ds](#).

Referenced by [Sim3D\(\)](#).

Here is the caller graph for this function:



5.7.2.5 addPlaneWave1D()

```
void ICSetter::addPlaneWave1D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
```

function to add plane waves in 1D to their container vector

Add plane waves in 1D to their container vector

Definition at line 326 of file [ICSetters.cpp](#).

```
00328 {
00329     planeWaves1D.emplace_back(PlaneWave1D(k, p, phi));
00330 }
```

References [planeWaves1D](#).

Referenced by [Sim1D\(\)](#).

Here is the caller graph for this function:



5.7.2.6 addPlaneWave2D()

```
void ICSetter::addPlaneWave2D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
```

function to add plane waves in 2D to their container vector

Add plane waves in 2D to their container vector

Definition at line 333 of file [ICSetters.cpp](#).

```
00335 {
00336     planeWaves2D.emplace_back(PlaneWave2D(k, p, phi));
00337 }
```

References [planeWaves2D](#).

Referenced by [Sim2D\(\)](#).

Here is the caller graph for this function:



5.7.2.7 addPlaneWave3D()

```
void ICSetter::addPlaneWave3D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
```

function to add plane waves in 3D to their container vector

Add plane waves in 3D to their container vector

Definition at line 340 of file [ICSetters.cpp](#).

```
00342     {
00343     planeWaves3D.emplace_back(PlaneWave3D(k, p, phi));
00344 }
```

References [planeWaves3D](#).

Referenced by [Sim3D\(\)](#).

Here is the caller graph for this function:



5.7.2.8 eval()

```
void ICSetter::eval (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space )
```

function to set all coordinates to zero and then add the field values

Evaluate lattice point values to zero and then add initial field values

Definition at line 297 of file [ICSetters.cpp](#).

```
00298     {
00299     pTo6Space[0] = 0;
00300     pTo6Space[1] = 0;
00301     pTo6Space[2] = 0;
00302     pTo6Space[3] = 0;
00303     pTo6Space[4] = 0;
00304     pTo6Space[5] = 0;
00305     add(x, y, z, pTo6Space);
00306 }
```

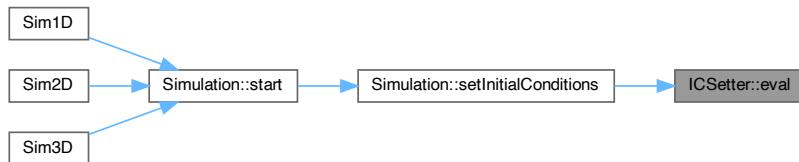
References [add\(\)](#).

Referenced by [Simulation::setInitialConditions\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.7.3 Field Documentation

5.7.3.1 gauss1Ds

```
std::vector<Gauss1D> ICSetter::gauss1Ds [private]
```

container vector for Gaussian wave packets in 1D

Definition at line [247](#) of file [ICSetters.h](#).

Referenced by [add\(\)](#), and [addGauss1D\(\)](#).

5.7.3.2 gauss2Ds

```
std::vector<Gauss2D> ICSetter::gauss2Ds [private]
```

container vector for Gaussian wave packets in 2D

Definition at line [249](#) of file [ICSetters.h](#).

Referenced by [add\(\)](#), and [addGauss2D\(\)](#).

5.7.3.3 gauss3Ds

```
std::vector<Gauss3D> ICSetter::gauss3Ds [private]
```

container vector for Gaussian wave packets in 3D

Definition at line [251](#) of file [ICSetters.h](#).

Referenced by [add\(\)](#), and [addGauss3D\(\)](#).

5.7.3.4 planeWaves1D

```
std::vector<PlaneWave1D> ICSetter::planeWaves1D [private]
```

container vector for plane waves in 1D

Definition at line [241](#) of file [ICSetters.h](#).

Referenced by [add\(\)](#), and [addPlaneWave1D\(\)](#).

5.7.3.5 planeWaves2D

```
std::vector<PlaneWave2D> ICSetter::planeWaves2D [private]
```

container vector for plane waves in 2D

Definition at line [243](#) of file [ICSetters.h](#).

Referenced by [add\(\)](#), and [addPlaneWave2D\(\)](#).

5.7.3.6 planeWaves3D

```
std::vector<PlaneWave3D> ICSetter::planeWaves3D [private]
```

container vector for plane waves in 3D

Definition at line [245](#) of file [ICSetters.h](#).

Referenced by [add\(\)](#), and [addPlaneWave3D\(\)](#).

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

- [src/ICSetters.h](#)
- [src/ICSetters.cpp](#)

5.8 Lattice Class Reference

[Lattice](#) class for the construction of the enveloping discrete simulation space.

```
#include <src/LatticePatch.h>
```

Public Member Functions

- void [initializeCommunicator](#) (const int Nx, const int Ny, const int Nz, const bool per)
function to create and deploy the cartesian communicator
- [Lattice](#) (const int StO)
default construction
- void [setDiscreteDimensions](#) (const sunindextype _nx, const sunindextype _ny, const sunindextype _nz)
component function for resizing the discrete dimensions of the lattice
- void [setPhysicalDimensions](#) (const sunrealtype _lx, const sunrealtype _ly, const sunrealtype _lz)
component function for resizing the physical size of the lattice

- const sunrealtype & [get_tot_lx](#) () const
- const sunrealtype & [get_tot_ly](#) () const
- const sunrealtype & [get_tot_lz](#) () const
- const sunindextype & [get_tot_nx](#) () const
- const sunindextype & [get_tot_ny](#) () const
- const sunindextype & [get_tot_nz](#) () const
- const sunindextype & [get_tot_noP](#) () const
- const sunindextype & [get_tot_noDP](#) () const
- const sunrealtype & [get_dx](#) () const
- const sunrealtype & [get_dy](#) () const
- const sunrealtype & [get_dz](#) () const
- constexpr int [get_dataPointDimension](#) () const
- const int & [get_stencilOrder](#) () const
- const int & [get_ghostLayerWidth](#) () const

Data Fields

- int [n_prc](#)
number of MPI processes
- int [my_prc](#)
number of MPI process
- MPI_Comm [comm](#)
personal communicator of the lattice
- SUNContext [suncxt](#)
SUNContext object.

Private Attributes

- sunrealtype `tot_lx`
physical size of the lattice in x-direction
- sunrealtype `tot_ly`
physical size of the lattice in y-direction
- sunrealtype `tot_lz`
physical size of the lattice in z-direction
- sunindextype `tot_nx`
number of points in x-direction
- sunindextype `tot_ny`
number of points in y-direction
- sunindextype `tot_nz`
number of points in z-direction
- sunindextype `tot_noP`
total number of lattice points
- sunindextype `tot_noDP`
number of lattice points times data dimension of each point
- sunrealtype `dx`
physical distance between lattice points in x-direction
- sunrealtype `dy`
physical distance between lattice points in y-direction
- sunrealtype `dz`
physical distance between lattice points in z-direction
- const int `StencilOrder`
stencil order
- const int `ghostLayerWidth`
required width of ghost layers (depends on the stencil order)
- unsigned int `statusFlags`
lattice status flags

Static Private Attributes

- static constexpr int `dataPointDimension` = 6
dimension of each data point set once and for all

5.8.1 Detailed Description

[Lattice](#) class for the construction of the enveloping discrete simulation space.

Definition at line 52 of file [LatticePatch.h](#).

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Lattice()

```
Lattice::Lattice (
    const int Sto )
```

default construction

Construct the lattice and set the stencil order.

Definition at line 32 of file [LatticePatch.cpp](#).

```
00032     : stencilOrder(Sto),
00033     ghostLayerWidth(Sto/2+1) {
00034     statusFlags = 0;
00035 }
```

References [statusFlags](#).

5.8.3 Member Function Documentation

5.8.3.1 get_dataPointDimension()

```
constexpr int Lattice::get_dataPointDimension ( ) const [inline], [constexpr]
```

getter function

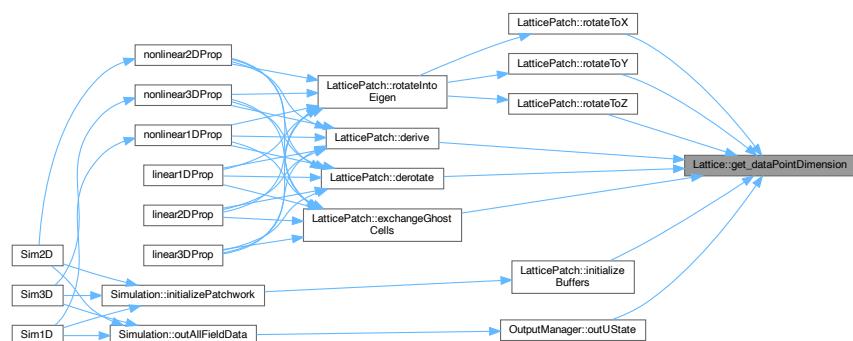
Definition at line 118 of file [LatticePatch.h](#).

```
00118 {
00119     return dataPointDimension;
00120 }
```

References [dataPointDimension](#).

Referenced by [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), [LatticePatch::initializeBuffers\(\)](#), [OutputManager::outUState\(\)](#), [LatticePatch::rotateToX\(\)](#), [LatticePatch::rotateToY\(\)](#), and [LatticePatch::rotateToZ\(\)](#).

Here is the caller graph for this function:



5.8.3.2 get_dx()

```
const sunrealtype & Lattice::get_dx ( ) const [inline]
```

getter function

Definition at line 115 of file [LatticePatch.h](#).

```
00115 { return dx; }
```

References [dx](#).

5.8.3.3 get_dy()

```
const sunrealtype & Lattice::get_dy ( ) const [inline]
```

getter function

Definition at line 116 of file [LatticePatch.h](#).

```
00116 { return dy; }
```

References [dy](#).

5.8.3.4 get_dz()

```
const sunrealtype & Lattice::get_dz ( ) const [inline]
```

getter function

Definition at line 117 of file [LatticePatch.h](#).

```
00117 { return dz; }
```

References [dz](#).

5.8.3.5 get_ghostLayerWidth()

```
const int & Lattice::get_ghostLayerWidth ( ) const [inline]
```

getter function

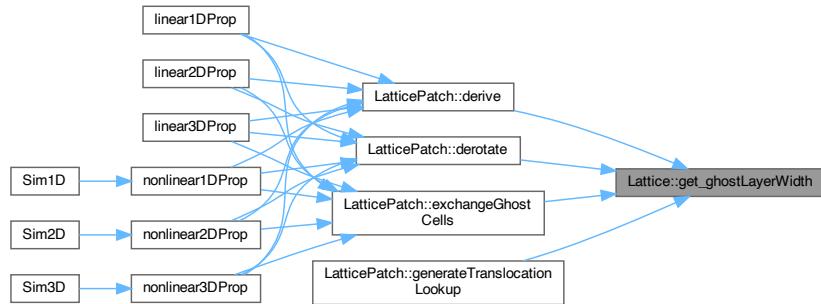
Definition at line 122 of file [LatticePatch.h](#).

```
00122
00123     return ghostLayerWidth;
00124 }
```

References [ghostLayerWidth](#).

Referenced by [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::generateTranslocationLookup\(\)](#).

Here is the caller graph for this function:



5.8.3.6 get_stencilOrder()

```
const int & Lattice::get_stencilOrder ( ) const [inline]
```

getter function

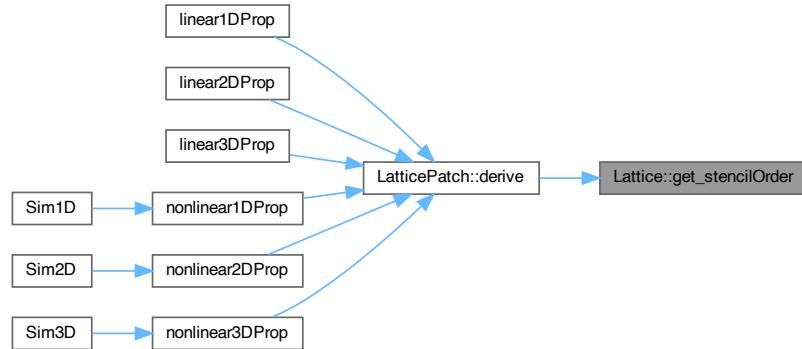
Definition at line 121 of file [LatticePatch.h](#).

```
00121 { return stencilOrder; }
```

References [stencilOrder](#).

Referenced by [LatticePatch::derive\(\)](#).

Here is the caller graph for this function:



5.8.3.7 `get_tot_lx()`

```
const sunrealtype & Lattice::get_tot_lx ( ) const [inline]
```

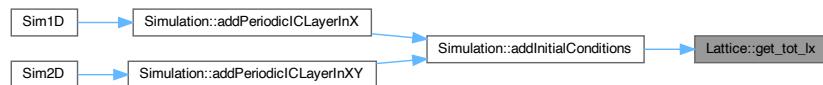
getter function

Definition at line 107 of file [LatticePatch.h](#).
00107 { **return tot_lx;** }

References [tot_lx](#).

Referenced by [Simulation::addInitialConditions\(\)](#).

Here is the caller graph for this function:



5.8.3.8 get_tot_ly()

```
const sunrealtype & Lattice::get_tot_ly ( ) const [inline]
```

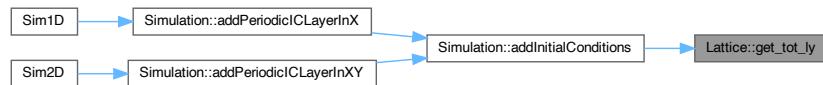
getter function

Definition at line 108 of file [LatticePatch.h](#).
00108 { **return tot_ly;** }

References [tot_ly](#).

Referenced by [Simulation::addInitialConditions\(\)](#).

Here is the caller graph for this function:



5.8.3.9 get_tot_lz()

```
const sunrealtype & Lattice::get_tot_lz ( ) const [inline]
```

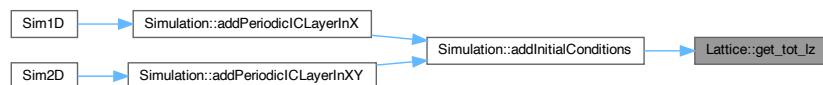
getter function

Definition at line 109 of file [LatticePatch.h](#).
00109 { **return tot_lz;** }

References [tot_lz](#).

Referenced by [Simulation::addInitialConditions\(\)](#).

Here is the caller graph for this function:



5.8.3.10 get_tot_noDP()

```
const sunindextype & Lattice::get_tot_noDP ( ) const [inline]
```

getter function

Definition at line 114 of file [LatticePatch.h](#).

```
00114 { return tot_noDP; }
```

References [tot_noDP](#).

5.8.3.11 get_tot_noP()

```
const sunindextype & Lattice::get_tot_noP ( ) const [inline]
```

getter function

Definition at line 113 of file [LatticePatch.h](#).

```
00113 { return tot_noP; }
```

References [tot_noP](#).

5.8.3.12 get_tot_nx()

```
const sunindextype & Lattice::get_tot_nx ( ) const [inline]
```

getter function

Definition at line 110 of file [LatticePatch.h](#).

```
00110 { return tot_nx; }
```

References [tot_nx](#).

5.8.3.13 get_tot_ny()

```
const sunindextype & Lattice::get_tot_ny ( ) const [inline]
```

getter function

Definition at line 111 of file [LatticePatch.h](#).

```
00111 { return tot_ny; }
```

References [tot_ny](#).

5.8.3.14 get_tot_nz()

```
const sunindextype & Lattice::get_tot_nz ( ) const [inline]
```

getter function

Definition at line 112 of file [LatticePatch.h](#).

```
00112 { return tot_nz; }
```

References [tot_nz](#).

5.8.3.15 initializeCommunicator()

```
void Lattice::initializeCommunicator (
    const int Nx,
    const int Ny,
    const int Nz,
    const bool per )
```

function to create and deploy the cartesian communicator

Initialize the cartesian communicator.

Definition at line 15 of file [LatticePatch.cpp](#).

```
00016 {
00017     const int dims[3] = {Nz, Ny, Nx};
00018     const int periods[3] = {static_cast<int>(per), static_cast<int>(per),
00019                             static_cast<int>(per)};
00020     // Create the cartesian communicator for MPI_COMM_WORLD
00021     MPI_Cart_create(MPI_COMM_WORLD, 3, dims, periods, 1, &comm);
00022     // Set MPI variables of the lattice
00023     MPI_Comm_size(comm, &n_prc);
00024     MPI_Comm_rank(comm, &my_prc);
00025     // Associate name to the communicator to identify it -> for debugging and
00026     // nicer error messages
00027     constexpr char lattice_comm_name[] = "Lattice";
00028     MPI_Comm_set_name(comm, lattice_comm_name);
00029 }
```

References [comm](#), [my_prc](#), and [n_prc](#).

Referenced by [Simulation::Simulation\(\)](#).

Here is the caller graph for this function:



5.8.3.16 setDiscreteDimensions()

```
void Lattice::setDiscreteDimensions (
    const sunindextype _nx,
    const sunindextype _ny,
    const sunindextype _nz )
```

component function for resizing the discrete dimensions of the lattice

Set the number of points in each dimension of the lattice.

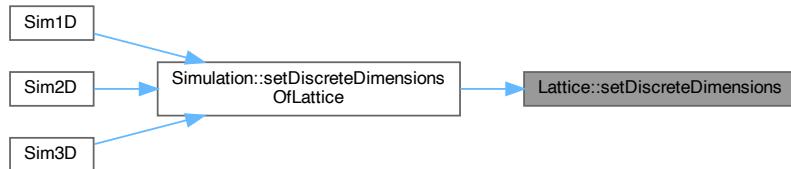
Definition at line 38 of file [LatticePatch.cpp](#).

```
00039 // copy the given data for number of points
00040 tot_nx = _nx;
00041 tot_ny = _ny;
00042 tot_nz = _nz;
00043 // compute the resulting number of points and datapoints
00044 tot_noP = tot_nx * tot_ny * tot_nz;
00045 tot_noDP = dataPointDimension * tot_noP;
00046 // compute the new Delta, the physical resolution
00047 dx = tot_lx / tot_nx;
00048 dy = tot_ly / tot_ny;
00049 dz = tot_lz / tot_nz;
00050
00051 }
```

References [dataPointDimension](#), [dx](#), [dy](#), [dz](#), [tot_lx](#), [tot_ly](#), [tot_lz](#), [tot_noDP](#), [tot_noP](#), [tot_nx](#), [tot_ny](#), and [tot_nz](#).

Referenced by [Simulation::setDiscreteDimensionsOfLattice\(\)](#).

Here is the caller graph for this function:



5.8.3.17 setPhysicalDimensions()

```
void Lattice::setPhysicalDimensions (
    const sunrealtype _lx,
    const sunrealtype _ly,
    const sunrealtype _lz )
```

component function for resizing the physical size of the lattice

Set the physical size of the lattice.

Definition at line 54 of file [LatticePatch.cpp](#).

```
00055
00056 tot_lx = _lx;
00057 tot_ly = _ly;
```

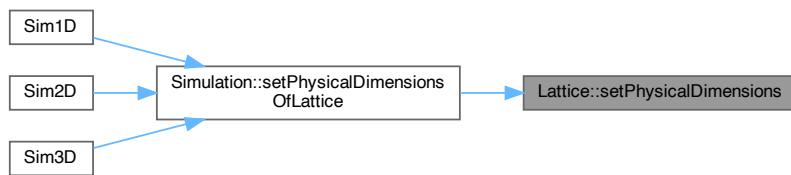
```

00058     tot_lz = _lz;
00059     // calculate physical distance between points
00060     dx = tot_lx / tot_nx;
00061     dy = tot_ly / tot_ny;
00062     dz = tot_lz / tot_nz;
00063     statusFlags |= FLatticeDimensionSet;
00064 }
```

References [dx](#), [dy](#), [dz](#), [FLatticeDimensionSet](#), [statusFlags](#), [tot_lx](#), [tot_ly](#), [tot_lz](#), [tot_nx](#), [tot_ny](#), and [tot_nz](#).

Referenced by [Simulation::setPhysicalDimensionsOfLattice\(\)](#).

Here is the caller graph for this function:



5.8.4 Field Documentation

5.8.4.1 comm

`MPI_Comm Lattice::comm`

personal communicator of the lattice

Definition at line 91 of file [LatticePatch.h](#).

Referenced by [Simulation::advanceToTime\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), [Simulation::get_cart_comm\(\)](#), [initializeCommunicator\(\)](#), [Simulation::initializeCVODEobject\(\)](#), [OutputManager::outUState\(\)](#), and [Simulation::Simulation\(\)](#).

5.8.4.2 dataPointDimension

`constexpr int Lattice::dataPointDimension = 6 [static], [constexpr], [private]`

dimension of each data point set once and for all

Definition at line 69 of file [LatticePatch.h](#).

Referenced by [get_dataPointDimension\(\)](#), and [setDiscreteDimensions\(\)](#).

5.8.4.3 dx

```
sunrealtype Lattice::dx [private]
```

physical distance between lattice points in x-direction

Definition at line 73 of file [LatticePatch.h](#).

Referenced by [get_dx\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.4 dy

```
sunrealtype Lattice::dy [private]
```

physical distance between lattice points in y-direction

Definition at line 75 of file [LatticePatch.h](#).

Referenced by [get_dy\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.5 dz

```
sunrealtype Lattice::dz [private]
```

physical distance between lattice points in z-direction

Definition at line 77 of file [LatticePatch.h](#).

Referenced by [get_dz\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.6 ghostLayerWidth

```
const int Lattice::ghostLayerWidth [private]
```

required width of ghost layers (depends on the stencil order)

Definition at line 81 of file [LatticePatch.h](#).

Referenced by [get_ghostLayerWidth\(\)](#).

5.8.4.7 my_prc

int Lattice::my_prc

number of MPI process

Definition at line 89 of file [LatticePatch.h](#).

Referenced by [Simulation::advanceToTime\(\)](#), [initializeCommunicator\(\)](#), [Simulation::initializeCVODEobject\(\)](#), [OutputManager::outUState\(\)](#), and [Simulation::Simulation\(\)](#).

5.8.4.8 n_prc

int Lattice::n_prc

number of MPI processes

Definition at line 87 of file [LatticePatch.h](#).

Referenced by [initializeCommunicator\(\)](#).

5.8.4.9 statusFlags

unsigned int Lattice::statusFlags [private]

lattice status flags

Definition at line 83 of file [LatticePatch.h](#).

Referenced by [Lattice\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.10 stencilOrder

const int Lattice::stencilOrder [private]

stencil order

Definition at line 79 of file [LatticePatch.h](#).

Referenced by [get_stencilOrder\(\)](#).

5.8.4.11 sunctx

```
SUNContext Lattice::sunctx
```

SUNContext object.

Definition at line 98 of file [LatticePatch.h](#).

Referenced by [Simulation::initializeCVODEobject\(\)](#), [Simulation::Simulation\(\)](#), and [Simulation::~Simulation\(\)](#).

5.8.4.12 tot_lx

```
sunrealtype Lattice::tot_lx [private]
```

physical size of the lattice in x-direction

Definition at line 55 of file [LatticePatch.h](#).

Referenced by [get_tot_lx\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.13 tot_ly

```
sunrealtype Lattice::tot_ly [private]
```

physical size of the lattice in y-direction

Definition at line 57 of file [LatticePatch.h](#).

Referenced by [get_tot_ly\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.14 tot_lz

```
sunrealtype Lattice::tot_lz [private]
```

physical size of the lattice in z-direction

Definition at line 59 of file [LatticePatch.h](#).

Referenced by [get_tot_lz\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.15 tot_noDP

```
sunindextype Lattice::tot_noDP [private]
```

number of lattice points times data dimension of each point

Definition at line 71 of file [LatticePatch.h](#).

Referenced by [get_tot_noDP\(\)](#), and [setDiscreteDimensions\(\)](#).

5.8.4.16 tot_noP

```
sunindextype Lattice::tot_noP [private]
```

total number of lattice points

Definition at line 67 of file [LatticePatch.h](#).

Referenced by [get_tot_noP\(\)](#), and [setDiscreteDimensions\(\)](#).

5.8.4.17 tot_nx

```
sunindextype Lattice::tot_nx [private]
```

number of points in x-direction

Definition at line 61 of file [LatticePatch.h](#).

Referenced by [get_tot_nx\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.18 tot_ny

```
sunindextype Lattice::tot_ny [private]
```

number of points in y-direction

Definition at line 63 of file [LatticePatch.h](#).

Referenced by [get_tot_ny\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

5.8.4.19 tot_nz

`sunindextype Lattice::tot_nz [private]`

number of points in z-direction

Definition at line 65 of file [LatticePatch.h](#).

Referenced by [get_tot_nz\(\)](#), [setDiscreteDimensions\(\)](#), and [setPhysicalDimensions\(\)](#).

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

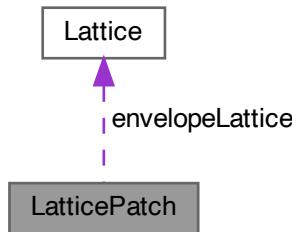
- src/[LatticePatch.h](#)
- src/[LatticePatch.cpp](#)

5.9 LatticePatch Class Reference

[LatticePatch](#) class for the construction of the patches in the enveloping lattice.

```
#include <src/LatticePatch.h>
```

Collaboration diagram for LatticePatch:



Public Member Functions

- [LatticePatch \(\)](#)
constructor setting up a default first lattice patch
- [~LatticePatch \(\)](#)
destructor freeing parallel vectors
- sunindextype [discreteSize \(int dir=0\) const](#)
function to get the discrete size of the LatticePatch
- sunrealtype [origin \(const int dir\) const](#)
function to get the origin of the patch
- sunrealtype [getDelta \(const int dir\) const](#)
function to get distance between points
- void [generateTranslocationLookup \(\)](#)

- function to fill out the lookup tables for cache efficiency
- void **rotateIntoEigen** (const int dir)
 - function to rotate u into Z-matrix eigenraum*
- void **derotate** (int dir, sunrealtype *buffOut)
 - function to derotate uAux into dudata lattice direction of x*
- void **initializeBuffers** ()
 - initialize buffers to save derivatives*
- void **exchangeGhostCells** (const int dir)
 - function to exchange ghost cells*
- void **derive** (const int dir)
 - function to derive the centered values in uAux and save them noncentered*
- void **checkFlag** (unsigned int flag) const
 - function to check if a flag has been set and if not abort*

Data Fields

- int **ID**
 - ID of the `LatticePatch`, corresponds to process number (for debugging)*
- N_Vector **uLocal**
 - NVector for saving field components $u=(E,B)$ in lattice points.*
- N_Vector **u**
- N_Vector **duLocal**
 - NVector for saving temporal derivatives of the field data.*
- N_Vector **du**
- sunrealtype * **uData**
 - pointer to field data*
- sunrealtype * **duData**
 - pointer to time-derivative data*
- sunrealtype * **uAuxData**
 - pointer to auxiliary data vector*
- std::array< sunrealtype *, 3 > **buffData**

- sunrealtype * **gCLData**
- sunrealtype * **gCRData**

Private Member Functions

- void **rotateToX** (sunrealtype *outArray, const sunrealtype *inArray, const std::vector< sunindextype > &lookup)
- void **rotateToY** (sunrealtype *outArray, const sunrealtype *inArray, const std::vector< sunindextype > &lookup)
- void **rotateToZ** (sunrealtype *outArray, const sunrealtype *inArray, const std::vector< sunindextype > &lookup)

Private Attributes

- sunrealtype `x0`
origin of the patch in physical space; x-coordinate
- sunrealtype `y0`
origin of the patch in physical space; y-coordinate
- sunrealtype `z0`
origin of the patch in physical space; z-coordinate
- sunindextype `Llx`
inner position of lattice-patch in the lattice patchwork; x-points
- sunindextype `Lly`
inner position of lattice-patch in the lattice patchwork; y-points
- sunindextype `Llz`
inner position of lattice-patch in the lattice patchwork; z-points
- sunrealtype `lx`
physical size of the lattice-patch in the x-dimension
- sunrealtype `ly`
physical size of the lattice-patch in the y-dimension
- sunrealtype `lz`
physical size of the lattice-patch in the z-dimension
- sunindextype `nx`
number of points in the lattice patch in the x-dimension
- sunindextype `ny`
number of points in the lattice patch in the y-dimension
- sunindextype `nz`
number of points in the lattice patch in the z-dimension
- sunrealtype `dx`
physical distance between lattice points in x-direction
- sunrealtype `dy`
physical distance between lattice points in y-direction
- sunrealtype `dz`
physical distance between lattice points in z-direction
- unsigned int `statusFlags`
lattice patch status flags
- const `Lattice * envelopeLattice`
pointer to the enveloping lattice
- std::vector< sunrealtype > `uAux`
aid (auxilliarily) vector including ghost cells to compute the derivatives

- std::vector< sunindextype > `uTox`
- std::vector< sunindextype > `uToy`
- std::vector< sunindextype > `uToz`
- std::vector< sunindextype > `xTou`
- std::vector< sunindextype > `yTou`
- std::vector< sunindextype > `zTou`

- std::vector< sunrealtype > `buffX`

- std::vector< sunrealtype > `buffY`
 - std::vector< sunrealtype > `buffZ`
-
- std::vector< sunrealtype > `ghostCellLeft`
 - std::vector< sunrealtype > `ghostCellRight`
 - std::vector< sunrealtype > `ghostCellLeftToSend`
 - std::vector< sunrealtype > `ghostCellRightToSend`
 - std::vector< sunrealtype > `ghostCellsToSend`
 - std::vector< sunrealtype > `ghostCells`
-
- std::vector< sunindextype > `lgcTox`
 - std::vector< sunindextype > `rgcTox`
 - std::vector< sunindextype > `lgcToy`
 - std::vector< sunindextype > `rgcToy`
 - std::vector< sunindextype > `lgcToz`
 - std::vector< sunindextype > `rgcToz`

Friends

- int `generatePatchwork` (const [Lattice](#) &envelopeLattice, [LatticePatch](#) &patchToMold, const int DLx, const int DLy, const int DLz)
friend function for creating the patchwork slicing of the overall lattice

5.9.1 Detailed Description

[LatticePatch](#) class for the construction of the patches in the enveloping lattice.

Definition at line 130 of file [LatticePatch.h](#).

5.9.2 Constructor & Destructor Documentation

5.9.2.1 LatticePatch()

LatticePatch::LatticePatch ()

constructor setting up a default first lattice patch

Construct the lattice patch.

Definition at line 71 of file [LatticePatch.cpp](#).

```
00071   {
00072     // set default origin coordinates to (0,0,0)
00073     x0 = y0 = z0 = 0;
00074     // set default position in Lattice-Patchwork to (0,0,0)
00075     Llx = Lly = Llz = 0;
00076     // set default physical length for lattice patch to (0,0,0)
00077     lx = ly = lz = 0;
00078     // set default discrete length for lattice patch to (0,1,1)
00079     /* This is done in this manner as even in 1D simulations require a 1 point
00080      * width */
00081     nx = 0;
00082     ny = nz = 1;
00083
00084     // u is not initialized as it wouldn't make any sense before the dimensions
00085     // are set idem for the enveloping lattice
00086
00087     // set default statusFlags to non set
00088     statusFlags = 0;
00089 }
```

References [Llx](#), [Lly](#), [Llz](#), [lx](#), [ly](#), [lz](#), [nx](#), [ny](#), [nz](#), [statusFlags](#), [x0](#), [y0](#), and [z0](#).

5.9.2.2 ~LatticePatch()

LatticePatch::~LatticePatch ()

destructor freeing parallel vectors

Destruct the patch and thereby destroy the NVectors.

Definition at line 92 of file [LatticePatch.cpp](#).

```
00092   {
00093     // Deallocate memory for solution vector
00094     if (statusFlags & FLatticePatchSetUp) {
00095       // Destroy data vectors
00096       #if defined(_OPENMP)
00097         N_VDestroy(u);
00098         N_VDestroy(du);
00099         N_VDestroy_OpenMP(uLocal);
00100         N_VDestroy_OpenMP(duLocal);
00101     #else
00102       N_VDestroy_Parallel(u);
00103       N_VDestroy_Parallel(du);
00104     #endif
00105   }
00106 }
```

References [du](#), [duLocal](#), [FLatticePatchSetUp](#), [statusFlags](#), [u](#), and [uLocal](#).

5.9.3 Member Function Documentation

5.9.3.1 checkFlag()

```
void LatticePatch::checkFlag (
    unsigned int flag ) const
```

function to check if a flag has been set and if not abort

Check if all flags are set.

Definition at line 631 of file [LatticePatch.cpp](#).

```
00631
00632     if (!(statusFlags & flag)) {
00633         std::string errorMessage;
00634         switch (flag) {
00635             case FLatticePatchSetUp:
00636                 errorMessage = "The Lattice patch was not set up please make sure to "
00637                             "initialize a Lattice topology";
00638                 break;
00639             case TranslocationLookupSetUp:
00640                 errorMessage = "The translocation lookup tables have not been generated, "
00641                             "please be sure to run generateTranslocationLookup()";
00642                 break;
00643             case GhostLayersInitialized:
00644                 errorMessage = "The space for the ghost layers has not been allocated, "
00645                             "please be sure that the ghost cells are initialized ";
00646                 break;
00647             case BuffersInitialized:
00648                 errorMessage = "The space for the buffers has not been allocated, please "
00649                             "be sure to run initializeBuffers()";
00650                 break;
00651         default:
00652             errorMessage = "Uppss, you've made a non-standard error, sadly I can't "
00653                             "help you there";
00654             break;
00655         }
00656         errorKill(errorMessage);
00657     }
00658     return;
00659 }
```

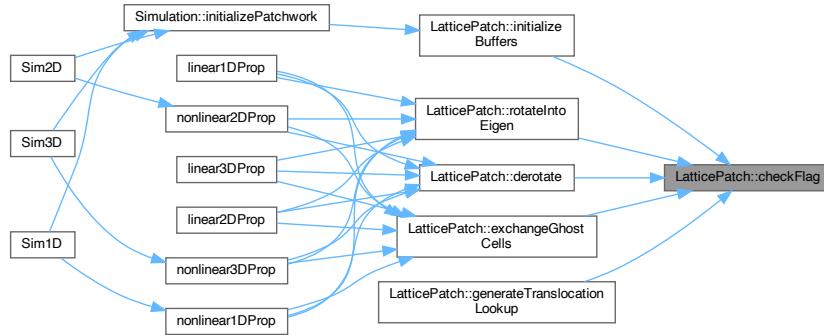
References [BuffersInitialized](#), [errorKill\(\)](#), [FLatticePatchSetUp](#), [GhostLayersInitialized](#), [statusFlags](#), and [TranslocationLookupSetUp](#).

Referenced by [derotate\(\)](#), [exchangeGhostCells\(\)](#), [generateTranslocationLookup\(\)](#), [initializeBuffers\(\)](#), and [rotateIntoEigen\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.2 derive()

```
void LatticePatch::derive (
    const int dir )
```

function to derive the centered values in uAux and save them noncentered

Calculate derivatives in the patch (uAux) in the specified direction.

Definition at line 662 of file [LatticePatch.cpp](#).

```
00662           {
00663     // ghost layer width adjusted to the chosen stencil order
00664     const int gLW = envelopeLattice->get_ghostLayerWidth();
00665     // dimensionality of data points -> 6
00666     const int dPD = envelopeLattice->get_dataPointDimension();
00667     // total width of patch in given direction including ghost layers at ends
00668     const sunindextype dirWidth = discreteSize(dir) + 2 * gLW;
00669     // width of patch only in given direction
00670     const sunindextype dirWidthO = discreteSize(dir);
00671     // size of plane perpendicular to given dimension
00672     const sunindextype perpPlainSize = discreteSize() / discreteSize(dir);
00673     // physical distance between points in that direction
00674     sunrealtype dxi = nan("0x12345");
00675     switch (dir) {
00676     case 1:
00677       dxi = dx;
00678       break;
00679     case 2:
00680       dxi = dy;
00681       break;
00682     case 3:
00683       dxi = dz;
00684       break;
00685     default:
00686       dxi = 1;
00687       errorKill("Tried to derive in the wrong direction");
00688       break;
00689     }
00690     // Derive according to chosen stencil accuracy order
00691     const int order = envelopeLattice->get_stencilOrder();
00692     switch (order) {
00693     case 1: // gLW=1
00694       #pragma omp parallel for default(none) \
00695       shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00696       for (sunindextype i = 0; i < perpPlainSize; i++) {
00697         #pragma omp simd
00698         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00699              j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00700           uAux[j + 0 - gLW * dPD] = slb(&uAux[j + 0]) / dxi;
00701         }
00702       }
00703     }
00704   }
```

```

00701     uAux[j + 1 - gLW * dPD] = s1b(&uAux[j + 1]) / dxii;
00702     uAux[j + 2 - gLW * dPD] = s1f(&uAux[j + 2]) / dxii;
00703     uAux[j + 3 - gLW * dPD] = s1f(&uAux[j + 3]) / dxii;
00704     uAux[j + 4 - gLW * dPD] = s1f(&uAux[j + 4]) / dxii;
00705     uAux[j + 5 - gLW * dPD] = s1f(&uAux[j + 5]) / dxii;
00706 }
00707 }
00708 break;
00709 case 2: // gLW=2
00710 #pragma omp parallel for default(none) \
00711 shared(perpPlainSize, dxii, dirWidth, dirWidthO, gLW, dPD, uAux)
00712 for (sunindextype i = 0; i < perpPlainSize; i++) {
00713 #pragma omp simd
00714 for (sunindextype j = (i * dirWidth + gLW) * dPD;
00715     j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00716     uAux[j + 0 - gLW * dPD] = s2b(&uAux[j + 0]) / dxii;
00717     uAux[j + 1 - gLW * dPD] = s2b(&uAux[j + 1]) / dxii;
00718     uAux[j + 2 - gLW * dPD] = s2f(&uAux[j + 2]) / dxii;
00719     uAux[j + 3 - gLW * dPD] = s2f(&uAux[j + 3]) / dxii;
00720     uAux[j + 4 - gLW * dPD] = s2c(&uAux[j + 4]) / dxii;
00721     uAux[j + 5 - gLW * dPD] = s2c(&uAux[j + 5]) / dxii;
00722 }
00723 }
00724 break;
00725 case 3: // gLW=2
00726 #pragma omp parallel for default(none) \
00727 shared(perpPlainSize, dxii, dirWidth, dirWidthO, gLW, dPD, uAux)
00728 for (sunindextype i = 0; i < perpPlainSize; i++) {
00729 #pragma omp simd
00730 for (sunindextype j = (i * dirWidth + gLW) * dPD;
00731     j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00732     uAux[j + 0 - gLW * dPD] = s3b(&uAux[j + 0]) / dxii;
00733     uAux[j + 1 - gLW * dPD] = s3b(&uAux[j + 1]) / dxii;
00734     uAux[j + 2 - gLW * dPD] = s3f(&uAux[j + 2]) / dxii;
00735     uAux[j + 3 - gLW * dPD] = s3f(&uAux[j + 3]) / dxii;
00736     uAux[j + 4 - gLW * dPD] = s3f(&uAux[j + 4]) / dxii;
00737     uAux[j + 5 - gLW * dPD] = s3f(&uAux[j + 5]) / dxii;
00738 }
00739 }
00740 break;
00741 case 4: // gLW=3
00742 #pragma omp parallel for default(none) \
00743 shared(perpPlainSize, dxii, dirWidth, dirWidthO, gLW, dPD, uAux)
00744 for (sunindextype i = 0; i < perpPlainSize; i++) {
00745 #pragma omp simd
00746 for (sunindextype j = (i * dirWidth + gLW) * dPD;
00747     j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00748     uAux[j + 0 - gLW * dPD] = s4b(&uAux[j + 0]) / dxii;
00749     uAux[j + 1 - gLW * dPD] = s4b(&uAux[j + 1]) / dxii;
00750     uAux[j + 2 - gLW * dPD] = s4f(&uAux[j + 2]) / dxii;
00751     uAux[j + 3 - gLW * dPD] = s4f(&uAux[j + 3]) / dxii;
00752     uAux[j + 4 - gLW * dPD] = s4c(&uAux[j + 4]) / dxii;
00753     uAux[j + 5 - gLW * dPD] = s4c(&uAux[j + 5]) / dxii;
00754 }
00755 }
00756 break;
00757 case 5: // gLW=3
00758 #pragma omp parallel for default(none) \
00759 shared(perpPlainSize, dxii, dirWidth, dirWidthO, gLW, dPD, uAux)
00760 for (sunindextype i = 0; i < perpPlainSize; i++) {
00761 #pragma omp simd
00762 for (sunindextype j = (i * dirWidth + gLW) * dPD;
00763     j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00764     uAux[j + 0 - gLW * dPD] = s5b(&uAux[j + 0]) / dxii;
00765     uAux[j + 1 - gLW * dPD] = s5b(&uAux[j + 1]) / dxii;
00766     uAux[j + 2 - gLW * dPD] = s5f(&uAux[j + 2]) / dxii;
00767     uAux[j + 3 - gLW * dPD] = s5f(&uAux[j + 3]) / dxii;
00768     uAux[j + 4 - gLW * dPD] = s5f(&uAux[j + 4]) / dxii;
00769     uAux[j + 5 - gLW * dPD] = s5f(&uAux[j + 5]) / dxii;
00770 }
00771 }
00772 break;
00773 case 6: // gLW=4
00774 #pragma omp parallel for default(none) \
00775 shared(perpPlainSize, dxii, dirWidth, dirWidthO, gLW, dPD, uAux)
00776 for (sunindextype i = 0; i < perpPlainSize; i++) {
00777 #pragma omp simd
00778 for (sunindextype j = (i * dirWidth + gLW) * dPD;
00779     j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00780     uAux[j + 0 - gLW * dPD] = s6b(&uAux[j + 0]) / dxii;
00781     uAux[j + 1 - gLW * dPD] = s6b(&uAux[j + 1]) / dxii;
00782     uAux[j + 2 - gLW * dPD] = s6f(&uAux[j + 2]) / dxii;
00783     uAux[j + 3 - gLW * dPD] = s6f(&uAux[j + 3]) / dxii;
00784     uAux[j + 4 - gLW * dPD] = s6c(&uAux[j + 4]) / dxii;
00785     uAux[j + 5 - gLW * dPD] = s6c(&uAux[j + 5]) / dxii;
00786 }
00787 }

```

```

00788     break;
00789 case 7: // gLW=4
00790 #pragma omp parallel for default(none) \
00791 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00792 for (sunindextype i = 0; i < perpPlainSize; i++) {
00793     #pragma omp simd
00794     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00795         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00796         uAux[j + 0 - gLW * dPD] = s7b(&uAux[j + 0]) / dxi;
00797         uAux[j + 1 - gLW * dPD] = s7b(&uAux[j + 1]) / dxi;
00798         uAux[j + 2 - gLW * dPD] = s7f(&uAux[j + 2]) / dxi;
00799         uAux[j + 3 - gLW * dPD] = s7f(&uAux[j + 3]) / dxi;
00800         uAux[j + 4 - gLW * dPD] = s7f(&uAux[j + 4]) / dxi;
00801         uAux[j + 5 - gLW * dPD] = s7f(&uAux[j + 5]) / dxi;
00802     }
00803 }
00804 break;
00805 case 8: // gLW=5
00806 #pragma omp parallel for default(none) \
00807 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00808 for (sunindextype i = 0; i < perpPlainSize; i++) {
00809     #pragma omp simd
00810     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00811         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00812         uAux[j + 0 - gLW * dPD] = s8b(&uAux[j + 0]) / dxi;
00813         uAux[j + 1 - gLW * dPD] = s8b(&uAux[j + 1]) / dxi;
00814         uAux[j + 2 - gLW * dPD] = s8f(&uAux[j + 2]) / dxi;
00815         uAux[j + 3 - gLW * dPD] = s8f(&uAux[j + 3]) / dxi;
00816         uAux[j + 4 - gLW * dPD] = s8c(&uAux[j + 4]) / dxi;
00817         uAux[j + 5 - gLW * dPD] = s8c(&uAux[j + 5]) / dxi;
00818     }
00819 }
00820 break;
00821 case 9: // gLW=5
00822 #pragma omp parallel for default(none) \
00823 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00824 for (sunindextype i = 0; i < perpPlainSize; i++) {
00825     #pragma omp simd
00826     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00827         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00828         uAux[j + 0 - gLW * dPD] = s9b(&uAux[j + 0]) / dxi;
00829         uAux[j + 1 - gLW * dPD] = s9b(&uAux[j + 1]) / dxi;
00830         uAux[j + 2 - gLW * dPD] = s9f(&uAux[j + 2]) / dxi;
00831         uAux[j + 3 - gLW * dPD] = s9f(&uAux[j + 3]) / dxi;
00832         uAux[j + 4 - gLW * dPD] = s9f(&uAux[j + 4]) / dxi;
00833         uAux[j + 5 - gLW * dPD] = s9f(&uAux[j + 5]) / dxi;
00834     }
00835 }
00836 break;
00837 case 10: // gLW=6
00838 #pragma omp parallel for default(none) \
00839 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00840 for (sunindextype i = 0; i < perpPlainSize; i++) {
00841     #pragma omp simd
00842     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00843         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00844         uAux[j + 0 - gLW * dPD] = s10b(&uAux[j + 0]) / dxi;
00845         uAux[j + 1 - gLW * dPD] = s10b(&uAux[j + 1]) / dxi;
00846         uAux[j + 2 - gLW * dPD] = s10f(&uAux[j + 2]) / dxi;
00847         uAux[j + 3 - gLW * dPD] = s10f(&uAux[j + 3]) / dxi;
00848         uAux[j + 4 - gLW * dPD] = s10c(&uAux[j + 4]) / dxi;
00849         uAux[j + 5 - gLW * dPD] = s10c(&uAux[j + 5]) / dxi;
00850     }
00851 }
00852 break;
00853 case 11: // gLW=6
00854 #pragma omp parallel for default(none) \
00855 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00856 for (sunindextype i = 0; i < perpPlainSize; i++) {
00857     #pragma omp simd
00858     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00859         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00860         uAux[j + 0 - gLW * dPD] = s11b(&uAux[j + 0]) / dxi;
00861         uAux[j + 1 - gLW * dPD] = s11b(&uAux[j + 1]) / dxi;
00862         uAux[j + 2 - gLW * dPD] = s11f(&uAux[j + 2]) / dxi;
00863         uAux[j + 3 - gLW * dPD] = s11f(&uAux[j + 3]) / dxi;
00864         uAux[j + 4 - gLW * dPD] = s11f(&uAux[j + 4]) / dxi;
00865         uAux[j + 5 - gLW * dPD] = s11f(&uAux[j + 5]) / dxi;
00866     }
00867 }
00868 break;
00869 case 12: // gLW=7
00870 #pragma omp parallel for default(none) \
00871 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00872 for (sunindextype i = 0; i < perpPlainSize; i++) {
00873     #pragma omp simd
00874     for (sunindextype j = (i * dirWidth + gLW) * dPD;

```

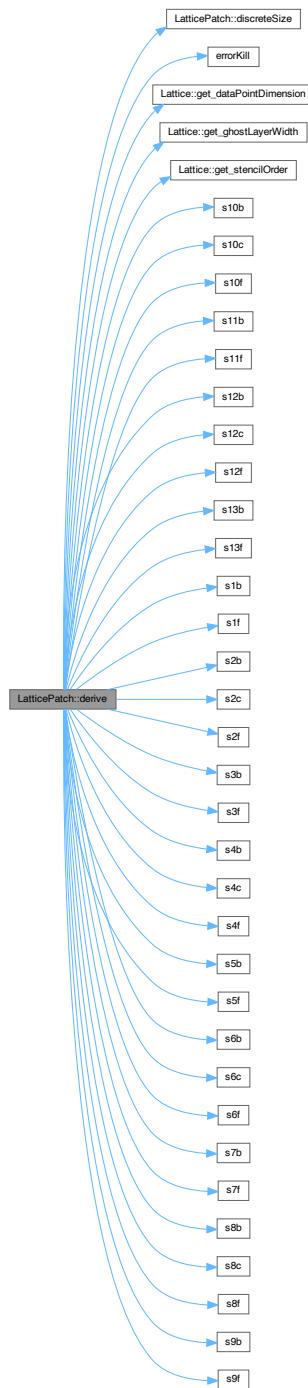
```

00875     j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00876     uAux[j + 0 - gLW * dPD] = s12b(&uAux[j + 0]) / dx;
00877     uAux[j + 1 - gLW * dPD] = s12b(&uAux[j + 1]) / dx;
00878     uAux[j + 2 - gLW * dPD] = s12f(&uAux[j + 2]) / dx;
00879     uAux[j + 3 - gLW * dPD] = s12f(&uAux[j + 3]) / dx;
00880     uAux[j + 4 - gLW * dPD] = s12c(&uAux[j + 4]) / dx;
00881     uAux[j + 5 - gLW * dPD] = s12c(&uAux[j + 5]) / dx;
00882 }
00883 }
00884 break;
00885 case 13: // gLW=7
00886 // For all points in the plane perpendicular to the given direction
00887 #pragma omp parallel for default(none) \
00888 shared(perpPlainSize, dx, dirWidth, dirWidthO, gLW, dPD, uAux)
00889 for (sunindextype i = 0; i < perpPlainSize; i++) {
00890     // iterate through the derivation direction
00891     #pragma omp simd
00892     for (sunindextype j = (i * dirWidth
00893         + gLW /*to shift left by gLW below */) * dPD;
00894         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00895         // Compute the stencil derivative for any of the six field components
00896         // and update position by ghost width shift
00897         uAux[j + 0 - gLW * dPD] = s13b(&uAux[j + 0]) / dx;
00898         uAux[j + 1 - gLW * dPD] = s13b(&uAux[j + 1]) / dx;
00899         uAux[j + 2 - gLW * dPD] = s13f(&uAux[j + 2]) / dx;
00900         uAux[j + 3 - gLW * dPD] = s13f(&uAux[j + 3]) / dx;
00901         uAux[j + 4 - gLW * dPD] = s13f(&uAux[j + 4]) / dx;
00902         uAux[j + 5 - gLW * dPD] = s13f(&uAux[j + 5]) / dx;
00903     }
00904 }
00905 break;
00906 default:
00907     errorKill("Please set an existing stencil order");
00908     break;
00909 }
00910 }
00911 }
```

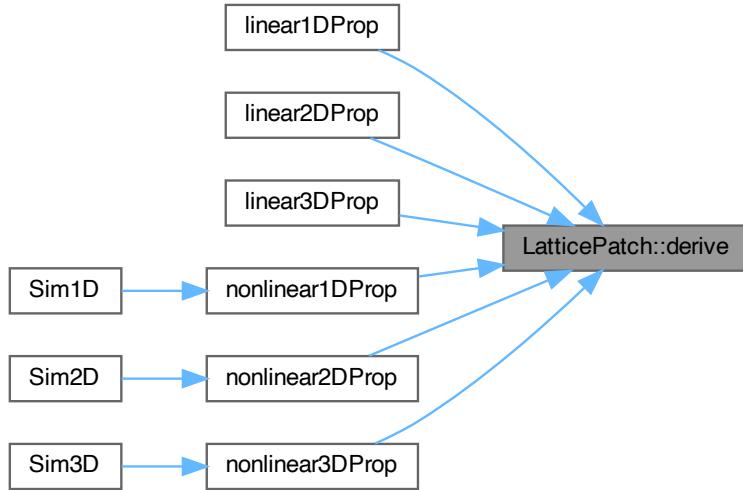
References [discreteSize\(\)](#), [dx](#), [dy](#), [dz](#), [envelopeLattice](#), [errorKill\(\)](#), [Lattice::get_dataPointDimension\(\)](#), [Lattice::get_ghostLayerWidth\(\)](#), [Lattice::get_stencilOrder\(\)](#), [s10b\(\)](#), [s10c\(\)](#), [s10f\(\)](#), [s11b\(\)](#), [s11f\(\)](#), [s12b\(\)](#), [s12c\(\)](#), [s12f\(\)](#), [s13b\(\)](#), [s13f\(\)](#), [s1b\(\)](#), [s1f\(\)](#), [s2b\(\)](#), [s2c\(\)](#), [s2f\(\)](#), [s3b\(\)](#), [s3f\(\)](#), [s4b\(\)](#), [s4c\(\)](#), [s4f\(\)](#), [s5b\(\)](#), [s5f\(\)](#), [s6b\(\)](#), [s6c\(\)](#), [s6f\(\)](#), [s7b\(\)](#), [s7f\(\)](#), [s8b\(\)](#), [s8c\(\)](#), [s8f\(\)](#), [s9b\(\)](#), [s9f\(\)](#), and [uAux](#).

Referenced by [linear1DProp\(\)](#), [linear2DProp\(\)](#), [linear3DProp\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), and [nonlinear3DProp\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.3 derotate()

```
void LatticePatch::derotate (
    int dir,
    sunrealtype * buffOut )
```

function to derotate uAux into dudata lattice direction of x

Derotate uAux with transposed rotation matrices and write to derivative buffer – normalization is done here by the factor 1/2

Definition at line 446 of file [LatticePatch.cpp](#).

```
00446
00447 // Check that the lattice as well as the translocation lookups have been set
00448 // up;
00449 checkFlag(FLatticePatchSetUp);
00450 checkFlag(TranslocationLookupSetUp);
00451 const int dPD = envelopeLattice->get_dataPointDimension();
00452 const int gLW = envelopeLattice->get_ghostLayerWidth();
00453 const sunindextype totalNP = discreteSize();
00454 sunindextype ii = 0, target = 0;
00455 switch (dir) {
00456 case 1:
00457 #pragma omp parallel for simd \
00458 private(ii, target) \
00459 shared(dPD, gLW, totalNP, uTox, uAux, buffOut) \
00460 schedule(static)
00461 for (sunindextype i = 0; i < totalNP; i++) {
00462 // get correct indices in u and rotation space
00463 target = dPD * i;
00464 ii = dPD * (uTox[i] - gLW);
00465 buffOut[target + 0] = uAux[5 + ii];
00466 buffOut[target + 1] = (-uAux[ii] + uAux[2 + ii]) / 2.;
00467 buffOut[target + 2] = (uAux[1 + ii] - uAux[3 + ii]) / 2.;
00468 buffOut[target + 3] = uAux[4 + ii];
00469 buffOut[target + 4] = (uAux[1 + ii] + uAux[3 + ii]) / 2.;
00470 buffOut[target + 5] = (uAux[ii] + uAux[2 + ii]) / 2.;
```

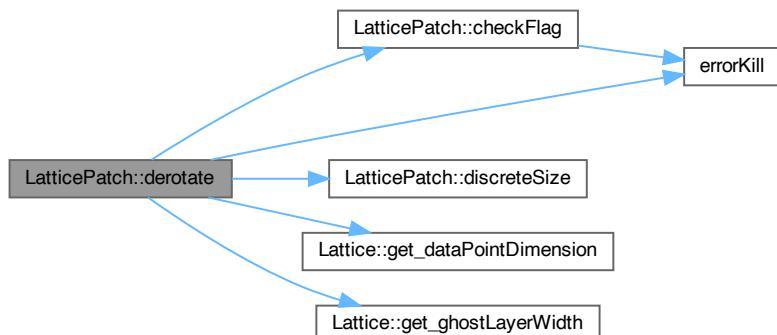
```

00471      }
00472      break;
00473  case 2:
00474      #pragma omp parallel for simd \
00475      private(ii, target) \
00476      shared(dPD, gLW, totalNP, uTox, uAux, buffOut) \
00477      schedule(static)
00478      for (sunindextype i = 0; i < totalNP; i++) {
00479          target = dPD * i;
00480          ii = dPD * (uToy[i] - gLW);
00481          buffOut[target + 0] = (uAux[ii] - uAux[2 + ii]) / 2.;
00482          buffOut[target + 1] = uAux[5 + ii];
00483          buffOut[target + 2] = (-uAux[1 + ii] + uAux[3 + ii]) / 2.;
00484          buffOut[target + 3] = (uAux[1 + ii] + uAux[3 + ii]) / 2.;
00485          buffOut[target + 4] = uAux[4 + ii];
00486          buffOut[target + 5] = (uAux[ii] + uAux[2 + ii]) / 2.;
00487      }
00488      break;
00489  case 3:
00490      #pragma omp parallel for simd \
00491      private(ii, target) \
00492      shared(dPD, gLW, totalNP, uTox, uAux, buffOut) \
00493      schedule(static)
00494      for (sunindextype i = 0; i < totalNP; i++) {
00495          target = dPD * i;
00496          ii = dPD * (uToz[i] - gLW);
00497          buffOut[target + 0] = (-uAux[ii] + uAux[2 + ii]) / 2.;
00498          buffOut[target + 1] = (uAux[1 + ii] - uAux[3 + ii]) / 2.;
00499          buffOut[target + 2] = uAux[5 + ii];
00500          buffOut[target + 3] = (uAux[1 + ii] + uAux[3 + ii]) / 2.;
00501          buffOut[target + 4] = (uAux[ii] + uAux[2 + ii]) / 2.;
00502          buffOut[target + 5] = uAux[4 + ii];
00503      }
00504      break;
00505  default:
00506      errorKill("Tried to derotate from the wrong direction");
00507      break;
00508  }
00509 }
```

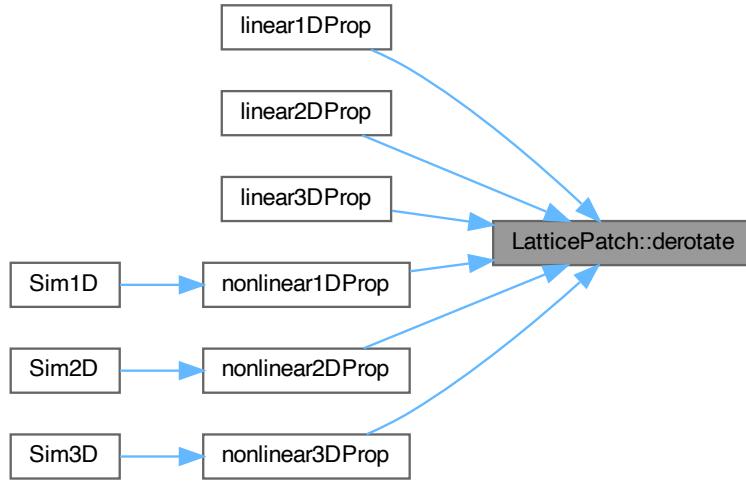
References [checkFlag\(\)](#), [discreteSize\(\)](#), [envelopeLattice](#), [errorKill\(\)](#), [FLatticePatchSetUp](#), [Lattice::get_dataPointDimension\(\)](#), [Lattice::get_ghostLayerWidth\(\)](#), [TranslocationLookupSetUp](#), [uAux](#), [uTox](#), [uToy](#), and [uToz](#).

Referenced by [linear1DProp\(\)](#), [linear2DProp\(\)](#), [linear3DProp\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), and [nonlinear3DProp\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.4 discreteSize()

```
sunindextype LatticePatch::discreteSize (
    int dir = 0 ) const
```

function to get the discrete size of the [LatticePatch](#)

Return the discrete size of the patch: number of lattice patch points in specified dimension

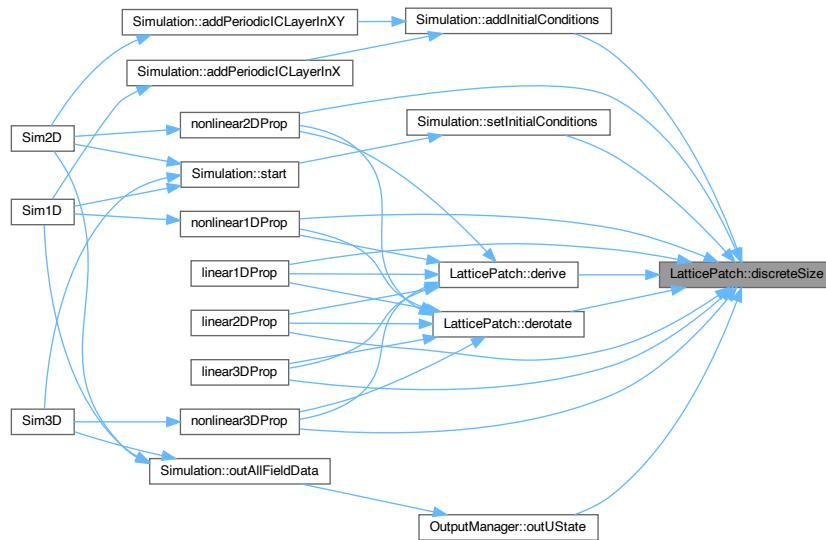
Definition at line 203 of file [LatticePatch.cpp](#).

```
00203
00204     switch (dir) {
00205         case 0:
00206             return nx * ny * nz;
00207         case 1:
00208             return nx;
00209         case 2:
00210             return ny;
00211         case 3:
00212             return nz;
00213         // case 4: return uAux.size(); // for debugging
00214     default:
00215         return -1;
00216     }
00217 }
```

References [nx](#), [ny](#), and [nz](#).

Referenced by [Simulation::addInitialConditions\(\)](#), [derive\(\)](#), [derotate\(\)](#), [linear1DProp\(\)](#), [linear2DProp\(\)](#), [linear3DProp\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), [nonlinear3DProp\(\)](#), [OutputManager::outUState\(\)](#), and [Simulation::setInitialConditions\(\)](#).

Here is the caller graph for this function:



5.9.3.5 exchangeGhostCells()

```
void LatticePatch::exchangeGhostCells (
    const int dir )
```

function to exchange ghost cells

Perform the ghost cell exchange in a specified direction.

Definition at line 527 of file [LatticePatch.cpp](#).

```
00527
00528 // Check that the lattice has been set up
00529 checkFlag(FLatticeDimensionSet);
00530 checkFlag(FLatticePatchSetUp);
00531 // Variables to per dimension calculate the halo indices, and distance to
00532 // other side halo boundary
00533 int mx = 1, my = 1, mz = 1, distToRight = 1;
00534 const int gLW = envelopeLattice->get_ghostLayerWidth();
00535 // In the chosen direction m is set to ghost layer width while the others
00536 // remain to form the plane
00537 switch (dir) {
00538 case 1:
00539     mx = gLW;
00540     my = ny;
00541     mz = nz;
00542     distToRight = (nx - gLW);
00543     break;
00544 case 2:
00545     mx = nx;
00546     my = gLW;
00547     mz = nz;
00548     distToRight = nx * (ny - gLW);
00549     break;
00550 case 3:
00551     mx = nx;
00552     my = ny;
00553     mz = gLW;
00554     distToRight = nx * ny * (nz - gLW);
00555     break;
00556 }
```

```

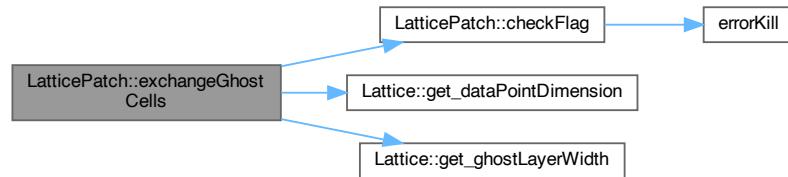
00557 // total number of exchanged points
00558 const int dPD = envelopeLattice->get_dataPointDimension();
00559 const sunindextype exchangeSize = mx * my * mz * dPD;
00560 // provide size of the halos for ghost cells
00561 ghostCellLeft.resize(exchangeSize);
00562 ghostCellRight.resize(ghostCellLeft.size());
00563 ghostCellLeftToSend.resize(ghostCellLeft.size());
00564 ghostCellRightToSend.resize(ghostCellLeft.size());
00565 gCLData = &ghostCellLeft[0];
00566 gCRData = &ghostCellRight[0];
00567 statusFlags |= GhostLayersInitialized;
00568
00569 // Initialize running index li for the halo buffers, and index ui of uData for
00570 // data transfer
00571 sunindextype li = 0, ui = 0;
00572 // Fill the halo buffers
00573 #pragma omp parallel for default(none) \
00574 private(ui, li) \
00575 shared(nx, ny, mx, my, mz, dPD, distToRight, uData, \
00576         ghostCellLeftToSend, ghostCellRightToSend)
00577 for (sunindextype iz = 0; iz < mz; iz++) {
00578     for (sunindextype iy = 0; iy < my; iy++) {
00579         // uData vector start index of halo data to be transferred
00580         // with each z-step add the whole xy-plane and with y-step the x-range ->
00581         // iterate all x-ranges
00582         ui = (iz * nx * ny + iy * nx) * dPD;
00583         // increase halo index by transferred items of previous iteration steps
00584         li = (iz * my * mx + iy * mx) * dPD;
00585         // copy left halo data from uData to buffer, transfer size is given by
00586         // x-length (not x-range)
00587         std::copy(&uData[ui], &uData[ui + mx * dPD], &ghostCellLeftToSend[li]);
00588         ui += distToRight * dPD;
00589         std::copy(&uData[ui], &uData[ui + mx * dPD], &ghostCellRightToSend[li]);
00590     }
00591 }
00592
00593 /* Send and receive the data to and from neighboring latticePatches */
00594 // Adjust direction to cartesian communicator
00595 int dim = 2; // default for dir==1
00596 if (dir == 2) {
00597     dim = 1;
00598 } else if (dir == 3) {
00599     dim = 0;
00600 }
00601 int rank_source = 0, rank_dest = 0;
00602 MPI_Cart_shift(envelopeLattice->comm, dim, -1, &rank_source,
00603                 &rank_dest); // s.t. rank_dest is left & v.v.
00604
00605 // nonblocking Irecv/Isend
00606
00607 MPI_Request requests[4];
00608 MPI_Irecv(&ghostCellRight[0], exchangeSize, MPI_SUNREALTYPE, rank_source, 1,
00609            envelopeLattice->comm, &requests[0]);
00610 MPI_Isend(&ghostCellLeftToSend[0], exchangeSize, MPI_SUNREALTYPE, rank_dest,
00611            1, envelopeLattice->comm, &requests[1]);
00612 MPI_Irecv(&ghostCellLeft[0], exchangeSize, MPI_SUNREALTYPE, rank_dest, 2,
00613            envelopeLattice->comm, &requests[2]);
00614 MPI_Isend(&ghostCellRightToSend[0], exchangeSize, MPI_SUNREALTYPE,
00615            rank_source, 2, envelopeLattice->comm, &requests[3]);
00616 MPI_Waitall(4, requests, MPI_STATUS_IGNORE);
00617
00618 // blocking Sendrecv:
00619 /*
00620 MPI_Sendrecv(&ghostCellLeftToSend[0], exchangeSize, MPI_SUNREALTYPE,
00621             rank_dest, 1, &ghostCellRight[0], exchangeSize, MPI_SUNREALTYPE,
00622             rank_source, 1, envelopeLattice->comm, MPI_STATUS_IGNORE);
00623 MPI_Sendrecv(&ghostCellRightToSend[0], exchangeSize, MPI_SUNREALTYPE,
00624             rank_source, 2, &ghostCellLeft[0], exchangeSize, MPI_SUNREALTYPE,
00625             rank_dest, 2, envelopeLattice->comm, MPI_STATUS_IGNORE);
00626 */
00627
00628 }

```

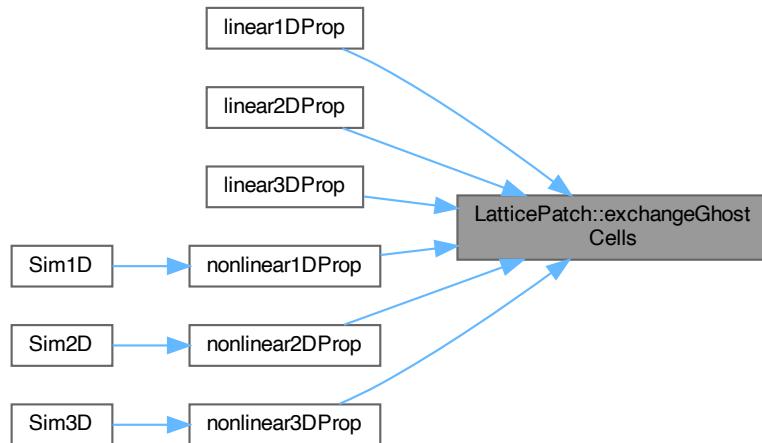
References [checkFlag\(\)](#), [Lattice::comm](#), [envelopeLattice](#), [FLatticeDimensionSet](#), [FLatticePatchSetUp](#), [gCLData](#), [gCRData](#), [Lattice::get_dataPointDimension\(\)](#), [Lattice::get_ghostLayerWidth\(\)](#), [ghostCellLeft](#), [ghostCellLeftToSend](#), [ghostCellRight](#), [ghostCellRightToSend](#), [GhostLayersInitialized](#), [nx](#), [ny](#), [nz](#), [statusFlags](#), and [uData](#).

Referenced by [linear1DProp\(\)](#), [linear2DProp\(\)](#), [linear3DProp\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), and [nonlinear3DProp\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.6 generateTranslocationLookup()

```
void LatticePatch::generateTranslocationLookup( )
```

function to fill out the lookup tables for cache efficiency

In order to avoid cache misses: create vectors to translate u vector into space coordinates and vice versa and same for left and right ghost layers to space

Definition at line 253 of file [LatticePatch.cpp](#).

```

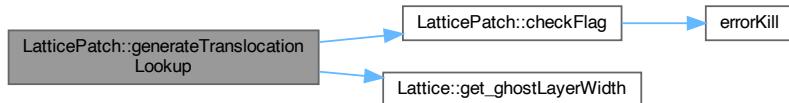
00253
00254 // Check that the lattice has been set up
00255 checkFlag(FLatticeDimensionSet);
00256 // lengths for auxilliary layers, including ghost layers
00257 const int gLW = envelopeLattice->get_ghostLayerWidth();
00258 const sunindextype mx = nx + 2 * gLW;
00259 const sunindextype my = ny + 2 * gLW;
00260 const sunindextype mz = nz + 2 * gLW;
00261 // sizes for lookup vectors
  
```

```

00262 const sunindextype totalNP = nx * ny * nz;
00263 const sunindextype haloXSize = mx * ny * nz;
00264 const sunindextype haloYSize = nx * my * nz;
00265 const sunindextype haloZSize = nx * ny * mz;
00266 // generate u->uAux
00267 uTox.resize(totalNP);
00268 uToy.resize(totalNP);
00269 uToz.resize(totalNP);
00270 // generate uAux->u with length including halo
00271 xTou.resize(haloXSize);
00272 yTou.resize(haloYSize);
00273 zTou.resize(haloZSize);
00274 // same for ghost layer lookup tables
00275 const sunindextype ghostXSize = gLW * ny * nz;
00276 const sunindextype ghostYSize = gLW * nx * nz;
00277 const sunindextype ghostZSize = gLW * nx * ny;
00278 lgcTox.resize(ghostXSize);
00279 rgcTox.resize(ghostXSize);
00280 lgcToy.resize(ghostYSize);
00281 rgcToy.resize(ghostYSize);
00282 lgcToz.resize(ghostZSize);
00283 rgcToz.resize(ghostZSize);
00284 // variables for cartesian position in the 3D discrete lattice
00285 sunindextype px = 0, py = 0, pz = 0;
00286 // Fill the lookup tables
00287 #pragma omp parallel default(none) \
00288 private(px, py, pz) \
00289 shared(uTox, uToy, uToz, xTou, yTou, zTou, \
00290         nx, ny, mx, my, mz, gLW, totalNP, \
00291         lgcTox, rgcTox, lgcToy, rgcToy, lgcToz, rgcToz, \
00292         ghostXSize, ghostYSize, ghostZSize)
00293 {
00294 #pragma omp for simd schedule(static)
00295 for (sunindextype i = 0; i < totalNP; i++) { // loop over the patch
00296     // calculate cartesian coordinates
00297     px = i % nx;
00298     py = (i / nx) % ny;
00299     pz = (i / nx) / ny;
00300     // fill lookups extended by halos (useful for y and z direction)
00301     uTox[i] = (px + gLW) + py * mx +
00302                 pz * mx * ny; // unroll (de-flatten) cartesian dimension
00303     xTou[px + py * mx + pz * mx * ny] =
00304         i; // match cartesian point to u location
00305     uToy[i] = (py + gLW) + pz * my + px * my * nz;
00306     yTou[py + pz * my + px * my * nz] = i;
00307     uToz[i] = (pz + gLW) + px * mz + py * mz * nx;
00308     zTou[pz + px * mz + py * mz * nx] = i;
00309 }
00310 #pragma omp for simd schedule(static)
00311 for (sunindextype i = 0; i < ghostXSize; i++) {
00312     px = i % gLW;
00313     py = (i / gLW) % ny;
00314     pz = (i / gLW) / ny;
00315     lgcTox[i] = px + py * mx + pz * mx * ny;
00316     rgcTox[i] = px + nx + gLW + py * mx + pz * mx * ny;
00317 }
00318 #pragma omp for simd schedule(static)
00319 for (sunindextype i = 0; i < ghostYSize; i++) {
00320     px = i % nx;
00321     py = (i / nx) % gLW;
00322     pz = (i / nx) / gLW;
00323     lgcToy[i] = py + pz * my + px * my * nz;
00324     rgcToy[i] = py + ny + gLW + pz * my + px * my * nz;
00325 }
00326 #pragma omp for simd schedule(static)
00327 for (sunindextype i = 0; i < ghostZSize; i++) {
00328     px = i % nx;
00329     py = (i / nx) % ny;
00330     pz = (i / nx) / ny;
00331     lgcToz[i] = pz + px * mz + py * mz * nx;
00332     rgcToz[i] = pz + nz + gLW + px * mz + py * mz * nx;
00333 }
00334 }
00335 statusFlags |= TranslocationLookupSetUp;
00336 }
```

References [checkFlag\(\)](#), [envelopeLattice](#), [FLatticeDimensionSet](#), [Lattice::get_ghostLayerWidth\(\)](#), [lgcTox](#), [lgcToy](#), [lgcToz](#), [nx](#), [ny](#), [nz](#), [rgcTox](#), [rgcToy](#), [rgcToz](#), [statusFlags](#), [TranslocationLookupSetUp](#), [uTox](#), [uToy](#), [uToz](#), [xTou](#), [yTou](#), and [zTou](#).

Here is the call graph for this function:



5.9.3.7 getDelta()

```
sunrealtype LatticePatch::getDelta (
    const int dir ) const
```

function to get distance between points

Return the distance between points in the patch in a dimension.

Definition at line 235 of file [LatticePatch.cpp](#).

```

00235
00236     switch (dir) {
00237         case 1:
00238             return dx;
00239         case 2:
00240             return dy;
00241         case 3:
00242             return dz;
00243     default:
00244         errorKill(
00245             "LatticePatch::getDelta function called with wrong dir parameter");
00246         return -1;
00247     }
00248 }
```

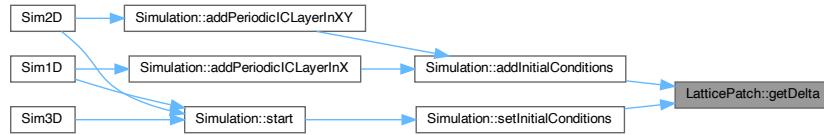
References [dx](#), [dy](#), [dz](#), and [errorKill\(\)](#).

Referenced by [Simulation::addInitialConditions\(\)](#), and [Simulation::setInitialConditions\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.8 initializeBuffers()

```
void LatticePatch::initializeBuffers ( )
```

initialize buffers to save derivatives

Create buffers to save derivative values, optimizing computational load.

Definition at line 512 of file [LatticePatch.cpp](#).

```
00512                                     {
00513     // Check that the lattice has been set up
00514     checkFlag(FLatticeDimensionSet);
00515     const int dPD = envelopeLattice->get_dataPointDimension();
00516     buffX.resize(nx * ny * nz * dPD);
00517     buffY.resize(nx * ny * nz * dPD);
00518     buffZ.resize(nx * ny * nz * dPD);
00519     // Set pointers used for propagation functions
00520     buffData[0] = &buffX[0];
00521     buffData[1] = &buffY[0];
00522     buffData[2] = &buffZ[0];
00523     statusFlags |= BuffersInitialized;
00524 }
```

References [buffData](#), [BuffersInitialized](#), [buffX](#), [buffY](#), [buffZ](#), [checkFlag\(\)](#), [envelopeLattice](#), [FLatticeDimensionSet](#), [Lattice::get_dataPointDimension\(\)](#), [nx](#), [ny](#), [nz](#), and [statusFlags](#).

Referenced by [Simulation::initializePatchwork\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.9 origin()

```
sunrealtype LatticePatch::origin (
    const int dir ) const
```

function to get the origin of the patch

Return the physical origin of the patch in a dimension.

Definition at line 220 of file [LatticePatch.cpp](#).

```

00220
00221     switch (dir) {
00222     case 1:
00223         return x0;
00224     case 2:
00225         return y0;
00226     case 3:
00227         return z0;
00228     default:
00229         errorKill("LatticePatch::origin function called with wrong dir parameter");
00230         return -1;
00231     }
00232 }
```

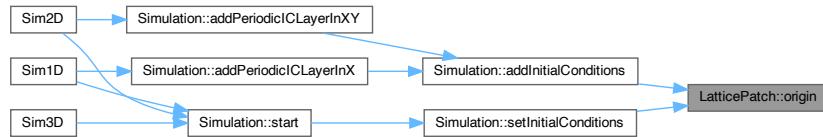
References [errorKill\(\)](#), [x0](#), [y0](#), and [z0](#).

Referenced by [Simulation::addInitialConditions\(\)](#), and [Simulation::setInitialConditions\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.10 rotateIntoEigen()

```
void LatticePatch::rotateIntoEigen (
    const int dir )
```

function to rotate u into Z-matrix eigenraum

Rotate into eigenraum along R matrices of paper using the rotation methods; uAuxData gets the rotated left-halo-, inner-patch-, right-halo-data

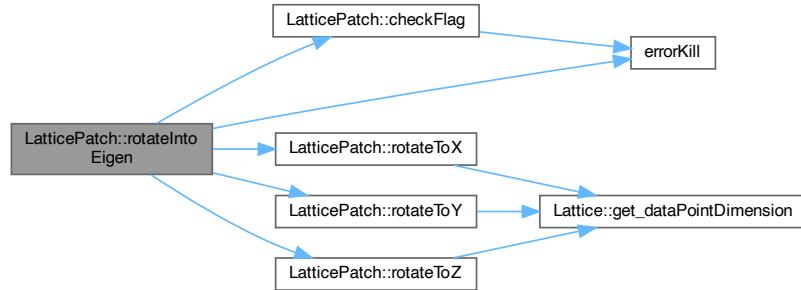
Definition at line 341 of file [LatticePatch.cpp](#).

```
00341
00342     // Check that the lattice, ghost layers as well as the translocation lookups
00343     // have been set up;
00344     checkFlag(FLatticePatchSetUp);
00345     checkFlag(TranslocationLookupSetUp);
00346     checkFlag(GhostLayersInitialized); // this check is only after call to
00347     // exchange ghost cells
00348     switch (dir) {
00349         case 1:
00350             rotateToX(uAuxData, gCLData, lgcTox);
00351             rotateToX(uAuxData, uData, uTox);
00352             rotateToX(uAuxData, gCRData, rgcTox);
00353             break;
00354         case 2:
00355             rotateToY(uAuxData, gCLData, lgcToy);
00356             rotateToY(uAuxData, uData, uToy);
00357             rotateToY(uAuxData, gCRData, rgcToy);
00358             break;
00359         case 3:
00360             rotateToZ(uAuxData, gCLData, lgcToz);
00361             rotateToZ(uAuxData, uData, uToz);
00362             rotateToZ(uAuxData, gCRData, rgcToz);
00363             break;
00364     default:
00365         errorKill("Tried to rotate into the wrong direction");
00366         break;
00367     }
00368 }
```

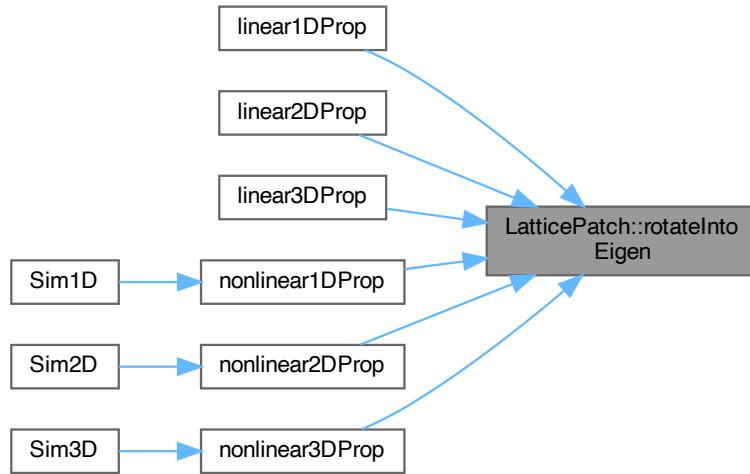
References [checkFlag\(\)](#), [errorKill\(\)](#), [FLatticePatchSetUp](#), [gCLData](#), [gCRData](#), [GhostLayersInitialized](#), [lgcTox](#), [lgcToy](#), [lgcToz](#), [rgcTox](#), [rgcToy](#), [rgcToz](#), [rotateToX\(\)](#), [rotateToY\(\)](#), [rotateToZ\(\)](#), [TranslocationLookupSetUp](#), [uAuxData](#), [uData](#), [uTox](#), [uToy](#), and [uToz](#).

Referenced by [linear1DProp\(\)](#), [linear2DProp\(\)](#), [linear3DProp\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), and [nonlinear3DProp\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.11 `rotateToX()`

```

void LatticePatch::rotateToX (
    sunrealtype * outArray,
    const sunrealtype * inArray,
    const std::vector< sunindextype > & lookup ) [inline], [private]
  
```

Rotate and translocate an input array according to a lookup into an output array

Rotate halo and inner-patch data vectors with rotation matrix Rx into eigenspace of Z matrix and write to auxiliary vector

Definition at line 372 of file [LatticePatch.cpp](#).

```

00374     sunindextype ii = 0, target = 0;
00375     const sunindextype size = lookup.size();
00376     const int dPD = envelopeLattice->get_dataPointDimension();
00377     #pragma omp parallel for simd \
00378     private(target, ii) \
00379     shared(lookup, outArray, inArray, size, dPD) \
00380     schedule(static)
00382     for (sunindextype i = 0; i < size; i++) {
00383         // get correct u-vector and spatial indices along previously defined lookup
00384         // tables
00385         target = dPD * lookup[i];
00386         ii = dPD * i;
00387         outArray[target + 0] = -inArray[1 + ii] + inArray[5 + ii];
00388         outArray[target + 1] = inArray[2 + ii] + inArray[4 + ii];
00389         outArray[target + 2] = inArray[1 + ii] + inArray[5 + ii];
00390         outArray[target + 3] = -inArray[2 + ii] + inArray[4 + ii];
00391         outArray[target + 4] = inArray[3 + ii];
00392         outArray[target + 5] = inArray[ii];
00393     }
00394 }
```

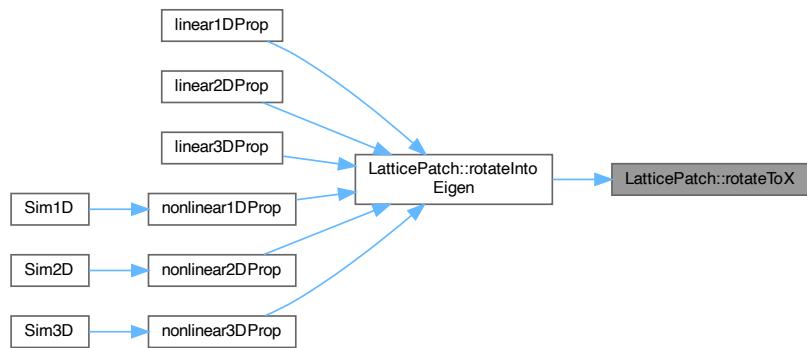
References [envelopeLattice](#), and [Lattice::get_dataPointDimension\(\)](#).

Referenced by [rotateIntoEigen\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.12 rotateToY()

```
void LatticePatch::rotateToY (
    sunrealtype * outArray,
    const sunrealtype * inArray,
    const std::vector< sunindextype > & lookup ) [inline], [private]
```

Rotate halo and inner-patch data vectors with rotation matrix Ry into eigenspace of Z matrix and write to auxiliary vector

Definition at line 398 of file [LatticePatch.cpp](#).

```
00400
00401     sunindextype ii = 0, target = 0;
00402     const int dPD = envelopeLattice->get_dataPointDimension();
00403     const sunindextype size = lookup.size();
00404     #pragma omp parallel for simd \
00405     private(target, ii) \
00406     shared(lookup, outArray, inArray, size, dPD) \
00407     schedule(static)
00408     for (sunindextype i = 0; i < size; i++) {
00409         target = dPD * lookup[i];
00410         ii = dPD * i;
00411         outArray[target + 0] = inArray[ii] + inArray[5 + ii];
00412         outArray[target + 1] = -inArray[2 + ii] + inArray[3 + ii];
00413         outArray[target + 2] = -inArray[ii] + inArray[5 + ii];
00414         outArray[target + 3] = inArray[2 + ii] + inArray[3 + ii];
00415         outArray[target + 4] = inArray[4 + ii];
00416         outArray[target + 5] = inArray[1 + ii];
00417     }
00418 }
```

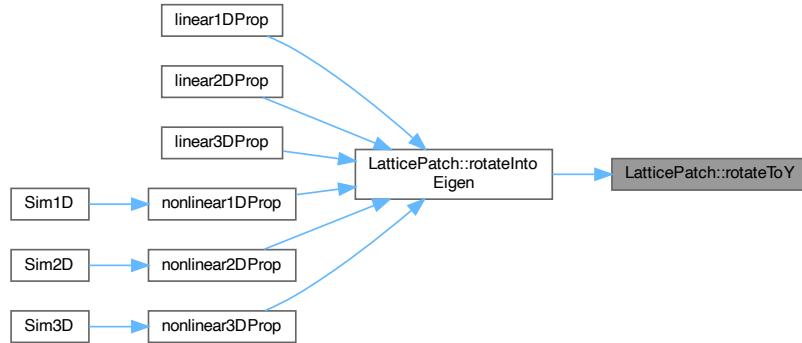
References [envelopeLattice](#), and [Lattice::get_dataPointDimension\(\)](#).

Referenced by [rotateIntoEigen\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.3.13 rotateToZ()

```
void LatticePatch::rotateToZ (
    sunrealtype * outArray,
    const sunrealtype * inArray,
    const std::vector< sunindextype > & lookup ) [inline], [private]
```

Rotate halo and inner-patch data vectors with rotation matrix Rz into eigenspace of Z matrix and write to auxiliary vector

Definition at line 422 of file [LatticePatch.cpp](#).

```
00424
00425     sunindextype ii = 0, target = 0;
00426     const sunindextype size = lookup.size();
00427     const int dPD = envelopeLattice->get_dataPointDimension();
00428     #pragma omp parallel for simd \
00429     private(target, ii) \
00430     shared(lookup, outArray, inArray, size, dPD) \
00431     schedule(static)
00432     for (sunindextype i = 0; i < size; i++) {
00433         target = dPD * lookup[i];
00434         ii = dPD * i;
00435         outArray[target + 0] = -inArray[ii] + inArray[4 + ii];
00436         outArray[target + 1] = inArray[1 + ii] + inArray[3 + ii];
00437         outArray[target + 2] = inArray[ii] + inArray[4 + ii];
00438         outArray[target + 3] = -inArray[1 + ii] + inArray[3 + ii];
00439         outArray[target + 4] = inArray[5 + ii];
00440         outArray[target + 5] = inArray[2 + ii];
00441     }
00442 }
```

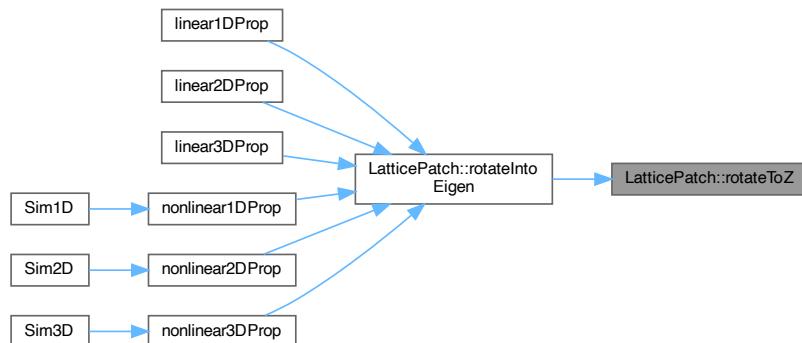
References [envelopeLattice](#), and [Lattice::get_dataPointDimension\(\)](#).

Referenced by [rotateIntoEigen\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.9.4 Friends And Related Function Documentation

5.9.4.1 generatePatchwork

```
int generatePatchwork (
    const Lattice & envelopeLattice,
    LatticePatch & patchToMold,
    const int DLx,
    const int DLy,
    const int DLz ) [friend]
```

friend function for creating the patchwork slicing of the overall lattice

Definition at line 109 of file [LatticePatch.cpp](#).

```
00111 // Retrieve the ghost layer depth
00112 const int gLW = envelopeLattice.get_ghostLayerWidth();
00113 // Retrieve the data point dimension
00114 const int dPD = envelopeLattice.get_dataPointDimension();
00115 // MPI process/patch
00116 const int my_prc = envelopeLattice.my_prc;
00117 // Determine thicknesses of the slice
00118 const sunindextype tot_NOXP = envelopeLattice.get_tot_nx();
00119 const sunindextype tot_NOYP = envelopeLattice.get_tot_ny();
00120 const sunindextype tot_NOZP = envelopeLattice.get_tot_nz();
00121 // position of the patch in the lattice of patches -> process associated to
00122 // position
00123 const sunindextype LIx = my_prc % DLx;
00124 const sunindextype LIy = (my_prc / DLx) % DLy;
00125 const sunindextype LIz = (my_prc / DLx) / DLy;
00126 // Determine the number of points in the patch and first absolute points in
00127 // each dimension
00128 const sunindextype local_NOXP = tot_NOXP / DLx;
00129 const sunindextype local_NOYP = tot_NOYP / DLy;
00130 const sunindextype local_NOZP = tot_NOZP / DLz;
00131 // absolute positions of the first point in each dimension
00132 const sunindextype firstXPoint = local_NOXP * LIx;
00133 const sunindextype firstYPoint = local_NOYP * LIy;
00134 const sunindextype firstZPoint = local_NOZP * LIz;
00135 // total number of points in the patch
00136 const sunindextype local_NODP = dPD * local_NOXP * local_NOYP * local_NOZP;
00137
00138 // Set patch up with above derived quantities
00139 patchToMold.dx = envelopeLattice.get_dx();
00140 patchToMold.dy = envelopeLattice.get_dy();
00141 patchToMold.dz = envelopeLattice.get_dz();
00142 patchToMold.x0 = firstXPoint * patchToMold.dx;
00143 patchToMold.y0 = firstYPoint * patchToMold.dy;
00144 patchToMold.z0 = firstZPoint * patchToMold.dz;
00145 patchToMold.LIx = LIx;
00146 patchToMold.LIy = LIy;
00147 patchToMold.LIz = LIz;
00148 patchToMold.nx = local_NOXP;
00149 patchToMold.ny = local_NOYP;
00150 patchToMold.nz = local_NOZP;
00151 patchToMold.lx = patchToMold.nx * patchToMold.dx;
00152 patchToMold.ly = patchToMold.ny * patchToMold.dy;
00153 patchToMold.lz = patchToMold.nz * patchToMold.dz;
00154
00155 #ifdef _OPENMP
00156 // OpenMP and MPI+X NVectors interoperability
00157 // OpenMP NVectors with local patch size
00158 // OpenMP NVectors containing local OpenMP NVectors
00159 int num_threads = 1;
00160 num_threads = omp_get_max_threads();
00161 patchToMold.ulocal = N_VNew_OpenMP(local_NODP, num_threads,
00162     envelopeLattice.sunctx);
00163 patchToMold.dulocal = N_VNew_OpenMP(local_NODP, num_threads,
00164     envelopeLattice.sunctx);
00165 // MPI+X NVectors containing local OpenMP NVectors
00166 patchToMold.u = N_VMake_MPIPlusX(envelopeLattice.comm, patchToMold.ulocal,
00167     envelopeLattice.sunctx);
00168 patchToMold.du = N_VMake_MPIPlusX(envelopeLattice.comm, patchToMold.dulocal,
00169     envelopeLattice.sunctx);
00170 // Pointers to local vectors
```

```

00171     patchToMold.uData = N_VGetArrayPointer_MPIPlusX(patchToMold.u);
00172     patchToMold.duData = N_VGetArrayPointer_MPIPlusX(patchToMold.du);
00173 #else
00174 // MPI NVectors with local patch and global lattice size
00175     patchToMold.u =
00176         N_VNew_Parallel(envelopeLattice.comm, local_NODP,
00177                         envelopeLattice.get_tot_noDP(), envelopeLattice.sunctx);
00178     patchToMold.du =
00179         N_VNew_Parallel(envelopeLattice.comm, local_NODP,
00180                         envelopeLattice.get_tot_noDP(), envelopeLattice.sunctx);
00181     patchToMold.uData = NV_DATA_P(patchToMold.u);
00182     patchToMold.duData = NV_DATA_P(patchToMold.du);
00183 #endif
00184
00185 // Allocate space for auxiliary uAux so that the lattice and all possible
00186 // directions of ghost Layers fit
00187 const sunindextype s1 = patchToMold.nx, s2 = patchToMold.ny,
00188     s3 = patchToMold.nz;
00189 const sunindextype s_min = std::min(s1, std::min(s2, s3));
00190 patchToMold.uAux.resize(s1 * s2 * s3 / s_min * (s_min + 2 * gLW) * dPD);
00191 patchToMold.uAuxData = &patchToMold.uAux[0];
00192 patchToMold.envelopeLattice = &envelopeLattice;
00193 // Set patch "name" to process number -> only for debugging
00194 // patchToMold.ID=my_prc;
00195 // set flag
00196 patchToMold.statusFlags = FLatticePatchSetUp;
00197 patchToMold.generateTranslocationLookup();
00198 return 0;
00199 }

```

5.9.5 Field Documentation

5.9.5.1 buffData

`std::array<sunrealtype *, 3> LatticePatch::buffData`

pointer to spatial derivative data buffers

Definition at line 213 of file [LatticePatch.h](#).

Referenced by [initializeBuffers\(\)](#), [linear1DProp\(\)](#), [linear2DProp\(\)](#), [linear3DProp\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), and [nonlinear3DProp\(\)](#).

5.9.5.2 buffX

`std::vector<sunrealtype> LatticePatch::buffX [private]`

buffer to save spatial derivative values

Definition at line 174 of file [LatticePatch.h](#).

Referenced by [initializeBuffers\(\)](#).

5.9.5.3 buffY

```
std::vector<sunrealtype> LatticePatch::buffY [private]
```

buffer to save spatial derivative values

Definition at line 174 of file [LatticePatch.h](#).

Referenced by [initializeBuffers\(\)](#).

5.9.5.4 buffZ

```
std::vector<sunrealtype> LatticePatch::buffZ [private]
```

buffer to save spatial derivative values

Definition at line 174 of file [LatticePatch.h](#).

Referenced by [initializeBuffers\(\)](#).

5.9.5.5 du

```
N_Vector LatticePatch::du
```

Definition at line 201 of file [LatticePatch.h](#).

Referenced by [~LatticePatch\(\)](#).

5.9.5.6 duData

```
sunrealtype* LatticePatch::duData
```

pointer to time-derivative data

Definition at line 205 of file [LatticePatch.h](#).

Referenced by [TimeEvolution::f\(\)](#), [linear1DProp\(\)](#), [linear2DProp\(\)](#), and [linear3DProp\(\)](#).

5.9.5.7 duLocal

```
N_Vector LatticePatch::duLocal
```

NVector for saving temporal derivatives of the field data.

Definition at line 201 of file [LatticePatch.h](#).

Referenced by [~LatticePatch\(\)](#).

5.9.5.8 dx

```
sunrealtype LatticePatch::dx [private]
```

physical distance between lattice points in x-direction

Definition at line 157 of file [LatticePatch.h](#).

Referenced by [derive\(\)](#), and [getDelta\(\)](#).

5.9.5.9 dy

```
sunrealtype LatticePatch::dy [private]
```

physical distance between lattice points in y-direction

Definition at line 159 of file [LatticePatch.h](#).

Referenced by [derive\(\)](#), and [getDelta\(\)](#).

5.9.5.10 dz

```
sunrealtype LatticePatch::dz [private]
```

physical distance between lattice points in z-direction

Definition at line 161 of file [LatticePatch.h](#).

Referenced by [derive\(\)](#), and [getDelta\(\)](#).

5.9.5.11 envelopeLattice

```
const Lattice* LatticePatch::envelopeLattice [private]
```

pointer to the enveloping lattice

Definition at line 165 of file [LatticePatch.h](#).

Referenced by [derive\(\)](#), [derotate\(\)](#), [exchangeGhostCells\(\)](#), [generateTranslocationLookup\(\)](#), [initializeBuffers\(\)](#), [rotateToX\(\)](#), [rotateToY\(\)](#), and [rotateToZ\(\)](#).

5.9.5.12 gCLData

```
sunrealtype* LatticePatch::gCLData
```

pointer to halo data

Definition at line 210 of file [LatticePatch.h](#).

Referenced by [exchangeGhostCells\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.13 gCRData

```
sunrealtype * LatticePatch::gCRData
```

pointer to halo data

Definition at line 210 of file [LatticePatch.h](#).

Referenced by [exchangeGhostCells\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.14 ghostCellLeft

```
std::vector<sunrealtype> LatticePatch::ghostCellLeft [private]
```

buffer for passing ghost cell data

Definition at line 178 of file [LatticePatch.h](#).

Referenced by [exchangeGhostCells\(\)](#).

5.9.5.15 ghostCellLeftToSend

```
std::vector<sunrealtype> LatticePatch::ghostCellLeftToSend [private]
```

buffer for passing ghost cell data

Definition at line 178 of file [LatticePatch.h](#).

Referenced by [exchangeGhostCells\(\)](#).

5.9.5.16 ghostCellRight

```
std::vector<sunrealtype> LatticePatch::ghostCellRight [private]
```

buffer for passing ghost cell data

Definition at line 178 of file [LatticePatch.h](#).

Referenced by [exchangeGhostCells\(\)](#).

5.9.5.17 ghostCellRightToSend

```
std::vector<sunrealtype> LatticePatch::ghostCellRightToSend [private]
```

buffer for passing ghost cell data

Definition at line 179 of file [LatticePatch.h](#).

Referenced by [exchangeGhostCells\(\)](#).

5.9.5.18 ghostCells

```
std::vector<sunrealtype> LatticePatch::ghostCells [private]
```

buffer for passing ghost cell data

Definition at line 179 of file [LatticePatch.h](#).

5.9.5.19 ghostCellsToSend

```
std::vector<sunrealtype> LatticePatch::ghostCellsToSend [private]
```

buffer for passing ghost cell data

Definition at line 179 of file [LatticePatch.h](#).

5.9.5.20 ID

```
int LatticePatch::ID
```

ID of the [LatticePatch](#), corresponds to process number (for debugging)

Definition at line 197 of file [LatticePatch.h](#).

5.9.5.21 lgcTox

```
std::vector<sunindextype> LatticePatch::lgcTox [private]
```

ghost cell translocation lookup table

Definition at line 183 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.22 lgcToy

```
std::vector<sunindextype> LatticePatch::lgcToy [private]
```

ghost cell translocation lookup table

Definition at line 183 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.23 lgcToz

```
std::vector<sunindextype> LatticePatch::lgcToz [private]
```

ghost cell translocation lookup table

Definition at line 183 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.24 LIx

```
sunindextype LatticePatch::LIx [private]
```

inner position of lattice-patch in the lattice patchwork; x-points

Definition at line 139 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#).

5.9.5.25 LIy

```
sunindextype LatticePatch::LIy [private]
```

inner position of lattice-patch in the lattice patchwork; y-points

Definition at line 141 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#).

5.9.5.26 LIz

```
sunindextype LatticePatch::LIz [private]
```

inner position of lattice-patch in the lattice patchwork; z-points

Definition at line 143 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#).

5.9.5.27 lx

```
sunrealtype LatticePatch::lx [private]
```

physical size of the lattice-patch in the x-dimension

Definition at line 145 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#).

5.9.5.28 ly

```
sunrealtype LatticePatch::ly [private]
```

physical size of the lattice-patch in the y-dimension

Definition at line 147 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#).

5.9.5.29 lz

```
sunrealtype LatticePatch::lz [private]
```

physical size of the lattice-patch in the z-dimension

Definition at line 149 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#).

5.9.5.30 nx

```
sunindextype LatticePatch::nx [private]
```

number of points in the lattice patch in the x-dimension

Definition at line 151 of file [LatticePatch.h](#).

Referenced by [discreteSize\(\)](#), [exchangeGhostCells\(\)](#), [generateTranslocationLookup\(\)](#), [initializeBuffers\(\)](#), and [LatticePatch\(\)](#).

5.9.5.31 ny

```
sunindextype LatticePatch::ny [private]
```

number of points in the lattice patch in the y-dimension

Definition at line 153 of file [LatticePatch.h](#).

Referenced by [discreteSize\(\)](#), [exchangeGhostCells\(\)](#), [generateTranslocationLookup\(\)](#), [initializeBuffers\(\)](#), and [LatticePatch\(\)](#).

5.9.5.32 nz

```
sunindextype LatticePatch::nz [private]
```

number of points in the lattice patch in the z-dimension

Definition at line 155 of file [LatticePatch.h](#).

Referenced by [discreteSize\(\)](#), [exchangeGhostCells\(\)](#), [generateTranslocationLookup\(\)](#), [initializeBuffers\(\)](#), and [LatticePatch\(\)](#).

5.9.5.33 rgcTox

```
std::vector<sunindextype> LatticePatch::rgcTox [private]
```

ghost cell translocation lookup table

Definition at line 183 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.34 rgcToy

```
std::vector<sunindextype> LatticePatch::rgcToy [private]
```

ghost cell translocation lookup table

Definition at line 183 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.35 rgcToz

```
std::vector<sunindextype> LatticePatch::rgcToz [private]
```

ghost cell translocation lookup table

Definition at line 183 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.36 statusFlags

```
unsigned int LatticePatch::statusFlags [private]
```

lattice patch status flags

Definition at line 163 of file [LatticePatch.h](#).

Referenced by [checkFlag\(\)](#), [exchangeGhostCells\(\)](#), [generateTranslocationLookup\(\)](#), [initializeBuffers\(\)](#), [LatticePatch\(\)](#), and [~LatticePatch\(\)](#).

5.9.5.37 u

```
N_Vector LatticePatch::u
```

Definition at line 199 of file [LatticePatch.h](#).

Referenced by [Simulation::advanceToTime\(\)](#), [Simulation::initializeCVODEobject\(\)](#), and [~LatticePatch\(\)](#).

5.9.5.38 uAux

```
std::vector<sunrealtype> LatticePatch::uAux [private]
```

aid (auxilliarly) vector including ghost cells to compute the derivatives

Definition at line 167 of file [LatticePatch.h](#).

Referenced by [derive\(\)](#), and [derotate\(\)](#).

5.9.5.39 uAuxData

```
sunrealtype* LatticePatch::uAuxData
```

pointer to auxiliary data vector

Definition at line 207 of file [LatticePatch.h](#).

Referenced by [rotateIntoEigen\(\)](#).

5.9.5.40 uData

```
sunrealtype* LatticePatch::uData
```

pointer to field data

Definition at line 203 of file [LatticePatch.h](#).

Referenced by [Simulation::addInitialConditions\(\)](#), [exchangeGhostCells\(\)](#), [TimeEvolution::f\(\)](#), [OutputManager::outUState\(\)](#), [rotateIntoEigen\(\)](#), and [Simulation::setInitialConditions\(\)](#).

5.9.5.41 uLocal

```
N_Vector LatticePatch::uLocal
```

NVector for saving field components $u=(E,B)$ in lattice points.

Definition at line 199 of file [LatticePatch.h](#).

Referenced by [~LatticePatch\(\)](#).

5.9.5.42 uTox

```
std::vector<sunindextype> LatticePatch::uTox [private]
```

translocation lookup table

Definition at line 170 of file [LatticePatch.h](#).

Referenced by [derotate\(\)](#), [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.43 uToy

```
std::vector<sunindextype> LatticePatch::uToy [private]
```

translocation lookup table

Definition at line 170 of file [LatticePatch.h](#).

Referenced by [derotate\(\)](#), [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.44 uToz

```
std::vector<sunindextype> LatticePatch::uToz [private]
```

translocation lookup table

Definition at line 170 of file [LatticePatch.h](#).

Referenced by [derotate\(\)](#), [generateTranslocationLookup\(\)](#), and [rotateIntoEigen\(\)](#).

5.9.5.45 x0

```
sunrealtype LatticePatch::x0 [private]
```

origin of the patch in physical space; x-coordinate

Definition at line 133 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#), and [origin\(\)](#).

5.9.5.46 xTou

```
std::vector<sunindextype> LatticePatch::xTou [private]
```

translocation lookup table

Definition at line 170 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#).

5.9.5.47 y0

```
sunrealtype LatticePatch::y0 [private]
```

origin of the patch in physical space; y-coordinate

Definition at line 135 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#), and [origin\(\)](#).

5.9.5.48 yTou

```
std::vector<sunindextype> LatticePatch::yTou [private]
```

translocation lookup table

Definition at line 170 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#).

5.9.5.49 z0

```
sunrealtype LatticePatch::z0 [private]
```

origin of the patch in physical space; z-coordinate

Definition at line 137 of file [LatticePatch.h](#).

Referenced by [LatticePatch\(\)](#), and [origin\(\)](#).

5.9.5.50 zTou

```
std::vector<sunindextype> LatticePatch::zTou [private]
```

translocation lookup table

Definition at line 170 of file [LatticePatch.h](#).

Referenced by [generateTranslocationLookup\(\)](#).

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

- src/[LatticePatch.h](#)
- src/[LatticePatch.cpp](#)

5.10 OutputManager Class Reference

Output Manager class to generate and coordinate output writing to disk.

```
#include <src/Outputters.h>
```

Public Member Functions

- `OutputManager ()`
default constructor
- `void generateOutputFolder (const std::string &dir)`
function that creates folder to save simulation data
- `void set_outputStyle (const char _outputStyle)`
set the output style
- `void outUState (const int &state, const Lattice &lattice, const LatticePatch &latticePatch)`
function to write data to disk in specified way
- `const std::string & getSimCode () const`
simCode getter function

Static Private Member Functions

- `static std::string SimCodeGenerator ()`
function to create the Code of the Simulations

Private Attributes

- `std::string simCode`
variable to save the SimCode generated at execution
- `std::string Path`
variable for the path to the output folder
- `char outputStyle`
output style; csv or binary

5.10.1 Detailed Description

Output Manager class to generate and coordinate output writing to disk.

Definition at line 21 of file [Outputters.h](#).

5.10.2 Constructor & Destructor Documentation

5.10.2.1 OutputManager()

```
OutputManager::OutputManager ( )
```

default constructor

Directly generate the simCode at construction.

Definition at line 12 of file [Outputters.cpp](#).

```
00012     {
00013     simCode = SimCodeGenerator();
00014     outputStyle = 'c';
00015 }
```

References [outputStyle](#), [simCode](#), and [SimCodeGenerator\(\)](#).

Here is the call graph for this function:



5.10.3 Member Function Documentation

5.10.3.1 generateOutputFolder()

```
void OutputManager::generateOutputFolder (
    const std::string & dir)
```

function that creates folder to save simulation data

Generate the folder to save the data to by one process: In the given directory it creates a direcory "SimResults" and a directory with the simCode. The relevant part of the main file is written to a "config.txt" file in that directory to log the settings.

Definition at line 47 of file [Outputters.cpp](#).

```
00047
00048 // Do this only once for the first process
00049 int myPrc;
00050 MPI_Comm_rank(MPI_COMM_WORLD, &myPrc);
00051 if (myPrc == 0) {
00052     if (!fs::is_directory(dir))
00053         fs::create_directory(dir);
00054     if (!fs::is_directory(dir + "/SimResults"))
00055         fs::create_directory(dir + "/SimResults");
00056     if (!fs::is_directory(dir + "/SimResults/" + simCode))
00057         fs::create_directory(dir + "/SimResults/" + simCode);
00058 }
00059 // path variable for the output generation
00060 Path = dir + "/SimResults/" + simCode + "/";
00061
00062 // Logging configurations from main.cpp
00063 std::ifstream fin("main.cpp");
00064 std::ofstream fout(Path + "config.txt");
00065 std::string line;
00066 int begin=1000;
```

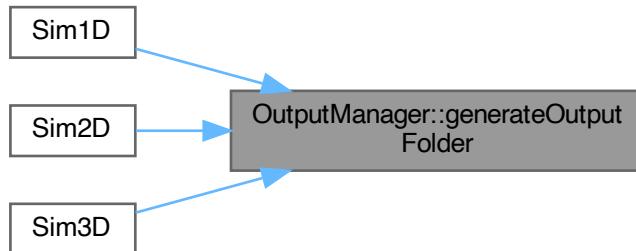
```

00067   for (int i = 1; !fin.eof(); i++) {
00068     getline(fin, line);
00069     if (line.starts_with("      //----- B")) {
00070       begin=i;
00071     }
00072     if (i < begin) {
00073       continue;
00074     }
00075     fout << line << std::endl;
00076     if (line.starts_with("      //----- E")) {
00077       break;
00078     }
00079   }
00080   return;
00081 }
```

References [Path](#), and [simCode](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the caller graph for this function:



5.10.3.2 getSimCode()

```
const std::string & OutputManager::getSimCode() const [inline]
```

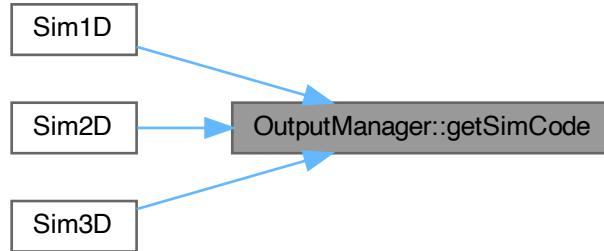
[simCode](#) getter function

Definition at line 42 of file [Outputters.h](#).
00042 { return simCode; }

References [simCode](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the caller graph for this function:



5.10.3.3 outUState()

```
void OutputManager::outUState (
    const int & state,
    const Lattice & lattice,
    const LatticePatch & latticePatch )
```

function to write data to disk in specified way

Write the field data either in csv format to one file per each process (patch) or in binary form to a single file. Files are stores inthe simCode directory. For csv files the state (simulation step) denotes the prefix and the suffix after an underscore is given by the process/patch number. Binary files are simply named after the step number.

Definition at line 92 of file [Outputters.cpp](#).

```
00093     {
00094     switch(outputStyle) {
00095         case 'c': { // one csv file per process
00096             std::ofstream ofs;
00097             ofs.open(Path + std::to_string(state) + "_"
00098                     + std::to_string(lattice.my_prc) + ".csv");
00099             // Precision of sunrealtype in significant decimal digits; 15 for IEEE double
00100             ofs << std::setprecision(std::numeric_limits<sunrealtype>::digits10);
00101
00102             // Walk through each lattice point
00103             const sunindextype totalNP = latticePatch.discreteSize();
00104             for (sunindextype i = 0; i < totalNP * 6; i += 6) {
00105                 // Six columns to contain the field data: Ex,Ey,Ez,Bx,By,Bz
00106                 ofs << latticePatch.uData[i + 0] << ","
00107                 << latticePatch.uData[i + 2] << ","
00108                 << latticePatch.uData[i + 4] << ","
00109                 << std::endl;
00110             }
00111             ofs.close();
00112             break;
00113         }
00114
00115         case 'b': { // a single binary file
00116             // Open the output file
00117             MPI_File fh;
00118             const std::string filename = Path+std::to_string(state);
00119             MPI_File_open(lattice.comm,&filename[0],MPI_MODE_WRONLY|MPI_MODE_CREATE,
00120                         MPI_INFO_NULL,&fh);
00121             // number of datapoints in the patch with process offset
00122             const sunindextype count = latticePatch.discreteSize() *
00123                 lattice.get_dataPointDimension();
00124             MPI_Offset offset = lattice.my_prc*count*sizeof(MPI_SUNREALTYPE);
```

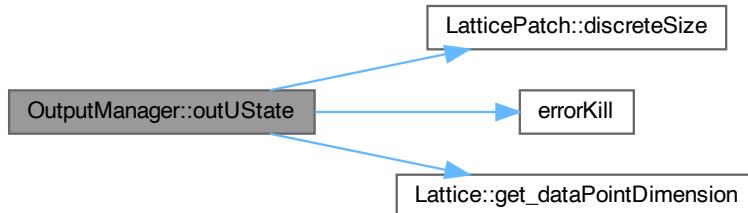
```

00125 // Go to offset in file and write data to it; maximal precision in
00126 // "native" representation
00127 MPI_File_set_view(fh,offset,MPI_SUNREALTYPE,MPI_SUNREALTYPE,"native",
00128     MPI_INFO_NULL);
00129 MPI_Request write_request;
00130 MPI_File_iwrite_all(fh,latticePatch.uData,count,MPI_SUNREALTYPE,
00131     &write_request);
00132 MPI_Wait(&write_request,MPI_STATUS_IGNORE);
00133 MPI_File_close(&fh);
00134 break;
00135 }
00136 default: {
00137 errorKill("No valid output style defined."
00138     " Choose between (c): one csv file per process,"
00139     " (b) one binary file");
00140 break;
00141 }
00142 }
00143 }
```

References [Lattice::comm](#), [LatticePatch::discreteSize\(\)](#), [errorKill\(\)](#), [Lattice::get_dataPointDimension\(\)](#), [Lattice::my_prc](#), [outputStyle](#), [Path](#), and [LatticePatch::uData](#).

Referenced by [Simulation::outAllFieldData\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.10.3.4 set_outputStyle()

```
void OutputManager::set_outputStyle (
    const char _outputStyle )
```

set the output style

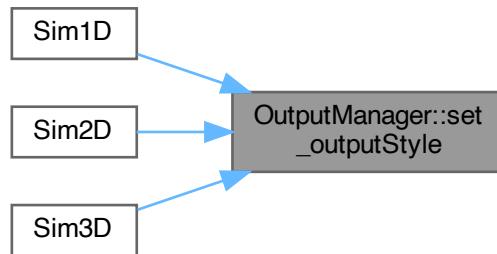
Definition at line 83 of file [Outputters.cpp](#).

```
00083
00084     outputStyle = _outputStyle;
00085 }
```

References [outputStyle](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the caller graph for this function:



5.10.3.5 SimCodeGenerator()

```
std::string OutputManager::SimCodeGenerator ( ) [static], [private]
```

function to create the Code of the Simulations

Generate the identifier number reverse from year to minute in the format yy-mm-dd_hh-MM-ss

Definition at line 19 of file [Outputters.cpp](#).

```
00019
00020     const chrono::time_point<chrono::system_clock> now{
00021         chrono::system_clock::now();
00022     const chrono::year_month_day ymd{chrono::floor<chrono::days>(now)};
00023     const auto tod = now - chrono::floor<chrono::days>(now);
00024     const chrono::hh_mm_ss hms{tod};
00025
00026     std::stringstream temp;
00027     temp << std::setfill('0') << std::setw(2)
00028         << static_cast<int>(ymd.year() - chrono::years(2000)) << "-"
00029         << std::setfill('0') << std::setw(2)
00030         << static_cast<unsigned>(ymd.month()) << "-"
00031         << std::setfill('0') << std::setw(2)
00032         << static_cast<unsigned>(ymd.day()) << "-"
00033         << std::setfill('0') << std::setw(2) << hms.hours().count()
00034         << "-" << std::setfill('0')
00035         << std::setw(2) << hms.minutes().count() << "-"
```

```

00036     << std::setfill('0') << std::setw(2)
00037     << hms.seconds().count();
00038     //<< "_" << hms.subseconds().count(); // subseconds render the filename
00039     // too large
00040     return temp.str();
00041 }
```

Referenced by [OutputManager\(\)](#).

Here is the caller graph for this function:



5.10.4 Field Documentation

5.10.4.1 outputStyle

```
char OutputManager::outputStyle [private]
```

output style; csv or binary

Definition at line 30 of file [Outputters.h](#).

Referenced by [OutputManager\(\)](#), [outUState\(\)](#), and [set_outputStyle\(\)](#).

5.10.4.2 Path

```
std::string OutputManager::Path [private]
```

variable for the path to the output folder

Definition at line 28 of file [Outputters.h](#).

Referenced by [generateOutputFolder\(\)](#), and [outUState\(\)](#).

5.10.4.3 simCode

`std::string OutputManager::simCode [private]`

variable to save the SimCode generated at execution

Definition at line 26 of file [Outputters.h](#).

Referenced by [generateOutputFolder\(\)](#), [getSimCode\(\)](#), and [OutputManager\(\)](#).

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

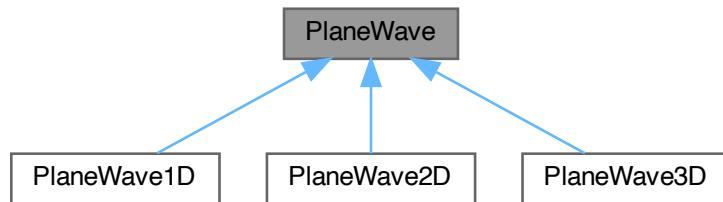
- [src/Outputters.h](#)
- [src/Outputters.cpp](#)

5.11 PlaneWave Class Reference

super-class for plane waves

```
#include <src/ICSetters.h>
```

Inheritance diagram for PlaneWave:



Protected Attributes

- sunrealtype `kx`
wavenumber k_x
- sunrealtype `ky`
wavenumber k_y
- sunrealtype `kz`
wavenumber k_z
- sunrealtype `px`
polarization & amplitude in x-direction, p_x
- sunrealtype `py`
polarization & amplitude in y-direction, p_y
- sunrealtype `pz`
polarization & amplitude in z-direction, p_z
- sunrealtype `phix`
phase shift in x-direction, ϕ_x
- sunrealtype `phiy`
phase shift in y-direction, ϕ_y
- sunrealtype `phiz`
phase shift in z-direction, ϕ_z

5.11.1 Detailed Description

super-class for plane waves

They are given in the form $\vec{E} = \vec{E}_0 \cos(\vec{k} \cdot \vec{x} - \phi)$

Definition at line 20 of file [ICSetters.h](#).

5.11.2 Field Documentation

5.11.2.1 kx

```
sunrealtype PlaneWave::kx [protected]
```

wavenumber k_x

Definition at line 23 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.2 ky

```
sunrealtype PlaneWave::ky [protected]
```

wavenumber k_y

Definition at line 25 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.3 kz

```
sunrealtype PlaneWave::kz [protected]
```

wavenumber k_z

Definition at line 27 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.4 phix

```
sunrealtype PlaneWave::phix [protected]
```

phase shift in x-direction, ϕ_x

Definition at line 35 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.5 phiy

```
sunrealtype PlaneWave::phiy [protected]
```

phase shift in y-direction, ϕ_y

Definition at line 37 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.6 phiz

```
sunrealtype PlaneWave::phiz [protected]
```

phase shift in z-direction, ϕ_z

Definition at line 39 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.7 px

```
sunrealtype PlaneWave::px [protected]
```

polarization & amplitude in x-direction, p_x

Definition at line 29 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.8 py

`sunrealtype PlaneWave::py [protected]`

polarization & amplitude in y-direction, p_y

Definition at line 31 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

5.11.2.9 pz

`sunrealtype PlaneWave::pz [protected]`

polarization & amplitude in z-direction, p_z

Definition at line 33 of file [ICSetters.h](#).

Referenced by [PlaneWave1D::addToSpace\(\)](#), [PlaneWave2D::addToSpace\(\)](#), [PlaneWave3D::addToSpace\(\)](#), [PlaneWave1D::PlaneWave1D\(\)](#), [PlaneWave2D::PlaneWave2D\(\)](#), and [PlaneWave3D::PlaneWave3D\(\)](#).

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

- `src/ICSetters.h`

5.12 planewave Struct Reference

plane wave structure

```
#include <src/SimulationFunctions.h>
```

Data Fields

- `std::array< unrealtype, 3 > k`
- `std::array< unrealtype, 3 > p`
- `std::array< unrealtype, 3 > phi`

5.12.1 Detailed Description

plane wave structure

Definition at line 19 of file [SimulationFunctions.h](#).

5.12.2 Field Documentation

5.12.2.1 k

```
std::array<sunrealtype, 3> planewave::k  
wavevector (normalized to  $1/\lambda$ )
```

Definition at line 20 of file [SimulationFunctions.h](#).

5.12.2.2 p

```
std::array<sunrealtype, 3> planewave::p  
amplitde & polarization vector
```

Definition at line 21 of file [SimulationFunctions.h](#).

5.12.2.3 phi

```
std::array<sunrealtype, 3> planewave::phi  
phase shift
```

Definition at line 22 of file [SimulationFunctions.h](#).

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

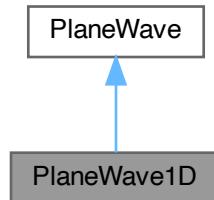
- src/[SimulationFunctions.h](#)

5.13 PlaneWave1D Class Reference

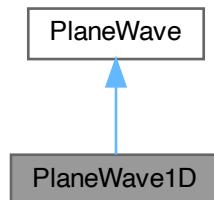
class for plane waves in 1D

```
#include <src/ICSetters.h>
```

Inheritance diagram for PlaneWave1D:



Collaboration diagram for PlaneWave1D:



Public Member Functions

- `PlaneWave1D (std::array< sunrealtype, 3 > k={1, 0, 0}, std::array< sunrealtype, 3 > p={0, 0, 1}, std::array< sunrealtype, 3 > phi={0, 0, 0})`
construction with default parameters
- `void addToSpace (sunrealtype x, sunrealtype y, sunrealtype z, sunrealtype *pTo6Space) const`
function for the actual implementation in the lattice

Additional Inherited Members

5.13.1 Detailed Description

class for plane waves in 1D

Definition at line 43 of file [ICSetters.h](#).

5.13.2 Constructor & Destructor Documentation

5.13.2.1 PlaneWave1D()

```
PlaneWave1D::PlaneWave1D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
```

construction with default parameters

[PlaneWave1D](#) construction with

- wavevectors k_x
- k_y
- k_z normalized to $1/\lambda$
- amplitude (polarization) in x-direction p_x
- amplitude (polarization) in y-direction p_y
- amplitude (polarization) in z-direction p_z
- phase shift in x-direction ϕ_x
- phase shift in y-direction ϕ_y
- phase shift in z-direction ϕ_z

Definition at line 11 of file [ICSetters.cpp](#).

```
00013     {
00014     kx = k[0]; /* - wavevectors \f$ k_x \f$ */
00015     ky = k[1]; /* - \f$ k_y \f$ */
00016     kz = k[2]; /* - \f$ k_z \f$ normalized to \f$ 1/\lambda \f$ */
00017 // Amplitude bug: lower by factor 3
00018     px = p[0] / 3; /* - amplitude (polarization) in x-direction \f$ p_x \f$ */
00019     py = p[1] / 3; /* - amplitude (polarization) in y-direction \f$ p_y \f$ */
00020     pz = p[2] / 3; /* - amplitude (polarization) in z-direction \f$ p_z \f$ */
00021     phix = phi[0]; /* - phase shift in x-direction \f$ \phi_x \f$ */
00022     phiy = phi[1]; /* - phase shift in y-direction \f$ \phi_y \f$ */
00023     phiz = phi[2]; /* - phase shift in z-direction \f$ \phi_z \f$ */
00024 }
```

References [PlaneWave::kx](#), [PlaneWave::ky](#), [PlaneWave::kz](#), [PlaneWave::phix](#), [PlaneWave::phiy](#), [PlaneWave::phiz](#), [PlaneWave::px](#), [PlaneWave::py](#), and [PlaneWave::pz](#).

5.13.3 Member Function Documentation

5.13.3.1 addToSpace()

```
void PlaneWave1D::addToSpace (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space ) const
```

function for the actual implementation in the lattice

[PlaneWave1D](#) implementation in space

Definition at line 27 of file [ICSetters.cpp](#).

```
00029     {
00030     const sunrealtype wavelength =
00031         sqrt(kx * kx + ky * ky + kz * kz); /* \f$ 1/\lambda \f$ */
00032     const sunrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 *
00033         std::numbers::pi; /* \f$ 2\pi \cdot \vec{k} \cdot \vec{x} \f$ */
00034     // Plane wave definition
00035     const std::array<sunrealtype, 3> E{{
00036         px * cos(kScalarX - phix), /* \f$ E_x \f$ */
00037         py * cos(kScalarX - phiy), /* \f$ E_y \f$ */
00038         pz * cos(kScalarX - phiz)} }; /* \f$ E_z \f$ */
00039     // Put E-field into space
00040     pTo6Space[0] += E[0];
00041     pTo6Space[1] += E[1];
00042     pTo6Space[2] += E[2];
00043     // and B-field
00044     pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
00045     pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00046     pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00047 }
```

References [PlaneWave::kx](#), [PlaneWave::ky](#), [PlaneWave::kz](#), [PlaneWave::phix](#), [PlaneWave::phiy](#), [PlaneWave::phiz](#), [PlaneWave::px](#), [PlaneWave::py](#), and [PlaneWave::pz](#).

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

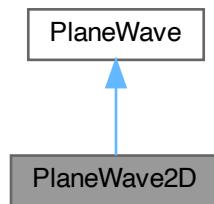
- [src/ICSetters.h](#)
- [src/ICSetters.cpp](#)

5.14 PlaneWave2D Class Reference

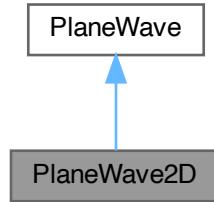
class for plane waves in 2D

```
#include <src/ICSetters.h>
```

Inheritance diagram for PlaneWave2D:



Collaboration diagram for PlaneWave2D:



Public Member Functions

- [PlaneWave2D](#) (std::array< sunrealtype, 3 > k={1, 0, 0}, std::array< sunrealtype, 3 > p={0, 0, 1}, std::array< sunrealtype, 3 > phi={0, 0, 0})
construction with default parameters
- void [addToSpace](#) (sunrealtype x, sunrealtype y, sunrealtype z, sunrealtype *pTo6Space) const
function for the actual implementation in the lattice

Additional Inherited Members

5.14.1 Detailed Description

class for plane waves in 2D

Definition at line 55 of file [ICSetters.h](#).

5.14.2 Constructor & Destructor Documentation

5.14.2.1 PlaneWave2D()

```

PlaneWave2D::PlaneWave2D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
  
```

construction with default parameters

[PlaneWave2D](#) construction with

- wavevectors k_x

- k_y
- k_z normalized to $1/\lambda$
- amplitude (polarization) in x-direction p_x
- amplitude (polarization) in y-direction p_y
- amplitude (polarization) in z-direction p_z
- phase shift in x-direction ϕ_x
- phase shift in y-direction ϕ_y
- phase shift in z-direction ϕ_z

Definition at line 50 of file [ICSetters.cpp](#).

```

00052   {
00053     kx = k[0]; /* - wavevectors k_x */
00054     ky = k[1]; /* - k_y */
00055     kz = k[2]; /* - k_z normalized to 1/\lambda */
00056 // Amplitude bug: lower by factor 9
00057     px = p[0] / 9; /* - amplitude (polarization) in x-direction p_x */
00058     py = p[1] / 9; /* - amplitude (polarization) in y-direction p_y */
00059     pz = p[2] / 9; /* - amplitude (polarization) in z-direction p_z */
00060     phix = phi[0]; /* - phase shift in x-direction \phi_x */
00061     phiy = phi[1]; /* - phase shift in y-direction \phi_y */
00062     phiz = phi[2]; /* - phase shift in z-direction \phi_z */
00063 }
```

References [PlaneWave::kx](#), [PlaneWave::ky](#), [PlaneWave::kz](#), [PlaneWave::phix](#), [PlaneWave::phiy](#), [PlaneWave::phiz](#), [PlaneWave::px](#), [PlaneWave::py](#), and [PlaneWave::pz](#).

5.14.3 Member Function Documentation

5.14.3.1 addToSpace()

```
void PlaneWave2D::addToSpace (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space ) const
```

function for the actual implementation in the lattice

[PlaneWave2D](#) implementation in space

Definition at line 66 of file [ICSetters.cpp](#).

```

00067   {
00068     const sunrealtype wavelength =
00069       sqrt(kx * kx + ky * ky + kz * kz); /* 1/\lambda */
00070     const sunrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 * 
00071       std::numbers::pi; /* 2\pi / |\vec{k}| \cdot \vec{e} \cdot \vec{x} */
00072 // Plane wave definition
00073     const std::array<sunrealtype, 3> E{{
00074       px * cos(kScalarX - phix), /* E_x */
00075       py * cos(kScalarX - phiy), /* E_y */
00076       pz * cos(kScalarX - phiz)} }; /* E_z */
00077 // Put E-field into space
00078     pTo6Space[0] += E[0];
00079     pTo6Space[1] += E[1];
00080     pTo6Space[2] += E[2];
00081 // and B-field
00082     pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
```

```
00083     pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00084     pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00085 }
```

References [PlaneWave::kx](#), [PlaneWave::ky](#), [PlaneWave::kz](#), [PlaneWave::phix](#), [PlaneWave::phiy](#), [PlaneWave::phiz](#), [PlaneWave::px](#), [PlaneWave::py](#), and [PlaneWave::pz](#).

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

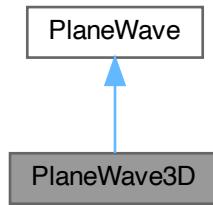
- [src/ICSetters.h](#)
- [src/ICSetters.cpp](#)

5.15 PlaneWave3D Class Reference

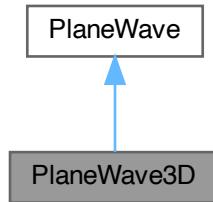
class for plane waves in 3D

```
#include <src/ICSetters.h>
```

Inheritance diagram for PlaneWave3D:



Collaboration diagram for PlaneWave3D:



Public Member Functions

- `PlaneWave3D` (`std::array< sunrealtype, 3 > k={1, 0, 0}, std::array< sunrealtype, 3 > p={0, 0, 1}, std::array< sunrealtype, 3 > phi={0, 0, 0})`
construction with default parameters
- void `addToSpace` (`sunrealtype x, sunrealtype y, sunrealtype z, sunrealtype *pTo6Space)` const
function for the actual implementation in space

Additional Inherited Members

5.15.1 Detailed Description

class for plane waves in 3D

Definition at line 67 of file [ICSetters.h](#).

5.15.2 Constructor & Destructor Documentation

5.15.2.1 `PlaneWave3D()`

```
PlaneWave3D::PlaneWave3D (
    std::array< sunrealtype, 3 > k = {1, 0, 0},
    std::array< sunrealtype, 3 > p = {0, 0, 1},
    std::array< sunrealtype, 3 > phi = {0, 0, 0} )
```

construction with default parameters

`PlaneWave3D` construction with

- wavevectors k_x
- k_y
- k_z normalized to $1/\lambda$
- amplitude (polarization) in x-direction p_x
- amplitude (polarization) in y-direction p_y
- amplitude (polarization) in z-direction p_z
- phase shift in x-direction ϕ_x
- phase shift in y-direction ϕ_y
- phase shift in z-direction ϕ_z

Definition at line 88 of file [ICSetters.cpp](#).

```
00090      {
00091     kx = k[0];    /** - wavevectors \f$ k_x \f$ */
00092     ky = k[1];    /** - \f$ k_y \f$ */
00093     kz = k[2];    /** - \f$ k_z \f$ normalized to \f$ 1/\lambda \f$ */
00094     px = p[0];    /** - amplitude (polarization) in x-direction \f$ p_x \f$ */
00095     py = p[1];    /** - amplitude (polarization) in y-direction \f$ p_y \f$ */
00096     pz = p[2];    /** - amplitude (polarization) in z-direction \f$ p_z \f$ */
00097     phix = phi[0]; /** - phase shift in x-direction \f$ \phi_{\text{x}} \f$ */
00098     phiy = phi[1]; /** - phase shift in y-direction \f$ \phi_{\text{y}} \f$ */
00099     phiz = phi[2]; /** - phase shift in z-direction \f$ \phi_{\text{z}} \f$ */
00100 }
```

References `PlaneWave::kx`, `PlaneWave::ky`, `PlaneWave::kz`, `PlaneWave::phix`, `PlaneWave::phiy`, `PlaneWave::phiz`, `PlaneWave::px`, `PlaneWave::py`, and `PlaneWave::pz`.

5.15.3 Member Function Documentation

5.15.3.1 addToSpace()

```
void PlaneWave3D::addToSpace (
    sunrealtype x,
    sunrealtype y,
    sunrealtype z,
    sunrealtype * pTo6Space ) const
```

function for the actual implementation in space

[PlaneWave3D](#) implementation in space

Definition at line 103 of file [ICSetters.cpp](#).

```
00104
00105     const sunrealtype wavelength =
00106         sqrt(kx * kx + ky * ky + kz * kz); /* \f$ 1/\lambda \f$ */
00107     const sunrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 *
00108         std::numbers::pi; /* \f$ 2\pi \vec{k} \cdot \vec{x} \f$ */
00109 // Plane wave definition
00110     const std::array<sunrealtype, 3> E{ /* E-field vector \f$ \vec{E} \f$ */
00111         px * cos(kScalarX - phix), /* \f$ E_x \f$ */
00112         py * cos(kScalarX - phiy), /* \f$ E_y \f$ */
00113         pz * cos(kScalarX - phiz)}; /* \f$ E_z \f$ */
00114 // Put E-field into space
00115     pTo6Space[0] += E[0];
00116     pTo6Space[1] += E[1];
00117     pTo6Space[2] += E[2];
00118 // and B-field
00119     pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
00120     pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00121     pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00122 }
```

References [PlaneWave::kx](#), [PlaneWave::ky](#), [PlaneWave::kz](#), [PlaneWave::phix](#), [PlaneWave::phiy](#), [PlaneWave::phiz](#), [PlaneWave::px](#), [PlaneWave::py](#), and [PlaneWave::pz](#).

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

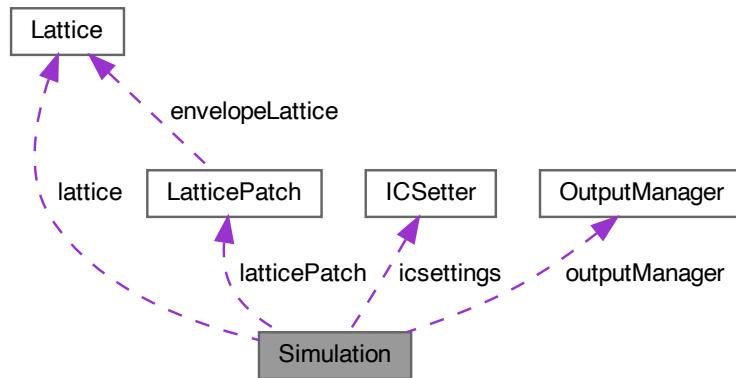
- [src/ICSetters.h](#)
- [src/ICSetters.cpp](#)

5.16 Simulation Class Reference

[Simulation](#) class to instantiate the whole walkthrough of a [Simulation](#).

```
#include <src/SimulationClass.h>
```

Collaboration diagram for Simulation:



Public Member Functions

- **Simulation** (const int Nx, const int Ny, const int Nz, const int StencilOrder, const bool periodicity)

constructor function for the creation of the cartesian communicator
- **~Simulation ()**

destructor function freeing CVode memory and Sundials context
- **MPI_Comm * get_cart_comm ()**

reference to the cartesian communicator of the lattice (for debugging)
- **void setDiscreteDimensionsOfLattice (const sunindextype _tot_nx, const sunindextype _tot_ny, const sunindextype _tot_nz)**

function to set discrete dimensions of the lattice
- **void setPhysicalDimensionsOfLattice (const sunrealtype lx, const sunrealtype ly, const sunrealtype lz)**

function to set physical dimensions of the lattice
- **void initializePatchwork (const int nx, const int ny, const int nz)**

function to initialize the Patchwork
- **void initializeCVODEobject (const sunrealtype reltol, const sunrealtype abstol)**

function to initialize the CVODE object with all requirements
- **void start ()**

function to start the simulation for time iteration
- **void setInitialConditions ()**

functions to set the initial field configuration onto the lattice
- **void addInitialConditions (const sunindextype xm, const sunindextype ym, const sunindextype zm=0)**

functions to add initial periodic field configurations
- **void addPeriodicICLayerInX ()**

function to add a periodic IC layer in one dimension
- **void addPeriodicICLayerInXY ()**

function to add periodic IC layers in two dimensions
- **void advanceToTime (const sunrealtype &tEnd)**

function to advance solution in time with CVODE
- **void outAllFieldData (const int &state)**

function to write field data to disk

- void `checkFlag` (unsigned int flag) const
function to check if flag has been set
- void `checkNoFlag` (unsigned int flag) const
function to check if flag has not been set

Data Fields

- `ICSetter icsettings`
IC Setter object.
- `OutputManager outputManager`
Output Manager object.
- `void * cvode_mem`
pointer to CVode memory object
- `SUNNonlinearSolver NLS`
nonlinear solver object

Private Attributes

- `Lattice lattice`
Lattice object.
- `LatticePatch latticePatch`
LatticePatch object.
- `sunrealtype t`
current time of the simulation
- `unsigned int statusFlags`
simulation status flags

5.16.1 Detailed Description

`Simulation` class to instantiate the whole walkthrough of a `Simulation`.

Definition at line 30 of file `SimulationClass.h`.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 Simulation()

```
Simulation::Simulation (
    const int Nx,
    const int Ny,
    const int Nz,
    const int StencilOrder,
    const bool periodicity )
```

constructor function for the creation of the cartesian communicator

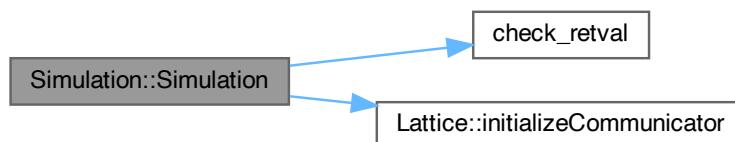
Along with the simulation object, create the cartesian communicator and SUNContext object

Definition at line 14 of file [SimulationClass.cpp](#).

```
00015 :
00016     lattice(StencilOrder) {
00017         statusFlags = 0;
00018         t = 0;
00019         // Initialize the cartesian communicator
00020         lattice.initializeCommunicator(Nx, Ny, Nz, periodicity);
00021
00022         // Create the SUNContext object associated with the thread of execution
00023         int retval = 0;
00024         retval = SUNContext_Create(&lattice.comm, &lattice.sunctx);
00025         if (check_retval(&retval, "SUNContext_Create", 1, lattice.my_prc))
00026             MPI_Abort(lattice.comm, 1);
00027 }
```

References [check_retval\(\)](#), [Lattice::comm](#), [Lattice::initializeCommunicator\(\)](#), [lattice](#), [Lattice::my_prc](#), [statusFlags](#), [Lattice::sunctx](#), and [t](#).

Here is the call graph for this function:



5.16.2.2 ~Simulation()

```
Simulation::~Simulation ( )
```

destructor function freeing CVode memory and Sundials context

Free the CVode solver memory and Sundials context object with the finish of the simulation

Definition at line 31 of file [SimulationClass.cpp](#).

```
00031     {
00032         // Free solver memory
00033         if (statusFlags & CvodeObjectSetUp) {
00034             CVodeFree(&cvode_mem);
00035             SUNNonlinSolFree(NLS);
00036             SUNContext_Free(&lattice.sunctx);
00037         }
00038     }
```

References [cvode_mem](#), [CvodeObjectSetUp](#), [lattice](#), [NLS](#), [statusFlags](#), and [Lattice::sunctx](#).

5.16.3 Member Function Documentation

5.16.3.1 addInitialConditions()

```
void Simulation::addInitialConditions (
    const sunindextype xm,
    const sunindextype ym,
    const sunindextype zm = 0 )
```

functions to add initial periodic field configurations

Use parameters to add periodic IC layers.

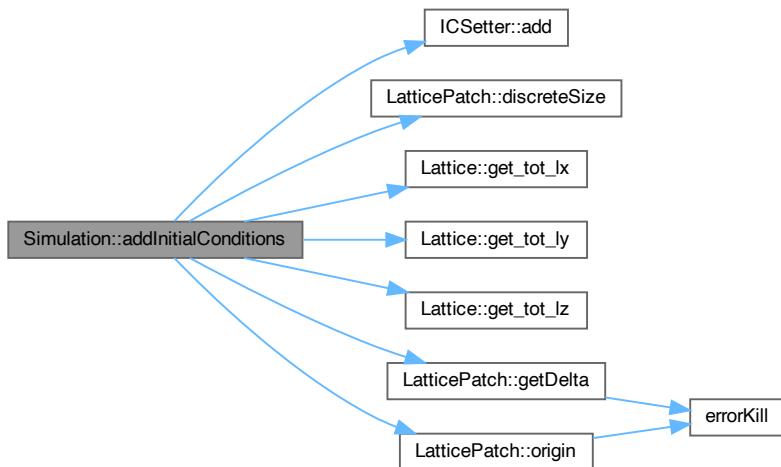
Definition at line 167 of file [SimulationClass.cpp](#).

```
00169     {
00170     const sunrealtype dx = latticePatch.getDelta(1);
00171     const sunrealtype dy = latticePatch.getDelta(2);
00172     const sunrealtype dz = latticePatch.getDelta(3);
00173     const sunindextype nx = latticePatch.discreteSize(1);
00174     const sunindextype ny = latticePatch.discreteSize(2);
00175     const sunindextype totalNP = latticePatch.discreteSize();
00176     // Correct for demanded displacement, rest as for setInitialConditions
00177     const sunrealtype x0 = latticePatch.origin(1) + xm*lattice.get_tot_lx();
00178     const sunrealtype y0 = latticePatch.origin(2) + ym*lattice.get_tot_ly();
00179     const sunrealtype z0 = latticePatch.origin(3) + zm*lattice.get_tot_lz();
00180     sunindextype px = 0, py = 0, pz = 0;
00181     for (sunindextype i = 0; i < totalNP * 6; i += 6) {
00182         px = (i / 6) % nx;
00183         py = ((i / 6) / nx) % ny;
00184         pz = ((i / 6) / nx) / ny;
00185         icsettings.add(static_cast<sunrealtype>(px) * dx + x0,
00186                         static_cast<sunrealtype>(py) * dy + y0,
00187                         static_cast<sunrealtype>(pz) * dz + z0, &latticePatch.uData[i]);
00188     }
00189     return;
00190 }
```

References [ICSetter::add\(\)](#), [LatticePatch::discreteSize\(\)](#), [Lattice::get_tot_lx\(\)](#), [Lattice::get_tot_ly\(\)](#), [Lattice::get_tot_lz\(\)](#), [LatticePatch::getDelta\(\)](#), [icsettings](#), [lattice](#), [latticePatch](#), [LatticePatch::origin\(\)](#), and [LatticePatch::uData](#).

Referenced by [addPeriodicICLayerInX\(\)](#), and [addPeriodicICLayerInXY\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.2 addPeriodicICLayerInX()

`void Simulation::addPeriodicICLayerInX ()`
function to add a periodic IC layer in one dimension

Add initial conditions in one dimension.

Definition at line 193 of file [SimulationClass.cpp](#).

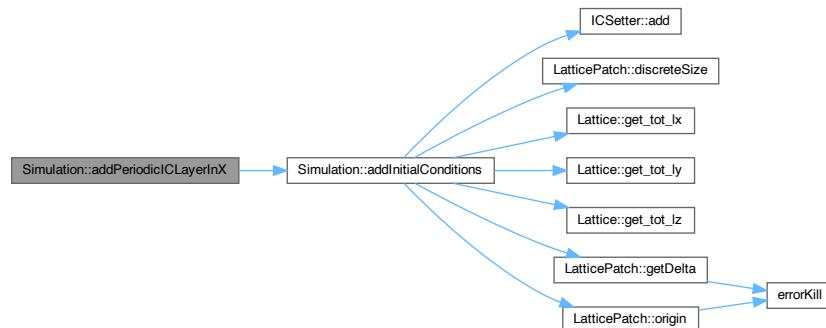
```

00193     {
00194     addInitialConditions(-1, 0, 0);
00195     addInitialConditions(1, 0, 0);
00196     return;
00197 }
```

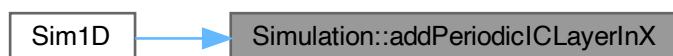
References [addInitialConditions\(\)](#).

Referenced by [Sim1D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.3 addPeriodicICLayerInXY()

```
void Simulation::addPeriodicICLayerInXY ( )
```

function to add periodic IC layers in two dimensions

Add initial conditions in two dimensions.

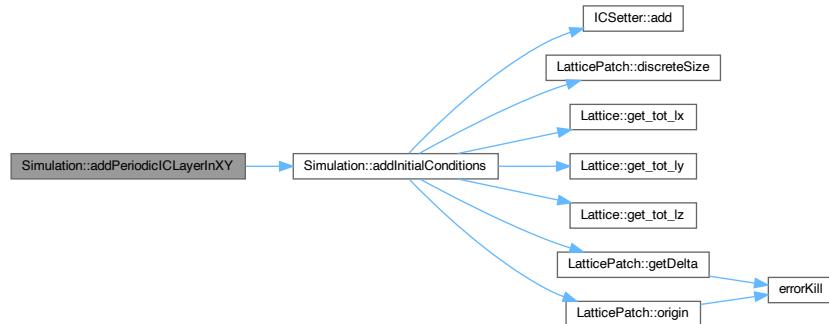
Definition at line 200 of file [SimulationClass.cpp](#).

```
00200
00201     addInitialConditions(-1, -1, 0);
00202     addInitialConditions(-1, 0, 0);
00203     addInitialConditions(-1, 1, 0);
00204     addInitialConditions(0, 1, 0);
00205     addInitialConditions(0, -1, 0);
00206     addInitialConditions(1, -1, 0);
00207     addInitialConditions(1, 0, 0);
00208     addInitialConditions(1, 1, 0);
00209     return;
00210 }
```

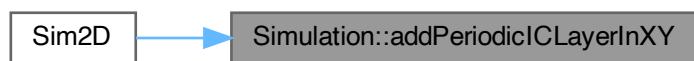
References [addInitialConditions\(\)](#).

Referenced by [Sim2D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.4 advanceToTime()

```
void Simulation::advanceToTime (
    const sunrealtype & tEnd )
```

function to advance solution in time with CVODE

Advance the solution in time -> integrate the ODE over an interval t.

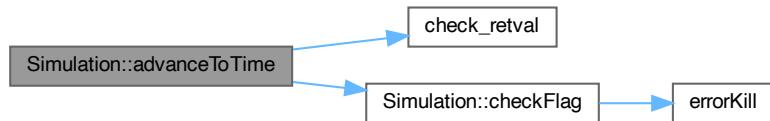
Definition at line 213 of file [SimulationClass.cpp](#).

```
00213
00214     checkFlag(SimulationStarted);
00215     int retval = 0;
00216     retval = CVode(cvode_mem, tEnd, latticePatch.u, &t,
00217                     CV_NORMAL); // CV_NORMAL: internal steps to reach tEnd, then
00218                     // interpolate to return latticePatch.u, return time
00219                     // reached by the solver as t
00220     if (check_retval(&retval, "CVode", 1, lattice.my_prc))
00221         MPI_Abort(lattice.comm, 1);
00222 }
```

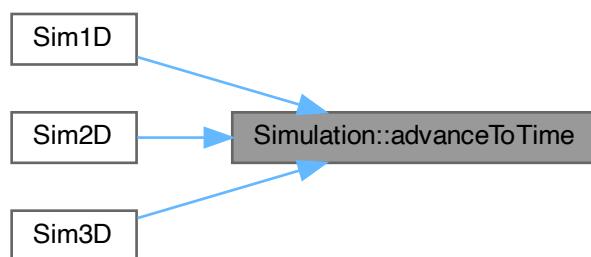
References [check_retval\(\)](#), [checkFlag\(\)](#), [Lattice::comm](#), [cvode_mem](#), [lattice](#), [latticePatch](#), [Lattice::my_prc](#), [SimulationStarted](#), [t](#), and [LatticePatch::u](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.5 checkFlag()

```
void Simulation::checkFlag (
    unsigned int flag ) const
```

function to check if flag has been set

Check presence of configuration flags.

Definition at line 231 of file [SimulationClass.cpp](#).

```
00231
00232     if (!(statusFlags & flag)) {
00233         std::string errorMessage;
00234         switch (flag) {
00235             case LatticeDiscreteSetUp:
00236                 errorMessage = "The discrete size of the Simulation has not been set up";
00237                 break;
00238             case LatticePhysicalSetUp:
00239                 errorMessage = "The physical size of the Simulation has not been set up";
00240                 break;
00241             case LatticePatchworkSetUp:
00242                 errorMessage = "The patchwork for the Simulation has not been set up";
00243                 break;
00244             case CvodeObjectsetUp:
00245                 errorMessage = "The CVODE object has not been initialized";
00246                 break;
00247             case SimulationStarted:
00248                 errorMessage = "The Simulation has not been started";
00249                 break;
00250             default:
00251                 errorMessage = "Uppss, you've made a non-standard error, sadly I can't "
00252                             "help you there";
00253                 break;
00254         }
00255         errorKill(errorMessage);
00256     }
00257     return;
00258 }
```

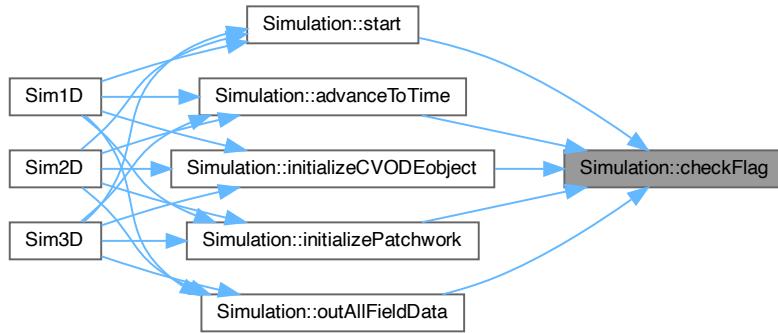
References [CvodeObjectsetUp](#), [errorKill\(\)](#), [LatticeDiscreteSetUp](#), [LatticePatchworkSetUp](#), [LatticePhysicalSetUp](#), [SimulationStarted](#), and [statusFlags](#).

Referenced by [advanceToTime\(\)](#), [initializeCVODEobject\(\)](#), [initializePatchwork\(\)](#), [outAllFieldData\(\)](#), and [start\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.6 `checkNoFlag()`

```
void Simulation::checkNoFlag (
    unsigned int flag ) const
```

function to check if flag has not been set

Check absence of configuration flags.

Definition at line 261 of file [SimulationClass.cpp](#).

```
00261
00262     if ((statusFlags & flag)) {
00263         std::string errorMessage;
00264         switch (flag) {
00265             case LatticeDiscreteSetUp:
00266                 errorMessage =
00267                     "The discrete size of the Simulation has already been set up";
00268                 break;
00269             case LatticePhysicalSetUp:
00270                 errorMessage =
00271                     "The physical size of the Simulation has already been set up";
00272                 break;
00273             case LatticePatchworkSetUp:
00274                 errorMessage = "The patchwork for the Simulation has already been set up";
00275                 break;
00276             case CvodeObjectSetUp:
00277                 errorMessage = "The CVODE object has already been initialized";
00278                 break;
00279             case SimulationStarted:
00280                 errorMessage = "The simulation has already started, some changes are no "
00281                             "longer possible";
00282                 break;
00283             default:
00284                 errorMessage = "Uppss, you've made a non-standard error, sadly I can't "
00285                             "help you there";
00286                 break;
00287         }
00288         errorKill(errorMessage);
00289     }
00290     return;
00291 }
```

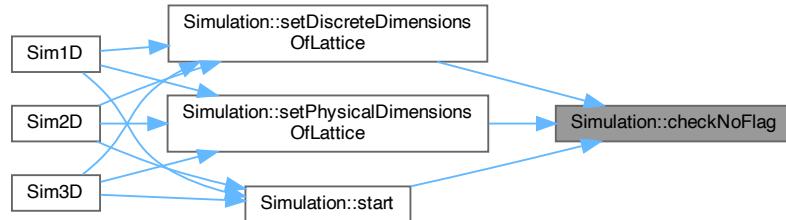
References [CvodeObjectSetUp](#), [errorKill\(\)](#), [LatticeDiscreteSetUp](#), [LatticePatchworkSetUp](#), [LatticePhysicalSetUp](#), [SimulationStarted](#), and [statusFlags](#).

Referenced by [setDiscreteDimensionsOfLattice\(\)](#), [setPhysicalDimensionsOfLattice\(\)](#), and [start\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.7 [get_cart_comm\(\)](#)

```
MPI_Comm * Simulation::get_cart_comm ( ) [inline]
```

reference to the cartesian communicator of the lattice (for debugging)

Definition at line 56 of file [SimulationClass.h](#).
00056 { [return](#) &lattice.comm; }

References [Lattice::comm](#), and [lattice](#).

5.16.3.8 initializeCVODEobject()

```
void Simulation::initializeCVODEobject (
    const sunrealtype reltol,
    const sunrealtype abstol )
```

function to initialize the CVODE object with all requirements

Configure CVODE.

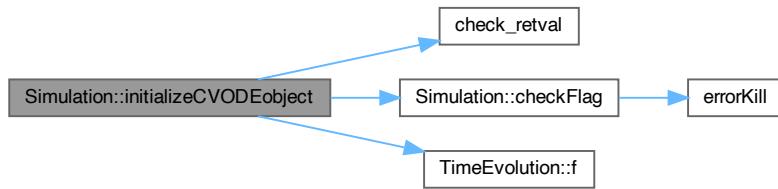
Definition at line 70 of file [SimulationClass.cpp](#).

```
00071     {
00072         checkFlag(SimulationStarted);
00073
00074         // CVode settings return value
00075         int retval = 0;
00076
00077         // Create CVODE object -- returns a pointer to the cvode memory structure
00078         // with Adams method (Adams-Moulton formula) solver chosen for non-stiff ODE
00079         cvode_mem = CVodeCreate(CV_ADAMS, lattice.sunctx);
00080
00081         // Specify user data and attach it to the main cvode memory block
00082         retval = CVodeSetUserData(
00083             cvode_mem,
00084             &latticePatch); // patch contains the user data as used in CVRhsFn
00085         if (check_retval(&retval, "CVodeSetUserData", 1, lattice.my_prc))
00086             MPI_Abort(lattice.comm, 1);
00087
00088         // Initialize CVODE solver
00089         retval = CVodeInit(cvode_mem, TimeEvolution::f, 0,
00090                             latticePatch.u); // allocate memory, CVRhsFn f, t_i=0, u
00091                             // contains the initial values
00092         if (check_retval(&retval, "CVodeInit", 1, lattice.my_prc))
00093             MPI_Abort(lattice.comm, 1);
00094
00095         // Create fixed point nonlinear solver object (suitable for non-stiff ODE) and
00096         // attach it to CVode
00097         NLS = SUNNonlinSol_FixedPoint(latticePatch.u, 0, lattice.sunctx);
00098         retval = CVodeSetNonlinearSolver(cvode_mem, NLS);
00099         if (check_retval(&retval, "CVodeSetNonlinearSolver", 1, lattice.my_prc))
00100             MPI_Abort(lattice.comm, 1);
00101
00102         // Anderson damping factor
00103         retval = SUNNonlinSolSetDamping_FixedPoint(NLS,1);
00104         if (check_retval(&retval, "SUNNonlinSolSetDamping_FixedPoint", 1,
00105             lattice.my_prc)) MPI_Abort(lattice.comm, 1);
00106
00107         // Specify integration tolerances -- a scalar relative tolerance and scalar
00108         // absolute tolerance
00109         retval = CVodeSStolerances(cvode_mem, reltol, abstol);
00110         if (check_retval(&retval, "CVodeSStolerances", 1, lattice.my_prc))
00111             MPI_Abort(lattice.comm, 1);
00112
00113         // Specify the maximum number of steps to be taken by the solver in its
00114         // attempt to reach the next tout
00115         retval = CVodeSetMaxNumSteps(cvode_mem, 10000);
00116         if (check_retval(&retval, "CVodeSetMaxNumSteps", 1, lattice.my_prc))
00117             MPI_Abort(lattice.comm, 1);
00118
00119         // maximum number of warnings for too small h
00120         retval = CVodeSetMaxHnilWarns(cvode_mem,3);
00121         if (check_retval(&retval, "CVodeSetMaxHnilWarns", 1, lattice.my_prc))
00122             MPI_Abort(lattice.comm, 1);
00123
00124         statusFlags |= CvodeObjectSetUp;
00125 }
```

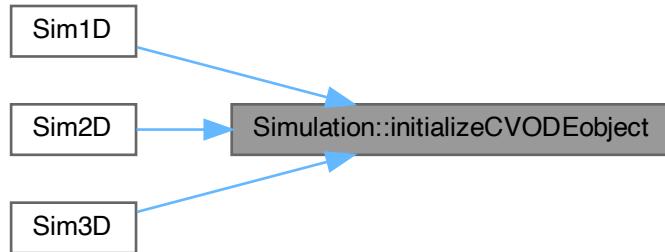
References [check_retval\(\)](#), [checkFlag\(\)](#), [Lattice::comm](#), [cvode_mem](#), [CvodeObjectSetUp](#), [TimeEvolution::f\(\)](#), [lattice](#), [latticePatch](#), [Lattice::my_prc](#), [NLS](#), [SimulationStarted](#), [statusFlags](#), [Lattice::sunctx](#), and [LatticePatch::u](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.9 initializePatchwork()

```
void Simulation::initializePatchwork (
    const int nx,
    const int ny,
    const int nz )
```

function to initialize the Patchwork

Check that the lattice dimensions are set up and generate the patchwork.

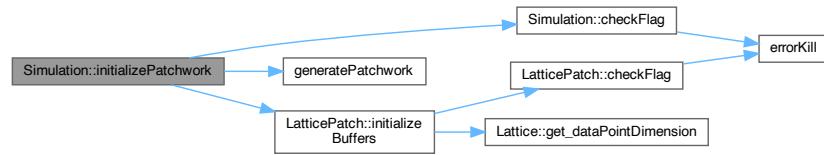
Definition at line 57 of file [SimulationClass.cpp](#).

```
00058     {
00059     checkFlag(LatticeDiscreteSetUp);
00060     checkFlag(LatticePhysicalSetUp);
00061
00062     // Generate the patchwork
00063     generatePatchwork(lattice, latticePatch, nx, ny, nz);
00064     latticePatch.initializeBuffers();
00065
00066     statusFlags |= LatticePatchworkSetUp;
00067 }
```

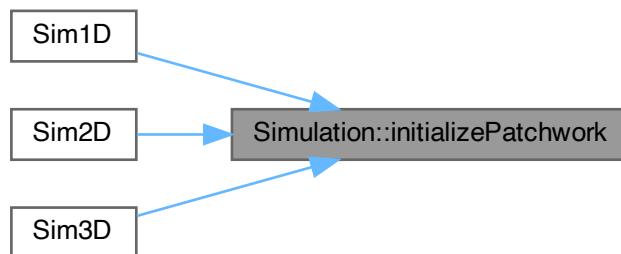
References [checkFlag\(\)](#), [generatePatchwork\(\)](#), [LatticePatch::initializeBuffers\(\)](#), [lattice](#), [LatticeDiscreteSetUp](#), [latticePatch](#), [LatticePatchworkSetUp](#), [LatticePhysicalSetUp](#), and [statusFlags](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.10 outAllFieldData()

```
void Simulation::outAllFieldData (
    const int & state )
```

function to write field data to disk

Write specified simulation steps to disk.

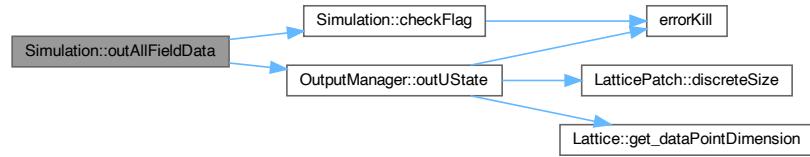
Definition at line 225 of file [SimulationClass.cpp](#).

```
00225
00226     checkFlag(SimulationStarted);
00227     outputManager.outUState(state, lattice, latticePatch);
00228 }
```

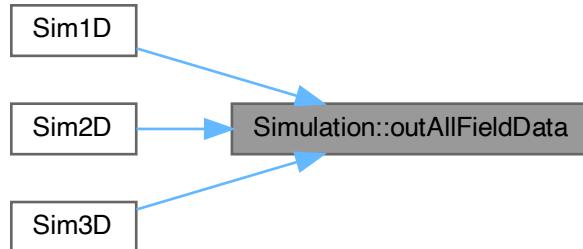
References [checkFlag\(\)](#), [lattice](#), [latticePatch](#), [outputManager](#), [OutputManager::outUState\(\)](#), and [SimulationStarted](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.11 setDiscreteDimensionsOfLattice()

```

void Simulation::setDiscreteDimensionsOfLattice (
    const sunindextype _tot_nx,
    const sunindextype _tot_ny,
    const sunindextype _tot_nz )
  
```

function to set discrete dimensions of the lattice

Set the discrete dimensions, the number of points per dimension.

Definition at line 41 of file [SimulationClass.cpp](#).

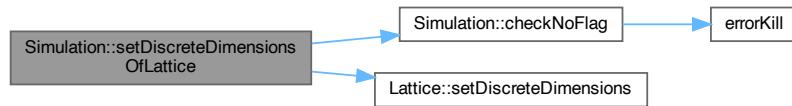
```

00042
00043     checkNoFlag(LatticePatchworkSetUp);
00044     lattice.setDiscreteDimensions(nx, ny, nz);
00045     statusFlags |= LatticeDiscreteSetUp;
00046 }
  
```

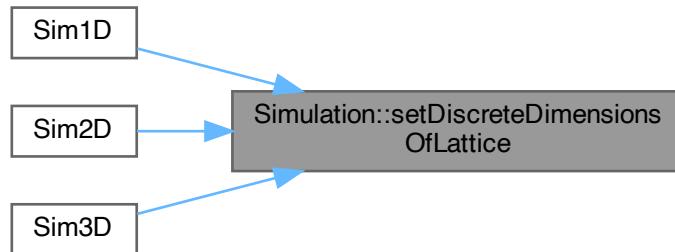
References [checkNoFlag\(\)](#), [lattice](#), [LatticeDiscreteSetUp](#), [LatticePatchworkSetUp](#), [Lattice::setDiscreteDimensions\(\)](#), and [statusFlags](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.12 `setInitialConditions()`

```
void Simulation::setInitialConditions ( )
```

functions to set the initial field configuration onto the lattice

Set initial conditions: Fill the lattice points with the initial field values

Definition at line 140 of file [SimulationClass.cpp](#).

```

00140 {
00141     const sunrealtyp dx = latticePatch.getDelta(1);
00142     const sunrealtyp dy = latticePatch.getDelta(2);
00143     const sunrealtyp dz = latticePatch.getDelta(3);
00144     const sunindextyp nx = latticePatch.discreteSize(1);
00145     const sunindextyp ny = latticePatch.discreteSize(2);
00146     const sunindextyp totalNP = latticePatch.discreteSize();
00147     const sunrealtyp x0 = latticePatch.origin(1);
00148     const sunrealtyp y0 = latticePatch.origin(2);
00149     const sunrealtyp z0 = latticePatch.origin(3);
00150     sunindextyp px = 0, py = 0, pz = 0;
00151 #pragma omp parallel for default(none) \
00152     shared(nx, ny, totalNP, dx, dy, dz, x0, y0, z0) \
00153     firstprivate(px, py, pz) schedule(static)
00154     for (sunindextyp i = 0; i < totalNP * 6; i += 6) {
00155         px = (i / 6) % nx;
00156         py = ((i / 6) / nx) % ny;
00157         pz = ((i / 6) / nx) / ny;
00158         // Call the 'eval' function to fill the lattice points with the field data
00159         icsettings.eval(static_cast<sunrealtyp>(px) * dx + x0,
00160                         static_cast<sunrealtyp>(py) * dy + y0,
  
```

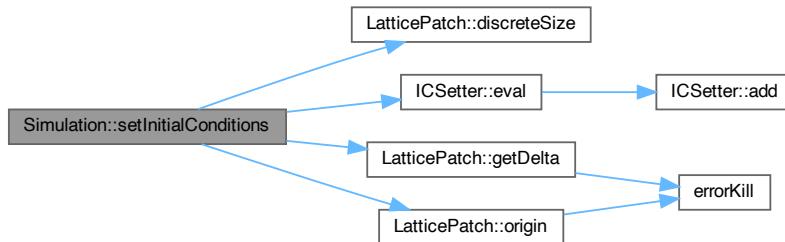
```

00161         static_cast<sunrealtype>(pz) * dz + z0, &latticePatch.uData[i]);
00162     }
00163     return;
00164 }
```

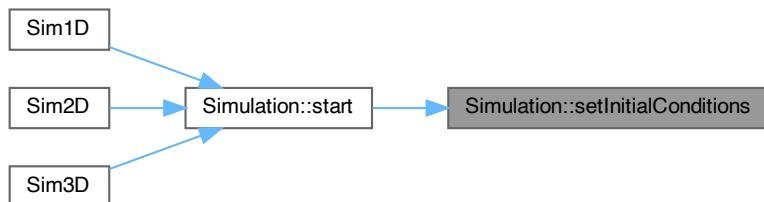
References [LatticePatch::discreteSize\(\)](#), [ICSetter::eval\(\)](#), [LatticePatch::getDelta\(\)](#), [icsettings](#), [latticePatch](#), [LatticePatch::origin\(\)](#), and [LatticePatch::uData](#).

Referenced by [start\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.13 setPhysicalDimensionsOfLattice()

```

void Simulation::setPhysicalDimensionsOfLattice (
    const unrealtype lx,
    const unrealtype ly,
    const unrealtype lz )
```

function to set physical dimensions of the lattice

Set the physical dimensions with lengths in micro meters.

Definition at line 49 of file [SimulationClass.cpp](#).

```
00050 {
```

```

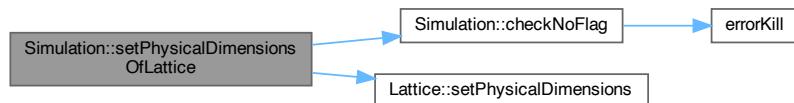
00051     checkNoFlag(LatticePatchworkSetUp);
00052     lattice.setPhysicalDimensions(lx, ly, lz);
00053     statusFlags |= LatticePhysicalSetUp;
00054 }

```

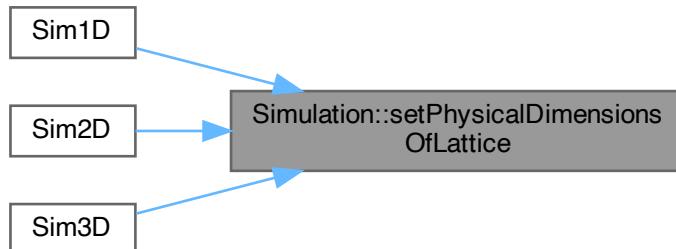
References [checkNoFlag\(\)](#), [lattice](#), [LatticePatchworkSetUp](#), [LatticePhysicalSetUp](#), [Lattice::setPhysicalDimensions\(\)](#), and [statusFlags](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.3.14 start()

```
void Simulation::start( )
```

function to start the simulation for time iteration

Check if the lattice patchwork is set up and set the initial conditions.

Definition at line 128 of file [SimulationClass.cpp](#).

```

00128     {
00129     checkFlag(LatticeDiscreteSetUp);
00130     checkFlag(LatticePhysicalSetUp);
00131     checkFlag(LatticePatchworkSetUp);
00132     checkNoFlag(SimulationStarted);
00133     checkNoFlag(CvodeObjectSetUp);
00134     setInitialConditions();
00135     statusFlags |= SimulationStarted;

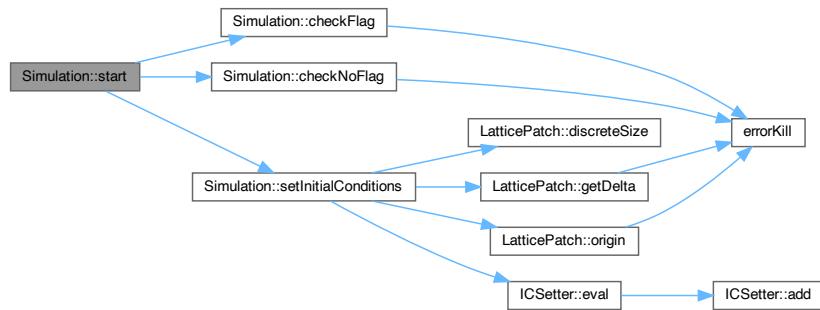
```

```
00136 }
```

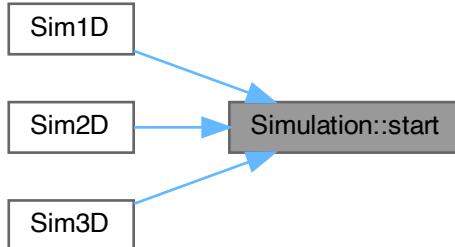
References [checkFlag\(\)](#), [checkNoFlag\(\)](#), [CvodeObjectSetUp](#), [LatticeDiscreteSetUp](#), [LatticePatchworkSetUp](#), [LatticePhysicalSetUp](#), [setInitialConditions\(\)](#), [SimulationStarted](#), and [statusFlags](#).

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



5.16.4 Field Documentation

5.16.4.1 cvode_mem

```
void* Simulation::cvode_mem
```

pointer to CVode memory object

Definition at line [47](#) of file [SimulationClass.h](#).

Referenced by [advanceToTime\(\)](#), [initializeCVODEobject\(\)](#), and [~Simulation\(\)](#).

5.16.4.2 icsettings

`ICSetter` `Simulation::icsettings`

IC Setter object.

Definition at line 43 of file [SimulationClass.h](#).

Referenced by [addInitialConditions\(\)](#), [setInitialConditions\(\)](#), [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

5.16.4.3 lattice

`Lattice` `Simulation::lattice` [private]

Lattice object.

Definition at line 33 of file [SimulationClass.h](#).

Referenced by [addInitialConditions\(\)](#), [advanceToTime\(\)](#), [get_cart_comm\(\)](#), [initializeCVODEobject\(\)](#), [initializePatchwork\(\)](#), [outAllFieldData\(\)](#), [setDiscreteDimensionsOfLattice\(\)](#), [setPhysicalDimensionsOfLattice\(\)](#), [Simulation\(\)](#), and [~Simulation\(\)](#).

5.16.4.4 latticePatch

`LatticePatch` `Simulation::latticePatch` [private]

LatticePatch object.

Definition at line 35 of file [SimulationClass.h](#).

Referenced by [addInitialConditions\(\)](#), [advanceToTime\(\)](#), [initializeCVODEobject\(\)](#), [initializePatchwork\(\)](#), [outAllFieldData\(\)](#), and [setInitialConditions\(\)](#).

5.16.4.5 NLS

`SUNNonlinearSolver` `Simulation::NLS`

nonlinear solver object

Definition at line 49 of file [SimulationClass.h](#).

Referenced by [initializeCVODEobject\(\)](#), and [~Simulation\(\)](#).

5.16.4.6 outputManager

`OutputManager Simulation::outputManager`

Output Manager object.

Definition at line 45 of file [SimulationClass.h](#).

Referenced by [outAllFieldData\(\)](#), [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

5.16.4.7 statusFlags

`unsigned int Simulation::statusFlags [private]`

simulation status flags

Definition at line 39 of file [SimulationClass.h](#).

Referenced by [checkFlag\(\)](#), [checkNoFlag\(\)](#), [initializeCVODEobject\(\)](#), [initializePatchwork\(\)](#), [setDiscreteDimensionsOfLattice\(\)](#), [setPhysicalDimensionsOfLattice\(\)](#), [Simulation\(\)](#), [start\(\)](#), and [~Simulation\(\)](#).

5.16.4.8 t

`sunrealtype Simulation::t [private]`

current time of the simulation

Definition at line 37 of file [SimulationClass.h](#).

Referenced by [advanceToTime\(\)](#), and [Simulation\(\)](#).

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

- src/[SimulationClass.h](#)
- src/[SimulationClass.cpp](#)

5.17 TimeEvolution Class Reference

monostate `TimeEvolution` class to propagate the field data in time in a given order of the HE weak-field expansion

```
#include <src/TimeEvolutionFunctions.h>
```

Static Public Member Functions

- static int `f` (`sunrealtype t, N_Vector u, N_Vector udot, void *data_loc`)
CVODE right hand side function (CVRhsFn) to provide IVP of the ODE.

Static Public Attributes

- static int * **c** = nullptr
choice which processes of the weak field expansion are included
- static void(* **TimeEvolver**)(LatticePatch *, N_Vector, N_Vector, int *) = **nonlinear1DProp**
Pointer to functions for differentiation and time evolution.

5.17.1 Detailed Description

monostate [TimeEvolution](#) class to propagate the field data in time in a given order of the HE weak-field expansion

Definition at line 15 of file [TimeEvolutionFunctions.h](#).

5.17.2 Member Function Documentation

5.17.2.1 f()

```
int TimeEvolution::f (
    sunrealtype t,
    N_Vector u,
    N_Vector udot,
    void * data_loc ) [static]
```

CVODE right hand side function (CVRhsFn) to provide IVP of the ODE.

CVode right-hand-side function (CVRhsFn)

Definition at line 11 of file [TimeEvolutionFunctions.cpp](#).

```
00011
00012
00013 // Set recover pointer to provided lattice patch where the field data resides
00014 LatticePatch *data = static_cast<LatticePatch *>(data_loc);
00015
00016 // update circle
00017 // Access provided field values and temp. derivatievees with NVector pointers
00018 #if defined(_OPENMP)
00019     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00020             *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00021 #else
00022     sunrealtype *udata = NV_DATA_P(u),
00023             *dudata = NV_DATA_P(udot);
00024 #endif
00025
00026 // Store original data location of the patch
00027 sunrealtype *originaluData = data->uData,
00028         *originalduData = data->duData;
00029
00030 // Point patch data to arguments of f
00031 data->uData = udata;
00032 data->duData = dudata;
00033
00034 // Time-evolve these arguments (the field data) with specific propagator below
00035 TimeEvolver(data, u, udot, c);
00036
00037 // Refer patch data back to original location
00038 data->uData = originaluData;
00039 data->duData = originalduData;
00040
00041 return (0);
00042 }
```

References [c](#), [LatticePatch::duData](#), [TimeEvolver](#), and [LatticePatch::uData](#).

Referenced by [Simulation::initializeCVODEobject\(\)](#).

Here is the caller graph for this function:



5.17.3 Field Documentation

5.17.3.1 c

```
int * TimeEvolution::c = nullptr [static]
```

choice which processes of the weak field expansion are included

Definition at line 18 of file [TimeEvolutionFunctions.h](#).

Referenced by [f\(\)](#), [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

5.17.3.2 TimeEvolver

```
void(* TimeEvolution::TimeEvolver) (LatticePatch *, N_Vector, N_Vector, int *) = nonlinear1DProp [static]
```

Pointer to functions for differentiation and time evolution.

Definition at line 21 of file [TimeEvolutionFunctions.h](#).

Referenced by [f\(\)](#), [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

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

- [src/TimeEvolutionFunctions.h](#)
- [src/SimulationFunctions.cpp](#)
- [src/TimeEvolutionFunctions.cpp](#)

Chapter 6

File Documentation

6.1 README.md File Reference

6.2 src/build/CMakeFiles/3.24.2/CompilerIdC/CMakeCCCompilerId.c File Reference

Macros

- #define __has_include(x) 0
- #define COMPILER_ID ""
- #define STRINGIFY_HELPER(X) #X
- #define STRINGIFY(X) STRINGIFY_HELPER(X)
- #define PLATFORM_ID
- #define ARCHITECTURE_ID
- #define DEC(n)
- #define HEX(n)
- #define C_VERSION

Functions

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

Variables

- char const * info_compiler = "INFO ":" compiler[" COMPILER_ID "]"
- char const * info_platform = "INFO ":" platform[" PLATFORM_ID "]"
- char const * info_arch = "INFO ":" arch[" ARCHITECTURE_ID "]"
- const char * info_language_standard_default
- const char * info_language_extensions_default

6.2.1 Macro Definition Documentation

6.2.1.1 __has_include

```
#define __has_include(  
    x ) 0
```

Definition at line 17 of file [CMakeCCompilerId.c](#).

6.2.1.2 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

Definition at line 688 of file [CMakeCCompilerId.c](#).

6.2.1.3 C_VERSION

```
#define C_VERSION
```

Definition at line 777 of file [CMakeCCompilerId.c](#).

6.2.1.4 COMPILER_ID

```
#define COMPILER_ID ""
```

Definition at line 423 of file [CMakeCCompilerId.c](#).

6.2.1.5 DEC

```
#define DEC(  
    n )
```

Value:

```
('0' + (((n) / 10000000)%10)), \  
('0' + (((n) / 1000000)%10)), \  
('0' + (((n) / 100000)%10)), \  
('0' + (((n) / 10000)%10)), \  
('0' + (((n) / 1000)%10)), \  
('0' + (((n) / 100)%10)), \  
('0' + (((n) / 10)%10)), \  
('0' + ((n) % 10))
```

Definition at line 692 of file [CMakeCCompilerId.c](#).

6.2.1.6 HEX

```
#define HEX(  
    n )
```

Value:

```
('0' + ((n) >> 28 & 0xF)), \  
'0' + ((n) >> 24 & 0xF)), \  
'0' + ((n) >> 20 & 0xF)), \  
'0' + ((n) >> 16 & 0xF)), \  
'0' + ((n) >> 12 & 0xF)), \  
'0' + ((n) >> 8 & 0xF)), \  
'0' + ((n) >> 4 & 0xF)), \  
'0' + ((n) & 0xF))
```

Definition at line 703 of file [CMakeCCompilerId.c](#).

6.2.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

Definition at line 554 of file [CMakeCCompilerId.c](#).

6.2.1.8 STRINGIFY

```
#define STRINGIFY(  
    X ) STRINGIFY_HELPER(X)
```

Definition at line 444 of file [CMakeCCompilerId.c](#).

6.2.1.9 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(  
    X ) #X
```

Definition at line 443 of file [CMakeCCompilerId.c](#).

6.2.2 Function Documentation

6.2.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Definition at line 811 of file [CMakeCCompilerId.c](#).

```
00813 {
00814     int require = 0;
00815     require += info\_compiler[argc];
00816     require += info\_platform[argc];
00817     require += info\_arch[argc];
00818 #ifdef COMPILER_VERSION_MAJOR
00819     require += info\_version[argc];
00820 #endif
00821 #ifdef COMPILER_VERSION_INTERNAL
00822     require += info\_version\_internal[argc];
00823 #endif
00824 #ifdef SIMULATE_ID
00825     require += info\_simulate[argc];
00826 #endif
00827 #ifdef SIMULATE_VERSION_MAJOR
00828     require += info\_simulate\_version[argc];
00829 #endif
00830 #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
00831     require += info\_cray[argc];
00832 #endif
00833     require += info\_language\_standard\_default[argc];
00834     require += info\_language\_extensions\_default[argc];
00835     (void)argv;
00836     return require;
00837 }
```

References [info_arch](#), [info_compiler](#), [info_language_extensions_default](#), [info_language_standard_default](#), and [info_platform](#).

6.2.3 Variable Documentation

6.2.3.1 info_arch

```
char const* info\_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

Definition at line 769 of file [CMakeCCompilerId.c](#).

Referenced by [main\(\)](#).

6.2.3.2 info_compiler

```
char const* info\_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

Definition at line 430 of file [CMakeCCompilerId.c](#).

Referenced by [main\(\)](#).

6.2.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

Initial value:

```
= "INFO" ":" "extensions_default["
    "OFF"
"]"
```

Definition at line 793 of file [CMakeCCompilerId.c](#).

Referenced by [main\(\)](#).

6.2.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

Initial value:

```
= "INFO" ":" "standard_default[" C_VERSION "]"
```

Definition at line 790 of file [CMakeCCompilerId.c](#).

Referenced by [main\(\)](#).

6.2.3.5 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

Definition at line 768 of file [CMakeCCompilerId.c](#).

Referenced by [main\(\)](#).

6.3 CMakeCCompilerId.c

[Go to the documentation of this file.](#)

```
00001 #ifdef __cplusplus
00002 # error "A C++ compiler has been selected for C."
00003 #endif
00004
00005 #if defined(__18CXX)
00006 # define ID_VOID_MAIN
00007 #endif
00008 #if defined(__CLASSIC_C__)
00009 /* cv-qualifiers did not exist in K&R C */
00010 # define const
00011 # define volatile
00012 #endif
00013
00014 #if !defined(__has_include)
00015 /* If the compiler does not have __has_include, pretend the answer is
00016     always no. */
00017 # define __has_include(x) 0
00018 #endif
00019
00020
00021 /* Version number components: V=Version, R=Revision, P=Patch
```

```

00022     Version date components:  YYYY=Year, MM=Month,   DD=Day  */
00023
00024 #if defined(__INTEL_COMPILER) || defined(__ICC)
00025 # define COMPILER_ID "Intel"
00026 # if defined(__MSC_VER)
00027 #  define SIMULATE_ID "MSVC"
00028 # endif
00029 # if defined(__GNUC__)
00030 #  define SIMULATE_ID "GNU"
00031 # endif
00032 /* __INTEL_COMPILER = VRR prior to 2021, and then VVVV for 2021 and later,
00033    except that a few beta releases use the old format with V=2021.  */
00034 # if __INTEL_COMPILER < 2021 || __INTEL_COMPILER == 202110 || __INTEL_COMPILER == 202111
00035 #  define COMPILER_VERSION_MAJOR DEC(__INTEL_COMPILER/100)
00036 #  define COMPILER_VERSION_MINOR DEC(__INTEL_COMPILER/10 % 10)
00037 # if defined(__INTEL_COMPILER_UPDATE)
00038 #  define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER_UPDATE)
00039 # else
00040 #  define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER % 10)
00041 # endif
00042 # else
00043 #  define COMPILER_VERSION_MAJOR DEC(__INTEL_COMPILER)
00044 #  define COMPILER_VERSION_MINOR DEC(__INTEL_COMPILER_UPDATE)
00045 /* The third version component from --version is an update index,
00046    but no macro is provided for it.  */
00047 #  define COMPILER_VERSION_PATCH DEC(0)
00048 # endif
00049 # if defined(__INTEL_COMPILER_BUILD_DATE)
00050 /* __INTEL_COMPILER_BUILD_DATE = YYYYMMDD */
00051 #  define COMPILER_VERSION_TWEAK DEC(__INTEL_COMPILER_BUILD_DATE)
00052 # endif
00053 # if defined(__MSC_VER)
00054 /* __MSC_VER = VVRR */
00055 #  define SIMULATE_VERSION_MAJOR DEC(__MSC_VER / 100)
00056 #  define SIMULATE_VERSION_MINOR DEC(__MSC_VER % 100)
00057 # endif
00058 # if defined(__GNUC__)
00059 #  define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
00060 # elif defined(__GNUG__)
00061 #  define SIMULATE_VERSION_MAJOR DEC(__GNUG__)
00062 # endif
00063 # if defined(__GNUC_MINOR__)
00064 #  define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
00065 # endif
00066 # if defined(__GNUC_PATCHLEVEL__)
00067 #  define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00068 # endif
00069
00070 #elif (defined(__clang__) && defined(__INTEL_CLANG_COMPILER)) || defined(__INTEL_LLVM_COMPILER)
00071 # define COMPILER_ID "IntellLVM"
00072 #if defined(__MSC_VER)
00073 # define SIMULATE_ID "MSVC"
00074 #endif
00075 #if defined(__GNUC__)
00076 # define SIMULATE_ID "GNU"
00077 #endif
00078 /* __INTEL_LLVM_COMPILER = VVVVRP prior to 2021.2.0, VVVVRRPP for 2021.2.0 and
00079 * later.  Look for 6 digit vs. 8 digit version number to decide encoding.
00080 * VVVV is no smaller than the current year when a version is released.
00081 */
00082 #if __INTEL_LLVM_COMPILER < 1000000
00083 #  define COMPILER_VERSION_MAJOR DEC(__INTEL_LLVM_COMPILER/100)
00084 #  define COMPILER_VERSION_MINOR DEC(__INTEL_LLVM_COMPILER/10 % 10)
00085 #  define COMPILER_VERSION_PATCH DEC(__INTEL_LLVM_COMPILER % 10)
00086 #else
00087 #  define COMPILER_VERSION_MAJOR DEC(__INTEL_LLVM_COMPILER/10000)
00088 #  define COMPILER_VERSION_MINOR DEC(__INTEL_LLVM_COMPILER/100 % 100)
00089 #  define COMPILER_VERSION_PATCH DEC(__INTEL_LLVM_COMPILER % 100)
00090 #endif
00091 #if defined(__MSC_VER)
00092 /* __MSC_VER = VVRR */
00093 #  define SIMULATE_VERSION_MAJOR DEC(__MSC_VER / 100)
00094 #  define SIMULATE_VERSION_MINOR DEC(__MSC_VER % 100)
00095 #endif
00096 #if defined(__GNUC__)
00097 #  define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
00098 #elif defined(__GNUG__)
00099 #  define SIMULATE_VERSION_MAJOR DEC(__GNUG__)
00100 #endif
00101 #if defined(__GNUC_MINOR__)
00102 #  define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
00103 #endif
00104 #if defined(__GNUC_PATCHLEVEL__)
00105 #  define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00106 #endif
00107
00108 #elif defined(__PATHCC__)

```

```

00109 # define COMPILER_ID "PathScale"
00110 # define COMPILER_VERSION_MAJOR DEC(__PATHCC__)
00111 # define COMPILER_VERSION_MINOR DEC(__PATHCC_MINOR__)
00112 # if defined(__PATHCC_PATCHLEVEL__)
00113 # define COMPILER_VERSION_PATCH DEC(__PATHCC_PATCHLEVEL__)
00114 # endif
00115
00116 #elif defined(__BORLANDC__) && defined(__CODEGEARC_VERSION__)
00117 # define COMPILER_ID "Embarcadero"
00118 # define COMPILER_VERSION_MAJOR HEX(__CODEGEARC_VERSION__>>24 & 0x0FFF)
00119 # define COMPILER_VERSION_MINOR HEX(__CODEGEARC_VERSION__>>16 & 0x0FFF)
00120 # define COMPILER_VERSION_PATCH DEC(__CODEGEARC_VERSION__           & 0xFFFF)
00121
00122 #elif defined(__BORLANDC__)
00123 # define COMPILER_ID "Borland"
00124 /* __BORLANDC__ = 0xVRR */
00125 # define COMPILER_VERSION_MAJOR HEX(__BORLANDC__>>8)
00126 # define COMPILER_VERSION_MINOR HEX(__BORLANDC__ & 0xFF)
00127
00128 #elif defined(__WATCOMC__) && __WATCOMC__ < 1200
00129 # define COMPILER_ID "Watcom"
00130 /* __WATCOMC__ = VVRR */
00131 # define COMPILER_VERSION_MAJOR DEC(__WATCOMC__ / 100)
00132 # define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
00133 # if (__WATCOMC__ % 10) > 0
00134 # define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
00135 # endif
00136
00137 #elif defined(__WATCOMC__)
00138 # define COMPILER_ID "OpenWatcom"
00139 /* __WATCOMC__ = VVRP + 1100 */
00140 # define COMPILER_VERSION_MAJOR DEC((__WATCOMC__ - 1100) / 100)
00141 # define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
00142 # if (__WATCOMC__ % 10) > 0
00143 # define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
00144 # endif
00145
00146 #elif defined(__SUNPRO_C)
00147 # define COMPILER_ID "SunPro"
00148 # if __SUNPRO_C >= 0x5100
00149 /* __SUNPRO_C = 0xVRRP */
00150 # define COMPILER_VERSION_MAJOR HEX(__SUNPRO_C>>12)
00151 # define COMPILER_VERSION_MINOR HEX(__SUNPRO_C>>4 & 0xFF)
00152 # define COMPILER_VERSION_PATCH HEX(__SUNPRO_C           & 0xF)
00153 # else
00154 /* __SUNPRO_CC = 0xVRP */
00155 # define COMPILER_VERSION_MAJOR HEX(__SUNPRO_C>>8)
00156 # define COMPILER_VERSION_MINOR HEX(__SUNPRO_C>>4 & 0xF)
00157 # define COMPILER_VERSION_PATCH HEX(__SUNPRO_C           & 0xF)
00158 # endif
00159
00160 #elif defined(__HP_cc)
00161 # define COMPILER_ID "HP"
00162 /* __HP_cc = VVRTPPP */
00163 # define COMPILER_VERSION_MAJOR DEC(__HP_cc/10000)
00164 # define COMPILER_VERSION_MINOR DEC(__HP_cc/100 % 100)
00165 # define COMPILER_VERSION_PATCH DEC(__HP_cc           % 100)
00166
00167 #elif defined(__DECC)
00168 # define COMPILER_ID "Compaq"
00169 /* __DECC_VER = VVRRTPPP */
00170 # define COMPILER_VERSION_MAJOR DEC(__DECC_VER/10000000)
00171 # define COMPILER_VERSION_MINOR DEC(__DECC_VER/100000 % 100)
00172 # define COMPILER_VERSION_PATCH DEC(__DECC_VER           % 10000)
00173
00174 #elif defined(__IBMC__) && defined(__COMPILER_VER__)
00175 # define COMPILER_ID "zOS"
00176 /* __IBMC__ = VRP */
00177 # define COMPILER_VERSION_MAJOR DEC(__IBMC__/100)
00178 # define COMPILER_VERSION_MINOR DEC(__IBMC__/10 % 10)
00179 # define COMPILER_VERSION_PATCH DEC(__IBMC__           % 10)
00180
00181 #elif defined(__open_xl__) && defined(__clang__)
00182 # define COMPILER_ID "IBMCclang"
00183 # define COMPILER_VERSION_MAJOR DEC(__open_xl_version__)
00184 # define COMPILER_VERSION_MINOR DEC(__open_xl_release__)
00185 # define COMPILER_VERSION_PATCH DEC(__open_xl_modification__)
00186 # define COMPILER_VERSION_TWEAK DEC(__open_xl_ptf_fix_level__)
00187
00188
00189 #elif defined(__ibmxl__) && defined(__clang__)
00190 # define COMPILER_ID "XLClang"
00191 # define COMPILER_VERSION_MAJOR DEC(__ibmxl_version__)
00192 # define COMPILER_VERSION_MINOR DEC(__ibmxl_release__)
00193 # define COMPILER_VERSION_PATCH DEC(__ibmxl_modification__)
00194 # define COMPILER_VERSION_TWEAK DEC(__ibmxl_ptf_fix_level__)
00195

```

```

00196
00197 #elif defined(__IBMC__) && !defined(__COMPILER_VER__)
00198 # define COMPILER_ID "XL"
00199 /* __IBMC__ = VRP */
00200 # define COMPILER_VERSION_MAJOR DEC(__IBMC__/100)
00201 # define COMPILER_VERSION_MINOR DEC(__IBMC__/10 % 10)
00202 # define COMPILER_VERSION_PATCH DEC(__IBMC__ % 10)
00203
00204 #elif defined(__IBMC__) && !defined(__COMPILER_VER__)
00205 # define COMPILER_ID "VisualAge"
00206 /* __IBMC__ = VRP */
00207 # define COMPILER_VERSION_MAJOR DEC(__IBMC__/100)
00208 # define COMPILER_VERSION_MINOR DEC(__IBMC__/10 % 10)
00209 # define COMPILER_VERSION_PATCH DEC(__IBMC__ % 10)
00210
00211 #elif defined(__NVCOMPILER)
00212 # define COMPILER_ID "NVHPC"
00213 # define COMPILER_VERSION_MAJOR DEC(__NVCOMPILER_MAJOR__)
00214 # define COMPILER_VERSION_MINOR DEC(__NVCOMPILER_MINOR__)
00215 # if defined(__NVCOMPILER_PATCHLEVEL__)
00216 # define COMPILER_VERSION_PATCH DEC(__NVCOMPILER_PATCHLEVEL__)
00217 # endif
00218
00219 #elif defined(__PGI)
00220 # define COMPILER_ID "PGI"
00221 # define COMPILER_VERSION_MAJOR DEC(__PGIC__)
00222 # define COMPILER_VERSION_MINOR DEC(__PGIC_MINOR__)
00223 # if defined(__PGIC_PATCHLEVEL__)
00224 # define COMPILER_VERSION_PATCH DEC(__PGIC_PATCHLEVEL__)
00225 # endif
00226
00227 #elif defined(__CRAYC)
00228 # define COMPILER_ID "Cray"
00229 # define COMPILER_VERSION_MAJOR DEC(__RELEASE_MAJOR)
00230 # define COMPILER_VERSION_MINOR DEC(__RELEASE_MINOR)
00231
00232 #elif defined(__TI_COMPILER_VERSION__)
00233 # define COMPILER_ID "TI"
00234 /* __TI_COMPILER_VERSION__ = VVVRRPPP */
00235 # define COMPILER_VERSION_MAJOR DEC(__TI_COMPILER_VERSION__/1000000)
00236 # define COMPILER_VERSION_MINOR DEC(__TI_COMPILER_VERSION__/1000 % 1000)
00237 # define COMPILER_VERSION_PATCH DEC(__TI_COMPILER_VERSION__ % 1000)
00238
00239 #elif defined(__CLANG_FUJITSU)
00240 # define COMPILER_ID "FujitsuClang"
00241 # define COMPILER_VERSION_MAJOR DEC(__FCC_major__)
00242 # define COMPILER_VERSION_MINOR DEC(__FCC_minor__)
00243 # define COMPILER_VERSION_PATCH DEC(__FCC_patchlevel__)
00244 # define COMPILER_VERSION_INTERNAL_STR __clang_version__
00245
00246
00247 #elif defined(__FUJITSU)
00248 # define COMPILER_ID "Fujitsu"
00249 # if defined(__FCC_version__)
00250 # define COMPILER_VERSION __FCC_version__
00251 # elif defined(__FCC_major__)
00252 # define COMPILER_VERSION_MAJOR DEC(__FCC_major__)
00253 # define COMPILER_VERSION_MINOR DEC(__FCC_minor__)
00254 # define COMPILER_VERSION_PATCH DEC(__FCC_patchlevel__)
00255 # endif
00256 # if defined(__fcc_version__)
00257 # define COMPILER_VERSION_INTERNAL DEC(__fcc_version__)
00258 # elif defined(__FCC_VERSION)
00259 # define COMPILER_VERSION_INTERNAL DEC(__FCC_VERSION)
00260 # endif
00261
00262
00263 #elif defined(__ghs__)
00264 # define COMPILER_ID "GHS"
00265 /* __GHS_VERSION_NUMBER = VVVVRP */
00266 # ifdef __GHS_VERSION_NUMBER
00267 # define COMPILER_VERSION_MAJOR DEC(__GHS_VERSION_NUMBER / 100)
00268 # define COMPILER_VERSION_MINOR DEC(__GHS_VERSION_NUMBER / 10 % 10)
00269 # define COMPILER_VERSION_PATCH DEC(__GHS_VERSION_NUMBER % 10)
00270 # endif
00271
00272 #elif defined(__TINYC__)
00273 # define COMPILER_ID "TinyCC"
00274
00275 #elif defined(__BCC__)
00276 # define COMPILER_ID "Bruce"
00277
00278 #elif defined(__SCO_VERSION__)
00279 # define COMPILER_ID "SCO"
00280
00281 #elif defined(__ARMCC_VERSION) && !defined(__clang__)
00282 # define COMPILER_ID "ARMCC"

```

```

00283 #if __ARMCC_VERSION >= 1000000
00284 /* __ARMCC_VERSION = VRRPPP */
00285 # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/1000000)
00286 # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 100)
00287 # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION % 10000)
00288 #else
00289 /* __ARMCC_VERSION = VRPPPP */
00290 # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/100000)
00291 # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 10)
00292 # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION % 10000)
00293 #endif
00294
00295
00296 #elif defined(__clang__) && defined(__apple_build_version__)
00297 # define COMPILER_ID "AppleClang"
00298 # if defined(_MSC_VER)
00299 # define SIMULATE_ID "MSVC"
00300 # endif
00301 # define COMPILER_VERSION_MAJOR DEC(__clang_major__)
00302 # define COMPILER_VERSION_MINOR DEC(__clang_minor__)
00303 # define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
00304 # if defined(_MSC_VER)
00305 /* _MSC_VER = VVRR */
00306 # define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
00307 # define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
00308 # endif
00309 # define COMPILER_VERSION_TWEAK DEC(__apple_build_version__)
00310
00311 #elif defined(__clang__) && defined(__ARMCOMPILER_VERSION)
00312 # define COMPILER_ID "ARMClang"
00313 # define COMPILER_VERSION_MAJOR DEC(__ARMCOMPILER_VERSION/1000000)
00314 # define COMPILER_VERSION_MINOR DEC(__ARMCOMPILER_VERSION/10000 % 100)
00315 # define COMPILER_VERSION_PATCH DEC(__ARMCOMPILER_VERSION % 10000)
00316 # define COMPILER_VERSION_INTERNAL DEC(__ARMCOMPILER_VERSION)
00317
00318 #elif defined(__clang__)
00319 # define COMPILER_ID "Clang"
00320 # if defined(_MSC_VER)
00321 # define SIMULATE_ID "MSVC"
00322 # endif
00323 # define COMPILER_VERSION_MAJOR DEC(__clang_major__)
00324 # define COMPILER_VERSION_MINOR DEC(__clang_minor__)
00325 # define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
00326 # if defined(_MSC_VER)
00327 /* _MSC_VER = VVRR */
00328 # define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
00329 # define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
00330 # endif
00331
00332 #elif defined(__LCC__) && (defined(__GNUC__) || defined(__GNUG__) || defined(__MCST__))
00333 # define COMPILER_ID "LCC"
00334 # define COMPILER_VERSION_MAJOR DEC(1)
00335 # if defined(__LCC__)
00336 # define COMPILER_VERSION_MINOR DEC(__LCC__ - 100)
00337 # endif
00338 # if defined(__LCC_MINOR__)
00339 # define COMPILER_VERSION_PATCH DEC(__LCC_MINOR__)
00340 # endif
00341 # if defined(__GNUC__) && defined(__GNUC_MINOR__)
00342 # define SIMULATE_ID "GNU"
00343 # define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
00344 # define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
00345 # if defined(__GNUC_PATCHLEVEL__)
00346 # define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00347 # endif
00348 # endif
00349
00350 #elif defined(__GNUC__)
00351 # define COMPILER_ID "GNU"
00352 # define COMPILER_VERSION_MAJOR DEC(__GNUC__)
00353 # if defined(__GNUC_MINOR__)
00354 # define COMPILER_VERSION_MINOR DEC(__GNUC_MINOR__)
00355 # endif
00356 # if defined(__GNUC_PATCHLEVEL__)
00357 # define COMPILER_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00358 # endif
00359
00360 #elif defined(_MSC_VER)
00361 # define COMPILER_ID "MSVC"
00362 /* _MSC_VER = VVRR */
00363 # define COMPILER_VERSION_MAJOR DEC(_MSC_VER / 100)
00364 # define COMPILER_VERSION_MINOR DEC(_MSC_VER % 100)
00365 # if defined(_MSC_FULL_VER)
00366 # if _MSC_VER >= 1400
00367 /* _MSC_FULL_VER = VVRRPPPP */
00368 # define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 100000)
00369 # else

```

```

00370     /* _MSC_FULL_VER = VVRRPPPP */
00371 # define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 10000)
00372 # endif
00373 # endif
00374 # if defined(_MSC_BUILD)
00375 # define COMPILER_VERSION_TWEAK DEC(_MSC_BUILD)
00376 # endif
00377
00378 #elif defined(_ADI_COMPILER)
00379 # define COMPILER_ID "ADSP"
00380 #if defined(__VERSIONNUM__)
00381     /* __VERSIONNUM__ = 0xVVRRPPTT */
00382 # define COMPILER_VERSION_MAJOR DEC(__VERSIONNUM__ >> 24 & 0xFF)
00383 # define COMPILER_VERSION_MINOR DEC(__VERSIONNUM__ >> 16 & 0xFF)
00384 # define COMPILER_VERSION_PATCH DEC(__VERSIONNUM__ >> 8 & 0xFF)
00385 # define COMPILER_VERSION_TWEAK DEC(__VERSIONNUM__ & 0xFF)
00386#endif
00387
00388 #elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
00389 # define COMPILER_ID "IAR"
00390 # if defined(__VER__) && defined(__ICCARM__)
00391 # define COMPILER_VERSION_MAJOR DEC((__VER__ / 1000000)
00392 # define COMPILER_VERSION_MINOR DEC((__VER__ / 1000) % 1000)
00393 # define COMPILER_VERSION_PATCH DEC((__VER__ % 1000)
00394 # define COMPILER_VERSION_INTERNAL DEC(__IAR_SYSTEMS_ICC__)
00395 # elif defined(__VER__) && (defined(__ICCAVR__) || defined(__ICCRX_) || defined(__ICCRH850__)
00396     || defined(__ICCRL78_) || defined(__ICC430_) || defined(__ICCRISCV_) || defined(__ICCV850__)
00397     || defined(__ICC8051_) || defined(__ICCSSTM8__))
00398 # define COMPILER_VERSION_MAJOR DEC((__VER__ / 100)
00399 # define COMPILER_VERSION_MINOR DEC((__VER__ - ((__VER__ / 100)*100)))
00400 # define COMPILER_VERSION_PATCH DEC(__SUBVERSION__)
00401 # define COMPILER_VERSION_INTERNAL DEC(__IAR_SYSTEMS_ICC__)
00402#endif
00403
00404 #elif defined(__SDCC_VERSION_MAJOR) || defined(SDCC)
00405 # define COMPILER_ID "SDCC"
00406 # if defined(__SDCC_VERSION_MAJOR)
00407 # define COMPILER_VERSION_MAJOR DEC(__SDCC_VERSION_MAJOR)
00408 # define COMPILER_VERSION_MINOR DEC(__SDCC_VERSION_MINOR)
00409 # define COMPILER_VERSION_PATCH DEC(__SDCC_VERSION_PATCH)
00410 # else
00411     /* SDCC = VRP */
00412 # define COMPILER_VERSION_MAJOR DEC(SDCC/100)
00413 # define COMPILER_VERSION_MINOR DEC(SDCC/10 % 10)
00414 # define COMPILER_VERSION_PATCH DEC(SDCC % 10)
00415#endif
00416 /* These compilers are either not known or too old to define an
00417   identification macro. Try to identify the platform and guess that
00418   it is the native compiler. */
00419 #elif defined(__hpux) || defined(__hpua)
00420 # define COMPILER_ID "HP"
00421
00422 #else /* unknown compiler */
00423 # define COMPILER_ID ""
00424#endif
00425
00426 /* Construct the string literal in pieces to prevent the source from
00427   getting matched. Store it in a pointer rather than an array
00428   because some compilers will just produce instructions to fill the
00429   array rather than assigning a pointer to a static array. */
00430 char const* info_compiler = "INFO ":" compiler[" COMPILER_ID "];
00431 #ifdef SIMULATE_ID
00432 char const* info_simulate = "INFO ":" simulate[" SIMULATE_ID "];
00433#endif
00434
00435 #ifdef __QNXNTO__
00436 char const* qnxnto = "INFO ":" qnxnto[];
00437#endif
00438
00439 #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
00440 char const *info_cray = "INFO ":" compiler_wrapper[CrayPrgEnv]";
00441#endif
00442
00443 #define STRINGIFY_HELPER(X) #
00444 #define STRINGIFY(X) STRINGIFY_HELPER(X)
00445
00446 /* Identify known platforms by name. */
00447 #if defined(__linux) || defined(__linux__)
00448 # define PLATFORM_ID "Linux"
00449
00450 #elif defined(__MSYS__)
00451 # define PLATFORM_ID "MSYS"
00452
00453 #elif defined(__CYGWIN__)
00454 # define PLATFORM_ID "Cygwin"

```

```
00455 #elif defined(__MINGW32__)
00456 # define PLATFORM_ID "MinGW"
00458
00459 #elif defined(__APPLE__)
00460 # define PLATFORM_ID "Darwin"
00461
00462 #elif defined(_WIN32) || defined(__WIN32__) || defined(WIN32)
00463 # define PLATFORM_ID "Windows"
00464
00465 #elif defined(__FreeBSD__) || defined(__FreeBSD__)
00466 # define PLATFORM_ID "FreeBSD"
00467
00468 #elif defined(__NetBSD__) || defined(__NetBSD__)
00469 # define PLATFORM_ID "NetBSD"
00470
00471 #elif defined(__OpenBSD__) || defined(__OPENBSD__)
00472 # define PLATFORM_ID "OpenBSD"
00473
00474 #elif defined(_sun) || defined(sun)
00475 # define PLATFORM_ID "SunOS"
00476
00477 #elif defined(_AIX) || defined(__AIX) || defined(__AIX__)
00478 # define PLATFORM_ID "AIX"
00479
00480 #elif defined(_hpx) || defined(__hpx__)
00481 # define PLATFORM_ID "HP-UX"
00482
00483 #elif defined(__HAIKU__)
00484 # define PLATFORM_ID "Haiku"
00485
00486 #elif defined(_BeOS) || defined(__BEOS__) || defined(_BEOS)
00487 # define PLATFORM_ID "BeOS"
00488
00489 #elif defined(__QNX__) || defined(__QNXNTO__)
00490 # define PLATFORM_ID "QNX"
00491
00492 #elif defined(_tru64) || defined(_tr64) || defined(__TRU64__)
00493 # define PLATFORM_ID "Tru64"
00494
00495 #elif defined(_riscos) || defined(__riscos__)
00496 # define PLATFORM_ID "RISCos"
00497
00498 #elif defined(_sinix) || defined(__sinix__) || defined(__SINIX__)
00499 # define PLATFORM_ID "SINIX"
00500
00501 #elif defined(__UNIX_SV__)
00502 # define PLATFORM_ID "UNIX_SV"
00503
00504 #elif defined(_bsdos__)
00505 # define PLATFORM_ID "BSDOS"
00506
00507 #elif defined(_MPRAS) || defined(MPRAS)
00508 # define PLATFORM_ID "MP-RAS"
00509
00510 #elif defined(_osf) || defined(__osf__)
00511 # define PLATFORM_ID "OSF1"
00512
00513 #elif defined(_SCO_SV) || defined(SCO_SV) || defined(sco_sv)
00514 # define PLATFORM_ID "SCO_SV"
00515
00516 #elif defined(_ultrix) || defined(__ultrix__) || defined(_ULTRIX)
00517 # define PLATFORM_ID "ULTRIX"
00518
00519 #elif defined(__XENIX) || defined(_XENIX) || defined(XENIX)
00520 # define PLATFORM_ID "Xenix"
00521
00522 #elif defined(_WATCOMC__)
00523 # if defined(__LINUX__)
00524 # define PLATFORM_ID "Linux"
00525
00526 # elif defined(__DOS__)
00527 # define PLATFORM_ID "DOS"
00528
00529 # elif defined(__OS2__)
00530 # define PLATFORM_ID "OS2"
00531
00532 # elif defined(__WINDOWS__)
00533 # define PLATFORM_ID "Windows3x"
00534
00535 # elif defined(__VXWORKS__)
00536 # define PLATFORM_ID "VxWorks"
00537
00538 # else /* unknown platform */
00539 # define PLATFORM_ID
00540 # endif
00541
```

```
00542 #elif defined(__INTEGRITY)
00543 # if defined(INT_178B)
00544 # define PLATFORM_ID "Integrity178"
00545
00546 # else /* regular Integrity */
00547 # define PLATFORM_ID "Integrity"
00548 # endif
00549
00550 # elif defined(_ADI_COMPILER)
00551 # define PLATFORM_ID "ADSP"
00552
00553 #else /* unknown platform */
00554 # define PLATFORM_ID
00555
00556 #endif
00557
00558 /* For windows compilers MSVC and Intel we can determine
00559   the architecture of the compiler being used. This is because
00560   the compilers do not have flags that can change the architecture,
00561   but rather depend on which compiler is being used
00562 */
00563 #if defined(_WIN32) && defined(_MSC_VER)
00564 # if defined(_M_IA64)
00565 # define ARCHITECTURE_ID "IA64"
00566
00567 # elif defined(_M_ARM64EC)
00568 # define ARCHITECTURE_ID "ARM64EC"
00569
00570 # elif defined(_M_X64) || defined(_M_AMD64)
00571 # define ARCHITECTURE_ID "x64"
00572
00573 # elif defined(_M_IX86)
00574 # define ARCHITECTURE_ID "X86"
00575
00576 # elif defined(_M_ARM64)
00577 # define ARCHITECTURE_ID "ARM64"
00578
00579 # elif defined(_M_ARM)
00580 # if _M_ARM == 4
00581 # define ARCHITECTURE_ID "ARMV4I"
00582 # elif _M_ARM == 5
00583 # define ARCHITECTURE_ID "ARMV5I"
00584 # else
00585 # define ARCHITECTURE_ID "ARMV" STRINGIFY(_M_ARM)
00586 # endif
00587
00588 # elif defined(_M_MIPS)
00589 # define ARCHITECTURE_ID "MIPS"
00590
00591 # elif defined(_M_SH)
00592 # define ARCHITECTURE_ID "SHx"
00593
00594 # else /* unknown architecture */
00595 # define ARCHITECTURE_ID ""
00596 # endif
00597
00598 #elif defined(__WATCOMC__)
00599 # if defined(_M_I86)
00600 # define ARCHITECTURE_ID "I86"
00601
00602 # elif defined(_M_IX86)
00603 # define ARCHITECTURE_ID "X86"
00604
00605 # else /* unknown architecture */
00606 # define ARCHITECTURE_ID ""
00607 # endif
00608
00609 #elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
00610 # if defined(__ICCARM__)
00611 # define ARCHITECTURE_ID "ARM"
00612
00613 # elif defined(__ICCRX__)
00614 # define ARCHITECTURE_ID "RX"
00615
00616 # elif defined(__ICCRH850__)
00617 # define ARCHITECTURE_ID "RH850"
00618
00619 # elif defined(__ICCRL78__)
00620 # define ARCHITECTURE_ID "RL78"
00621
00622 # elif defined(__ICCRISCV__)
00623 # define ARCHITECTURE_ID "RISCV"
00624
00625 # elif defined(__ICCAVR__)
00626 # define ARCHITECTURE_ID "AVR"
00627
00628 # elif defined(__ICC430__)
```

```

00629 # define ARCHITECTURE_ID "MSP430"
00630
00631 # elif defined(__ICCV850__)
00632 # define ARCHITECTURE_ID "V850"
00633
00634 # elif defined(__ICC8051__)
00635 # define ARCHITECTURE_ID "8051"
00636
00637 # elif defined(__ICCSTM8__)
00638 # define ARCHITECTURE_ID "STM8"
00639
00640 # else /* unknown architecture */
00641 # define ARCHITECTURE_ID ""
00642 # endif
00643
00644 #elif defined(__ghs__)
00645 # if defined(__PPC64__)
00646 # define ARCHITECTURE_ID "PPC64"
00647
00648 # elif defined(__ppc__)
00649 # define ARCHITECTURE_ID "PPC"
00650
00651 # elif defined(__ARM__)
00652 # define ARCHITECTURE_ID "ARM"
00653
00654 # elif defined(__x86_64__)
00655 # define ARCHITECTURE_ID "x64"
00656
00657 # elif defined(__i386__)
00658 # define ARCHITECTURE_ID "X86"
00659
00660 # else /* unknown architecture */
00661 # define ARCHITECTURE_ID ""
00662 # endif
00663
00664 #elif defined(__TI_COMPILER_VERSION__)
00665 # if defined(__TI_ARM__)
00666 # define ARCHITECTURE_ID "ARM"
00667
00668 # elif defined(__MSP430__)
00669 # define ARCHITECTURE_ID "MSP430"
00670
00671 # elif defined(__TMS320C28XX__)
00672 # define ARCHITECTURE_ID "TMS320C28x"
00673
00674 # elif defined(__TMS320C6X__) || defined(__TMS320C6x__)
00675 # define ARCHITECTURE_ID "TMS320C6x"
00676
00677 # else /* unknown architecture */
00678 # define ARCHITECTURE_ID ""
00679 # endif
00680
00681 # elif defined(__ADSPSHARC__)
00682 # define ARCHITECTURE_ID "SHARC"
00683
00684 # elif defined(__ADSPBLACKFIN__)
00685 # define ARCHITECTURE_ID "Blackfin"
00686
00687 #else
00688 # define ARCHITECTURE_ID
00689 #endif
00690
00691 /* Convert integer to decimal digit literals. */
00692 #define DEC(n) \
00693   ('0' + (((n) / 10000000)%10)), \
00694   ('0' + (((n) / 1000000)%10)), \
00695   ('0' + (((n) / 100000)%10)), \
00696   ('0' + (((n) / 10000)%10)), \
00697   ('0' + (((n) / 1000)%10)), \
00698   ('0' + (((n) / 100)%10)), \
00699   ('0' + (((n) / 10)%10)), \
00700   ('0' + ((n) % 10))
00701
00702 /* Convert integer to hex digit literals. */
00703 #define HEX(n) \
00704   ('0' + (((n)>28 & 0xF))), \
00705   ('0' + (((n)>24 & 0xF))), \
00706   ('0' + (((n)>20 & 0xF))), \
00707   ('0' + (((n)>16 & 0xF))), \
00708   ('0' + (((n)>12 & 0xF))), \
00709   ('0' + (((n)>8 & 0xF))), \
00710   ('0' + (((n)>4 & 0xF))), \
00711   ('0' + ((n) & 0xF))
00712
00713 /* Construct a string literal encoding the version number. */
00714 #ifdef COMPILER_VERSION
00715 char const* info_version = "INFO" ":" compiler_version["COMPILER_VERSION "]";

```

```

00716
00717 /* Construct a string literal encoding the version number components. */
00718 #elif defined(COMPILER_VERSION_MAJOR)
00719 char const info_version[] = {
00720     'I', 'N', 'F', 'O', ':',
00721     'c', 'o', 'm', 'p', 'i', 'l', 'e', 'r', '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '[',
00722     COMPILER_VERSION_MAJOR,
00723     '#ifdef COMPILER_VERSION_MINOR
00724     '.', COMPILER_VERSION_MINOR,
00725     '#ifdef COMPILER_VERSION_PATCH
00726     '.', COMPILER_VERSION_PATCH,
00727     '#ifdef COMPILER_VERSION_TWEAK
00728     '.', COMPILER_VERSION_TWEAK,
00729     '#endif
00730     '#endif
00731     '#endif
00732     ']', '\0'};
00733 #endif
00734
00735 /* Construct a string literal encoding the internal version number. */
00736 #ifdef COMPILER_VERSION_INTERNAL
00737 char const info_version_internal[] = {
00738     'I', 'N', 'F', 'O', ':',
00739     'c', 'o', 'm', 'p', 'i', 'l', 'e', 'r', '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '_',
00740     'i', 'n', 't', 'e', 'r', 'n', 'a', 'l', '[',
00741     COMPILER_VERSION_INTERNAL, ']' , '\0'};
00742 #elif defined(COMPILER_VERSION_INTERNAL_STR)
00743 char const* info_version_internal = "INFO ":" compiler_version_internal["
    COMPILER_VERSION_INTERNAL_STR "]";
00744 #endif
00745
00746 /* Construct a string literal encoding the version number components. */
00747 #ifdef SIMULATE_VERSION_MAJOR
00748 char const info_simulate_version[] = {
00749     'I', 'N', 'F', 'O', ':',
00750     's', 'i', 'm', 'u', 'l', 'a', 't', 'e', 'r', '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '[',
00751     SIMULATE_VERSION_MAJOR,
00752     '#ifdef SIMULATE_VERSION_MINOR
00753     '.', SIMULATE_VERSION_MINOR,
00754     '#ifdef SIMULATE_VERSION_PATCH
00755     '.', SIMULATE_VERSION_PATCH,
00756     '#ifdef SIMULATE_VERSION_TWEAK
00757     '.', SIMULATE_VERSION_TWEAK,
00758     '#endif
00759     '#endif
00760     '#endif
00761     ']', '\0'};
00762 #endif
00763
00764 /* Construct the string literal in pieces to prevent the source from
00765     getting matched. Store it in a pointer rather than an array
00766     because some compilers will just produce instructions to fill the
00767     array rather than assigning a pointer to a static array. */
00768 char const* info_platform = "INFO ":" platform[" PLATFORM_ID "]";
00769 char const* info_arch = "INFO ":" arch[" ARCHITECTURE_ID "]";
00770
00771
00772
00773 #if !defined(__STDC__) && !defined(__clang__)
00774 # if defined(_MSC_VER) || defined(__ibmxl__) || defined(__IBMC__)
00775 # define C_VERSION "90"
00776 # else
00777 # define C_VERSION
00778 # endif
00779 #elif __STDC_VERSION__ > 201710L
00780 # define C_VERSION "23"
00781 #elif __STDC_VERSION__ >= 201710L
00782 # define C_VERSION "17"
00783 #elif __STDC_VERSION__ >= 201000L
00784 # define C_VERSION "11"
00785 #elif __STDC_VERSION__ >= 199901L
00786 # define C_VERSION "99"
00787 #else
00788 # define C_VERSION "90"
00789 #endif
00790 const char* info_language_standard_default =
00791     "INFO ":" standard_default[" C_VERSION "]";
00792
00793 const char* info_language_extensions_default = "INFO ":" extensions_default["
00794 #if (defined(__clang__) || defined(__GNUC__) || defined(__xlc__)) ||
00795     defined(__TI_COMPILER_VERSION__)) &&
00796     !defined(__STRICT_ANSI__)
00797     "ON"
00798 #else
00799     "OFF"
00800 #endif
00801 "]";

```

```

00802
00803 -----*/
00804
00805 #ifdef ID_VOID_MAIN
00806 void main() {}
00807 #else
00808 # if defined(__CLASSIC_C__)
00809 int main(argc, argv) int argc; char *argv[];
00810 # else
00811 int main(int argc, char* argv[])
00812 # endif
00813 {
00814     int require = 0;
00815     require += info_compiler[argc];
00816     require += info_platform[argc];
00817     require += info_arch[argc];
00818 #ifdef COMPILER_VERSION_MAJOR
00819     require += info_version[argc];
00820 #endif
00821 #ifdef COMPILER_VERSION_INTERNAL
00822     require += info_version_internal[argc];
00823 #endif
00824 #ifdef SIMULATE_ID
00825     require += info_simulate[argc];
00826 #endif
00827 #ifdef SIMULATE_VERSION_MAJOR
00828     require += info_simulate_version[argc];
00829 #endif
00830 #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
00831     require += info_cray[argc];
00832 #endif
00833     require += info_language_standard_default[argc];
00834     require += info_language_extensions_default[argc];
00835     (void)argv;
00836     return require;
00837 }
00838 #endif

```

6.4 src/build/CMakeFiles/3.24.2/CompilerIdCXX/CMakeCXXCompilerId.cpp File Reference

Macros

- `#define __has_include(x) 0`
- `#define COMPILER_ID ""`
- `#define STRINGIFY_HELPER(X) #X`
- `#define STRINGIFY(X) STRINGIFY_HELPER(X)`
- `#define PLATFORM_ID`
- `#define ARCHITECTURE_ID`
- `#define DEC(n)`
- `#define HEX(n)`
- `#define CXX_STD __cplusplus`

Functions

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

Variables

- `char const * info_compiler = "INFO ":" compiler[" COMPILER_ID "]"`
- `char const * info_platform = "INFO ":" platform[" PLATFORM_ID "]"`
- `char const * info_arch = "INFO ":" arch[" ARCHITECTURE_ID "]"`
- `const char * info_language_standard_default`
- `const char * info_language_extensions_default`

6.4.1 Macro Definition Documentation

6.4.1.1 __has_include

```
#define __has_include( x ) 0
```

Definition at line 11 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.2 ARCHITECTURE_ID

```
#define ARCHITECTURE_ID
```

Definition at line 673 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.3 COMPILER_ID

```
#define COMPILER_ID ""
```

Definition at line 408 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.4 CXX_STD

```
#define CXX_STD __cplusplus
```

Definition at line 771 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.5 DEC

```
#define DEC( n )
```

Value:

```
('0' + (((n) / 10000000)%10)), \
('0' + (((n) / 1000000)%10)), \
('0' + (((n) / 100000)%10)), \
('0' + (((n) / 10000)%10)), \
('0' + (((n) / 1000)%10)), \
('0' + (((n) / 100)%10)), \
('0' + (((n) / 10)%10)), \
('0' + ((n) % 10))
```

Definition at line 677 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.6 HEX

```
#define HEX(  
    n )
```

Value:

```
('0' + ((n) >> 28 & 0xF)), \  
'0' + ((n) >> 24 & 0xF)), \  
'0' + ((n) >> 20 & 0xF)), \  
'0' + ((n) >> 16 & 0xF)), \  
'0' + ((n) >> 12 & 0xF)), \  
'0' + ((n) >> 8 & 0xF)), \  
'0' + ((n) >> 4 & 0xF)), \  
'0' + ((n) & 0xF))
```

Definition at line 688 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.7 PLATFORM_ID

```
#define PLATFORM_ID
```

Definition at line 539 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.8 STRINGIFY

```
#define STRINGIFY(  
    X ) STRINGIFY_HELPER(X)
```

Definition at line 429 of file [CMakeCXXCompilerId.cpp](#).

6.4.1.9 STRINGIFY_HELPER

```
#define STRINGIFY_HELPER(  
    X ) #X
```

Definition at line 428 of file [CMakeCXXCompilerId.cpp](#).

6.4.2 Function Documentation

6.4.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Definition at line 802 of file [CMakeCXXCompilerId.cpp](#).

```
00803 {
00804     int require = 0;
00805     require += info_compiler[argc];
00806     require += info_platform[argc];
00807 #ifdef COMPILER_VERSION_MAJOR
00808     require += info_version[argc];
00809 #endif
00810 #ifdef COMPILER_VERSION_INTERNAL
00811     require += info_version_internal[argc];
00812 #endif
00813 #ifdef SIMULATE_ID
00814     require += info_simulate[argc];
00815 #endif
00816 #ifdef SIMULATE_VERSION_MAJOR
00817     require += info_simulate_version[argc];
00818 #endif
00819 #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
00820     require += info_cray[argc];
00821 #endif
00822     require += info_language_standard_default[argc];
00823     require += info_language_extensions_default[argc];
00824     (void)argv;
00825     return require;
00826 }
```

References [info_compiler](#), [info_language_extensions_default](#), [info_language_standard_default](#), and [info_platform](#).

6.4.3 Variable Documentation

6.4.3.1 info_arch

```
char const* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"
```

Definition at line 754 of file [CMakeCXXCompilerId.cpp](#).

6.4.3.2 info_compiler

```
char const* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"
```

Definition at line 415 of file [CMakeCXXCompilerId.cpp](#).

Referenced by [main\(\)](#).

6.4.3.3 info_language_extensions_default

```
const char* info_language_extensions_default
```

Initial value:

```
= "INFO" ":" "extensions_default["
    "OFF"
"]"
```

Definition at line 790 of file [CMakeCXXCompilerId.cpp](#).

Referenced by [main\(\)](#).

6.4.3.4 info_language_standard_default

```
const char* info_language_standard_default
```

Initial value:

```
= "INFO" ":" "standard_default["
    "98"
"]"
```

Definition at line 774 of file [CMakeCXXCompilerId.cpp](#).

Referenced by [main\(\)](#).

6.4.3.5 info_platform

```
char const* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"
```

Definition at line 753 of file [CMakeCXXCompilerId.cpp](#).

Referenced by [main\(\)](#).

6.5 CMakeCXXCompilerId.cpp

[Go to the documentation of this file.](#)

```
00001 /* This source file must have a .cpp extension so that all C++ compilers
00002     recognize the extension without flags. Borland does not know .cxx for
00003     example. */
00004 #ifndef __cplusplus
00005 # error "A C compiler has been selected for C++."
00006 #endif
00007
00008 #if !defined(__has_include)
00009 /* If the compiler does not have __has_include, pretend the answer is
00010     always no. */
00011 # define __has_include(x) 0
00012 #endif
00013
00014
00015 /* Version number components: V=Version, R=Revision, P=Patch
00016     Version date components: YYYY=Year, MM=Month, DD=Day */
00017
00018 #if defined(__COMO__)
00019 # define COMPILER_ID "Comeau"
00020 /* __COMO_VERSION__ = VRR */
```

```

00021 # define COMPILER_VERSION_MAJOR DEC(__COMO_VERSION__ / 100)
00022 # define COMPILER_VERSION_MINOR DEC(__COMO_VERSION__ % 100)
00023
00024 #elif defined(__INTEL_COMPILER) || defined(__ICC)
00025 # define COMPILER_ID "Intel"
00026 # if defined(_MSC_VER)
00027 # define SIMULATE_ID "MSVC"
00028 # endif
00029 # if defined(__GNUC__)
00030 # define SIMULATE_ID "GNU"
00031 # endif
00032 /* __INTEL_COMPILER = VRP prior to 2021, and then VVVV for 2021 and later,
00033 except that a few beta releases use the old format with V=2021. */
00034 # if __INTEL_COMPILER < 2021 || __INTEL_COMPILER == 202110 || __INTEL_COMPILER == 202111
00035 # define COMPILER_VERSION_MAJOR DEC(__INTEL_COMPILER/100)
00036 # define COMPILER_VERSION_MINOR DEC(__INTEL_COMPILER/10 % 10)
00037 # if defined(__INTEL_COMPILER_UPDATE)
00038 # define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER_UPDATE)
00039 # else
00040 # define COMPILER_VERSION_PATCH DEC(__INTEL_COMPILER % 10)
00041 # endif
00042 # else
00043 # define COMPILER_VERSION_MAJOR DEC(__INTEL_COMPILER)
00044 # define COMPILER_VERSION_MINOR DEC(__INTEL_COMPILER_UPDATE)
00045 /* The third version component from --version is an update index,
00046 but no macro is provided for it. */
00047 # define COMPILER_VERSION_PATCH DEC(0)
00048 # endif
00049 # if defined(__INTEL_COMPILER_BUILD_DATE)
00050 /* __INTEL_COMPILER_BUILD_DATE = YYYYMMDD */
00051 # define COMPILER_VERSION_TWEAK DEC(__INTEL_COMPILER_BUILD_DATE)
00052 # endif
00053 # if defined(_MSC_VER)
00054 /* _MSC_VER = VVRR */
00055 # define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
00056 # define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
00057 # endif
00058 # if defined(__GNUC__)
00059 # define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
00060 # elif defined(__GNUG__)
00061 # define SIMULATE_VERSION_MAJOR DEC(__GNUG__)
00062 # endif
00063 # if defined(__GNUC_MINOR__)
00064 # define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
00065 # endif
00066 # if defined(__GNUC_PATCHLEVEL__)
00067 # define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00068 # endif
00069
00070 #elif (defined(__clang__) && defined(__INTEL_CLANG_COMPILER)) || defined(__INTEL_LLVM_COMPILER)
00071 # define COMPILER_ID "IntellLVM"
00072 #if defined(_MSC_VER)
00073 # define SIMULATE_ID "MSVC"
00074 #endif
00075 #if defined(__GNUC__)
00076 # define SIMULATE_ID "GNU"
00077 #endif
00078 /* __INTEL_LLVM_COMPILER = VVVVRP prior to 2021.2.0, VVVVRRPP for 2021.2.0 and
00079 * later. Look for 6 digit vs. 8 digit version number to decide encoding.
00080 * VVVV is no smaller than the current year when a version is released.
00081 */
00082 #if __INTEL_LLVM_COMPILER < 1000000L
00083 # define COMPILER_VERSION_MAJOR DEC(__INTEL_LLVM_COMPILER/100)
00084 # define COMPILER_VERSION_MINOR DEC(__INTEL_LLVM_COMPILER/10 % 10)
00085 # define COMPILER_VERSION_PATCH DEC(__INTEL_LLVM_COMPILER % 10)
00086 #else
00087 # define COMPILER_VERSION_MAJOR DEC(__INTEL_LLVM_COMPILER/10000)
00088 # define COMPILER_VERSION_MINOR DEC(__INTEL_LLVM_COMPILER/100 % 100)
00089 # define COMPILER_VERSION_PATCH DEC(__INTEL_LLVM_COMPILER % 100)
00090 #endif
00091 #if defined(_MSC_VER)
00092 /* _MSC_VER = VVRR */
00093 # define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
00094 # define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
00095 #endif
00096 #if defined(__GNUC__)
00097 # define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
00098 #elif defined(__GNUG__)
00099 # define SIMULATE_VERSION_MAJOR DEC(__GNUG__)
00100 #endif
00101 #if defined(__GNUC_MINOR__)
00102 # define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
00103 #endif
00104 #if defined(__GNUC_PATCHLEVEL__)
00105 # define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00106 #endif
00107

```

```

00108 #elif defined(__PATHCC__)
00109 # define COMPILER_ID "PathScale"
00110 # define COMPILER_VERSION_MAJOR DEC(__PATHCC__)
00111 # define COMPILER_VERSION_MINOR DEC(__PATHCC_MINOR__)
00112 # if defined(__PATHCC_PATCHLEVEL__)
00113 # define COMPILER_VERSION_PATCH DEC(__PATHCC_PATCHLEVEL__)
00114 # endif
00115
00116 #elif defined(__BORLANDC__) && defined(__CODEGEARC_VERSION__)
00117 # define COMPILER_ID "Embarcadero"
00118 # define COMPILER_VERSION_MAJOR HEX(__CODEGEARC_VERSION__>>24 & 0x0FFF)
00119 # define COMPILER_VERSION_MINOR HEX(__CODEGEARC_VERSION__>>16 & 0x0FFF)
00120 # define COMPILER_VERSION_PATCH DEC(__CODEGEARC_VERSION__           & 0xFFFF)
00121
00122 #elif defined(__BORLANDC__)
00123 # define COMPILER_ID "Borland"
00124 /* __BORLANDC__ = 0xVRR */
00125 # define COMPILER_VERSION_MAJOR HEX(__BORLANDC__>>8)
00126 # define COMPILER_VERSION_MINOR HEX(__BORLANDC__ & 0xFF)
00127
00128 #elif defined(__WATCOMC__) && __WATCOMC__ < 1200
00129 # define COMPILER_ID "Watcom"
00130 /* __WATCOMC__ = VVRR */
00131 # define COMPILER_VERSION_MAJOR DEC(__WATCOMC__ / 100)
00132 # define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
00133 # if (__WATCOMC__ % 10) > 0
00134 # define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
00135 # endif
00136
00137 #elif defined(__WATCOMC__)
00138 # define COMPILER_ID "OpenWatcom"
00139 /* __WATCOMC__ = VVRP + 1100 */
00140 # define COMPILER_VERSION_MAJOR DEC((__WATCOMC__ - 1100) / 100)
00141 # define COMPILER_VERSION_MINOR DEC((__WATCOMC__ / 10) % 10)
00142 # if (__WATCOMC__ % 10) > 0
00143 # define COMPILER_VERSION_PATCH DEC(__WATCOMC__ % 10)
00144 # endif
00145
00146 #elif defined(__SUNPRO_CC)
00147 # define COMPILER_ID "SunPro"
00148 # if __SUNPRO_CC >= 0x5100
00149 /* __SUNPRO_CC = 0xVRRP */
00150 # define COMPILER_VERSION_MAJOR HEX(__SUNPRO_CC>>12)
00151 # define COMPILER_VERSION_MINOR HEX(__SUNPRO_CC>>4 & 0xFF)
00152 # define COMPILER_VERSION_PATCH HEX(__SUNPRO_CC      & 0xF)
00153 # else
00154 /* __SUNPRO_CC = 0xVRP */
00155 # define COMPILER_VERSION_MAJOR HEX(__SUNPRO_CC>>8)
00156 # define COMPILER_VERSION_MINOR HEX(__SUNPRO_CC>>4 & 0xF)
00157 # define COMPILER_VERSION_PATCH HEX(__SUNPRO_CC      & 0xF)
00158 # endif
00159
00160 #elif defined(__HP_aCC)
00161 # define COMPILER_ID "HP"
00162 /* __HP_aCC = VVRRPP */
00163 # define COMPILER_VERSION_MAJOR DEC(__HP_aCC/10000)
00164 # define COMPILER_VERSION_MINOR DEC(__HP_aCC/100 % 100)
00165 # define COMPILER_VERSION_PATCH DEC(__HP_aCC      % 100)
00166
00167 #elif defined(__DECCXX)
00168 # define COMPILER_ID "Compaq"
00169 /* __DECCXX_VER = VVRRTPPPP */
00170 # define COMPILER_VERSION_MAJOR DEC(__DECCXX_VER/10000000)
00171 # define COMPILER_VERSION_MINOR DEC(__DECCXX_VER/100000 % 100)
00172 # define COMPILER_VERSION_PATCH DEC(__DECCXX_VER      % 10000)
00173
00174 #elif defined(__IBMCPP__) && defined(__COMPILER_VER__)
00175 # define COMPILER_ID "zOS"
00176 /* __IBMCPP__ = VRP */
00177 # define COMPILER_VERSION_MAJOR DEC(__IBMCPP__/100)
00178 # define COMPILER_VERSION_MINOR DEC(__IBMCPP__/10 % 10)
00179 # define COMPILER_VERSION_PATCH DEC(__IBMCPP__      % 10)
00180
00181 #elif defined(__open_xl__) && defined(__clang__)
00182 # define COMPILER_ID "IBMclang"
00183 # define COMPILER_VERSION_MAJOR DEC(__open_xl_version__)
00184 # define COMPILER_VERSION_MINOR DEC(__open_xl_release__)
00185 # define COMPILER_VERSION_PATCH DEC(__open_xl_modification__)
00186 # define COMPILER_VERSION_TWEAK DEC(__open_xl_ptf_fix_level__)
00187
00188
00189 #elif defined(__ibmxml) && defined(__clang__)
00190 # define COMPILER_ID "XLClang"
00191 # define COMPILER_VERSION_MAJOR DEC(__ibmxml_version__)
00192 # define COMPILER_VERSION_MINOR DEC(__ibmxml_release__)
00193 # define COMPILER_VERSION_PATCH DEC(__ibmxml_modification__)
00194 # define COMPILER_VERSION_TWEAK DEC(__ibmxml_ptf_fix_level__)

```

```

00195
00196
00197 #elif defined(__IBMCPP__) && !defined(__COMPILER_VER__)
00198 # define COMPILER_ID "XL"
00199 /* __IBMCPP__ = VRP */
00200 # define COMPILER_VERSION_MAJOR DEC(__IBMCPP__/100)
00201 # define COMPILER_VERSION_MINOR DEC(__IBMCPP__/10 % 10)
00202 # define COMPILER_VERSION_PATCH DEC(__IBMCPP__ % 10)
00203
00204 #elif defined(__IBMCPP__) && !defined(__COMPILER_VER__)
00205 # define COMPILER_ID "VisualAge"
00206 /* __IBMCPP__ = VRP */
00207 # define COMPILER_VERSION_MAJOR DEC(__IBMCPP__/100)
00208 # define COMPILER_VERSION_MINOR DEC(__IBMCPP__/10 % 10)
00209 # define COMPILER_VERSION_PATCH DEC(__IBMCPP__ % 10)
00210
00211 #elif defined(__NVCOMPILER)
00212 # define COMPILER_ID "NVHPC"
00213 # define COMPILER_VERSION_MAJOR DEC(__NVCOMPILER_MAJOR__)
00214 # define COMPILER_VERSION_MINOR DEC(__NVCOMPILER_MINOR__)
00215 # if defined(__NVCOMPILER_PATCHLEVEL__)
00216 # define COMPILER_VERSION_PATCH DEC(__NVCOMPILER_PATCHLEVEL__)
00217 # endif
00218
00219 #elif defined(__PGI)
00220 # define COMPILER_ID "PGI"
00221 # define COMPILER_VERSION_MAJOR DEC(__PGIC__)
00222 # define COMPILER_VERSION_MINOR DEC(__PGIC_MINOR__)
00223 # if defined(__PGIC_PATCHLEVEL__)
00224 # define COMPILER_VERSION_PATCH DEC(__PGIC_PATCHLEVEL__)
00225 # endif
00226
00227 #elif defined(__CRAYC)
00228 # define COMPILER_ID "Cray"
00229 # define COMPILER_VERSION_MAJOR DEC(__RELEASE_MAJOR)
00230 # define COMPILER_VERSION_MINOR DEC(__RELEASE_MINOR)
00231
00232 #elif defined(__TI_COMPILER_VERSION__)
00233 # define COMPILER_ID "TI"
00234 /* __TI_COMPILER_VERSION__ = VVVRRPPP */
00235 # define COMPILER_VERSION_MAJOR DEC(__TI_COMPILER_VERSION__/1000000)
00236 # define COMPILER_VERSION_MINOR DEC(__TI_COMPILER_VERSION__/1000 % 1000)
00237 # define COMPILER_VERSION_PATCH DEC(__TI_COMPILER_VERSION__ % 1000)
00238
00239 #elif defined(__CLANG_FUJITSU)
00240 # define COMPILER_ID "FujitsuClang"
00241 # define COMPILER_VERSION_MAJOR DEC(__FCC_major__)
00242 # define COMPILER_VERSION_MINOR DEC(__FCC_minor__)
00243 # define COMPILER_VERSION_PATCH DEC(__FCC_patchlevel__)
00244 # define COMPILER_VERSION_INTERNAL_STR __clang_version__
00245
00246
00247 #elif defined(__FUJITSU)
00248 # define COMPILER_ID "Fujitsu"
00249 # if defined(__FCC_version__)
00250 # define COMPILER_VERSION __FCC_version__
00251 # elif defined(__FCC_major__)
00252 # define COMPILER_VERSION_MAJOR DEC(__FCC_major__)
00253 # define COMPILER_VERSION_MINOR DEC(__FCC_minor__)
00254 # define COMPILER_VERSION_PATCH DEC(__FCC_patchlevel__)
00255 # endif
00256 # if defined(__fcc_version)
00257 # define COMPILER_VERSION_INTERNAL DEC(__fcc_version)
00258 # elif defined(__FCC_VERSION)
00259 # define COMPILER_VERSION_INTERNAL DEC(__FCC_VERSION)
00260 # endif
00261
00262
00263 #elif defined(__ghs__)
00264 # define COMPILER_ID "GHS"
00265 /* __GHS_VERSION_NUMBER = VVVVRP */
00266 # ifdef __GHS_VERSION_NUMBER
00267 # define COMPILER_VERSION_MAJOR DEC(__GHS_VERSION_NUMBER / 100)
00268 # define COMPILER_VERSION_MINOR DEC(__GHS_VERSION_NUMBER / 10 % 10)
00269 # define COMPILER_VERSION_PATCH DEC(__GHS_VERSION_NUMBER % 10)
00270 # endif
00271
00272 #elif defined(__SCO_VERSION__)
00273 # define COMPILER_ID "SCO"
00274
00275 #elif defined(__ARMCC_VERSION) && !defined(__clang__)
00276 # define COMPILER_ID "ARMCC"
00277 #if __ARMCC_VERSION >= 1000000
00278 /* __ARMCC_VERSION = VRRPPP */
00279 # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/1000000)
00280 # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 100)
00281 # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION % 10000)

```

```
00282 #else
00283 /* __ARMCC_VERSION = VRPPP */
00284 # define COMPILER_VERSION_MAJOR DEC(__ARMCC_VERSION/100000)
00285 # define COMPILER_VERSION_MINOR DEC(__ARMCC_VERSION/10000 % 10)
00286 # define COMPILER_VERSION_PATCH DEC(__ARMCC_VERSION % 10000)
00287 #endif
00288
00289
00290 #elif defined(__clang__) && defined(__apple_build_version__)
00291 # define COMPILER_ID "AppleClang"
00292 # if defined(_MSC_VER)
00293 # define SIMULATE_ID "MSVC"
00294 # endif
00295 # define COMPILER_VERSION_MAJOR DEC(__clang_major__)
00296 # define COMPILER_VERSION_MINOR DEC(__clang_minor__)
00297 # define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
00298 # if defined(_MSC_VER)
00299 /* _MSC_VER = VVRR */
00300 # define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
00301 # define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
00302 # endif
00303 # define COMPILER_VERSION_TWEAK DEC(__apple_build_version__)
00304
00305 #elif defined(__clang__) && defined(__ARMCOMPILER_VERSION)
00306 # define COMPILER_ID "ARMClang"
00307 # define COMPILER_VERSION_MAJOR DEC(__ARMCOMPILER_VERSION/1000000)
00308 # define COMPILER_VERSION_MINOR DEC(__ARMCOMPILER_VERSION/10000 % 100)
00309 # define COMPILER_VERSION_PATCH DEC(__ARMCOMPILER_VERSION % 10000)
00310 # define COMPILER_VERSION_INTERNAL DEC(__ARMCOMPILER_VERSION)
00311
00312 #elif defined(__clang__)
00313 # define COMPILER_ID "Clang"
00314 # if defined(_MSC_VER)
00315 # define SIMULATE_ID "MSVC"
00316 # endif
00317 # define COMPILER_VERSION_MAJOR DEC(__clang_major__)
00318 # define COMPILER_VERSION_MINOR DEC(__clang_minor__)
00319 # define COMPILER_VERSION_PATCH DEC(__clang_patchlevel__)
00320 # if defined(_MSC_VER)
00321 /* _MSC_VER = VVRR */
00322 # define SIMULATE_VERSION_MAJOR DEC(_MSC_VER / 100)
00323 # define SIMULATE_VERSION_MINOR DEC(_MSC_VER % 100)
00324 # endif
00325
00326 #elif defined(__LCC__) && (defined(__GNUC__) || defined(__GNUG__) || defined(__MCST__))
00327 # define COMPILER_ID "LCC"
00328 # define COMPILER_VERSION_MAJOR DEC(1)
00329 # if defined(__LCC__)
00330 # define COMPILER_VERSION_MINOR DEC(__LCC__ - 100)
00331 # endif
00332 # if defined(__LCC_MINOR__)
00333 # define COMPILER_VERSION_PATCH DEC(__LCC_MINOR__)
00334 # endif
00335 # if defined(__GNUC__) && defined(__GNUC_MINOR__)
00336 # define SIMULATE_ID "GNU"
00337 # define SIMULATE_VERSION_MAJOR DEC(__GNUC__)
00338 # define SIMULATE_VERSION_MINOR DEC(__GNUC_MINOR__)
00339 # if defined(__GNUC_PATCHLEVEL__)
00340 # define SIMULATE_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00341 # endif
00342 # endif
00343
00344 #elif defined(__GNUC__) || defined(__GNUG__)
00345 # define COMPILER_ID "GNU"
00346 # if defined(__GNUC__)
00347 # define COMPILER_VERSION_MAJOR DEC(__GNUC__)
00348 # else
00349 # define COMPILER_VERSION_MAJOR DEC(__GNUG__)
00350 # endif
00351 # if defined(__GNUC_MINOR__)
00352 # define COMPILER_VERSION_MINOR DEC(__GNUC_MINOR__)
00353 # endif
00354 # if defined(__GNUC_PATCHLEVEL__)
00355 # define COMPILER_VERSION_PATCH DEC(__GNUC_PATCHLEVEL__)
00356 # endif
00357
00358 #elif defined(_MSC_VER)
00359 # define COMPILER_ID "MSVC"
00360 /* _MSC_VER = VVRR */
00361 # define COMPILER_VERSION_MAJOR DEC(_MSC_VER / 100)
00362 # define COMPILER_VERSION_MINOR DEC(_MSC_VER % 100)
00363 # if defined(_MSC_FULL_VER)
00364 # if _MSC_VER >= 1400
00365 /* _MSC_FULL_VER = VVRRPPPP */
00366 # define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 100000)
00367 # else
00368 /* _MSC_FULL_VER = VVRRPPPP */
```

```

00369 # define COMPILER_VERSION_PATCH DEC(_MSC_FULL_VER % 10000)
00370 # endif
00371 # endif
00372 # if defined(_MSC_BUILD)
00373 # define COMPILER_VERSION_TWEAK DEC(_MSC_BUILD)
00374 # endif
00375
00376 #elif defined(_ADI_COMPILER)
00377 # define COMPILER_ID "ADSP"
00378 #if defined(__VERSIONNUM__)
00379 /* __VERSIONNUM__ = 0xVVRPPTT */
00380 # define COMPILER_VERSION_MAJOR DEC(__VERSIONNUM__ >> 24 & 0xFF)
00381 # define COMPILER_VERSION_MINOR DEC(__VERSIONNUM__ >> 16 & 0xFF)
00382 # define COMPILER_VERSION_PATCH DEC(__VERSIONNUM__ >> 8 & 0xFF)
00383 # define COMPILER_VERSION_TWEAK DEC(__VERSIONNUM__ & 0xFF)
00384#endif
00385
00386 #elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
00387 # define COMPILER_ID "IAR"
00388 #if defined(__VER__) && defined(__ICCARM__)
00389 # define COMPILER_VERSION_MAJOR DEC((__VER__) / 1000000)
00390 # define COMPILER_VERSION_MINOR DEC(((__VER__) / 1000) % 1000)
00391 # define COMPILER_VERSION_PATCH DEC((__VER__) % 1000)
00392 # define COMPILER_VERSION_INTERNAL DEC(__IAR_SYSTEMS_ICC__)
00393 #elif defined(__VER__) && (defined(__ICCAVR__) || defined(__ICCRX__) || defined(__ICCRH850__)
00394     || defined(__ICCRXL78__) || defined(__ICC430__) || defined(__ICCRISCV__) || defined(__ICCV850__)
00395     || defined(__ICC8051__) || defined(__ICCSTM8__))
00396 # define COMPILER_VERSION_MAJOR DEC(__VER__ / 100)
00397 # define COMPILER_VERSION_MINOR DEC(__VER__ - (((__VER__ / 100)*100)))
00398 # define COMPILER_VERSION_PATCH DEC(__SUBVERSION__)
00399 #endif
00400
00401 /* These compilers are either not known or too old to define an
00402 identification macro. Try to identify the platform and guess that
00403 it is the native compiler. */
00404 #elif defined(__hpux) || defined(__hpua)
00405 # define COMPILER_ID "HP"
00406
00407 #else /* unknown compiler */
00408 # define COMPILER_ID ""
00409#endif
00410
00411 /* Construct the string literal in pieces to prevent the source from
00412 getting matched. Store it in a pointer rather than an array
00413 because some compilers will just produce instructions to fill the
00414 array rather than assigning a pointer to a static array. */
00415 char const* info_compiler = "INFO ":"compiler[" COMPILER_ID "]";
00416 #ifdef SIMULATE_ID
00417 char const* info_simulate = "INFO ":"simulate[" SIMULATE_ID "]";
00418#endif
00419
00420 #ifdef __QNXNTO__
00421 char const* qnxnto = "INFO ":" "qnxnto[]";
00422#endif
00423
00424 #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
00425 char const *info_cray = "INFO ":" "compiler_wrapper[CrayPrgEnv]";
00426#endif
00427
00428 #define STRINGIFY_HELPER(X) #X
00429 #define STRINGIFY(X) STRINGIFY_HELPER(X)
00430
00431 /* Identify known platforms by name. */
00432 #if defined(__linux) || defined(__linux__) || defined(linux)
00433 # define PLATFORM_ID "Linux"
00434
00435 #elif defined(__MSYS__)
00436 # define PLATFORM_ID "MSYS"
00437
00438 #elif defined(__CYGWIN__)
00439 # define PLATFORM_ID "Cygwin"
00440
00441 #elif defined(__MINGW32__)
00442 # define PLATFORM_ID "MinGW"
00443
00444 #elif defined(__APPLE__)
00445 # define PLATFORM_ID "Darwin"
00446
00447 #elif defined(__WIN32) || defined(__WIN32__) || defined(WIN32)
00448 # define PLATFORM_ID "Windows"
00449
00450 #elif defined(__FreeBSD__) || defined(__FreeBSD)
00451 # define PLATFORM_ID "FreeBSD"
00452
00453 #elif defined(__NetBSD__) || defined(__NetBSD)

```

```
00454 # define PLATFORM_ID "NetBSD"
00455
00456 #elif defined(__OpenBSD__) || defined(__OPENBSD)
00457 # define PLATFORM_ID "OpenBSD"
00458
00459 #elif defined(_sun) || defined(sun)
00460 # define PLATFORM_ID "SunOS"
00461
00462 #elif defined(_AIX) || defined(_AIX) || defined(_AIX_) || defined(_aix) || defined(_aix_)
00463 # define PLATFORM_ID "AIX"
00464
00465 #elif defined(_hpx) || defined(_hpx__)
00466 # define PLATFORM_ID "HP-UX"
00467
00468 #elif defined(__HAIKU__)
00469 # define PLATFORM_ID "Haiku"
00470
00471 #elif defined(_BeOS) || defined(_BEOS_) || defined(_BEOS)
00472 # define PLATFORM_ID "BeOS"
00473
00474 #elif defined(__QNX__) || defined(__QNXNTO__)
00475 # define PLATFORM_ID "QNX"
00476
00477 #elif defined(_tru64) || defined(_tru64) || defined(_TRU64__)
00478 # define PLATFORM_ID "Tru64"
00479
00480 #elif defined(_riscos) || defined(_riscos__)
00481 # define PLATFORM_ID "RISCos"
00482
00483 #elif defined(__sinix) || defined(__sinix__) || defined(__SINIX__)
00484 # define PLATFORM_ID "SINIX"
00485
00486 #elif defined(__UNIX_SV__)
00487 # define PLATFORM_ID "UNIX_SV"
00488
00489 #elif defined(_bsdos__)
00490 # define PLATFORM_ID "BSDOS"
00491
00492 #elif defined(_MPRAS) || defined(MPRAS)
00493 # define PLATFORM_ID "MP-RAS"
00494
00495 #elif defined(_osf) || defined(_osf__)
00496 # define PLATFORM_ID "OSF1"
00497
00498 #elif defined(_SCO_SV) || defined(SCO_SV) || defined(sco_sv)
00499 # define PLATFORM_ID "SCO_SV"
00500
00501 #elif defined(_ultrix) || defined(_ultrix_) || defined(_ULTRIX)
00502 # define PLATFORM_ID "ULTRIX"
00503
00504 #elif defined(__XENIX__) || defined(_XENIX) || defined(XENIX)
00505 # define PLATFORM_ID "Xenix"
00506
00507 #elif defined(__WATCOMC__)
00508 # if defined(__LINUX__)
00509 # define PLATFORM_ID "Linux"
00510
00511 # elif defined(__DOS__)
00512 # define PLATFORM_ID "DOS"
00513
00514 # elif defined(__OS2__)
00515 # define PLATFORM_ID "OS2"
00516
00517 # elif defined(__WINDOWS__)
00518 # define PLATFORM_ID "Windows3x"
00519
00520 # elif defined(__VXWORKS__)
00521 # define PLATFORM_ID "VxWorks"
00522
00523 # else /* unknown platform */
00524 # define PLATFORM_ID
00525 # endif
00526
00527 #elif defined(__INTEGRITY)
00528 # if defined(INT_178B)
00529 # define PLATFORM_ID "Integrity178"
00530
00531 # else /* regular Integrity */
00532 # define PLATFORM_ID "Integrity"
00533 # endif
00534
00535 # elif defined(__ADI_COMPILER)
00536 # define PLATFORM_ID "ADSP"
00537
00538 #else /* unknown platform */
00539 # define PLATFORM_ID
00540
```

```
00541 #endif
00542
00543 /* For windows compilers MSVC and Intel we can determine
00544   the architecture of the compiler being used. This is because
00545   the compilers do not have flags that can change the architecture,
00546   but rather depend on which compiler is being used
00547 */
00548 #if defined(_WIN32) && defined(_MSC_VER)
00549 # if defined(_M_IA64)
00550 #  define ARCHITECTURE_ID "IA64"
00551
00552 # elif defined(_M_ARM64EC)
00553 #  define ARCHITECTURE_ID "ARM64EC"
00554
00555 # elif defined(_M_X64) || defined(_M_AMD64)
00556 #  define ARCHITECTURE_ID "x64"
00557
00558 # elif defined(_M_IX86)
00559 #  define ARCHITECTURE_ID "X86"
00560
00561 # elif defined(_M_ARM64)
00562 #  define ARCHITECTURE_ID "ARM64"
00563
00564 # elif defined(_M_ARM)
00565 #  if _M_ARM == 4
00566 #    define ARCHITECTURE_ID "ARMV4I"
00567 #  elif _M_ARM == 5
00568 #    define ARCHITECTURE_ID "ARMV5I"
00569 #  else
00570 #    define ARCHITECTURE_ID "ARMV" STRINGIFY(_M_ARM)
00571 #  endif
00572
00573 # elif defined(_M_MIPS)
00574 #  define ARCHITECTURE_ID "MIPS"
00575
00576 # elif defined(_M_SH)
00577 #  define ARCHITECTURE_ID "SHx"
00578
00579 # else /* unknown architecture */
00580 #  define ARCHITECTURE_ID ""
00581 # endif
00582
00583 #elif defined(__WATCOMC__)
00584 # if defined(_M_I86)
00585 #  define ARCHITECTURE_ID "I86"
00586
00587 # elif defined(_M_IX86)
00588 #  define ARCHITECTURE_ID "X86"
00589
00590 # else /* unknown architecture */
00591 #  define ARCHITECTURE_ID ""
00592 # endif
00593
00594 #elif defined(__IAR_SYSTEMS_ICC__) || defined(__IAR_SYSTEMS_ICC)
00595 # if defined(__ICCARM__)
00596 #  define ARCHITECTURE_ID "ARM"
00597
00598 # elif defined(__ICCRX__)
00599 #  define ARCHITECTURE_ID "RX"
00600
00601 # elif defined(__ICCRH850__)
00602 #  define ARCHITECTURE_ID "RH850"
00603
00604 # elif defined(__ICCRL78__)
00605 #  define ARCHITECTURE_ID "RL78"
00606
00607 # elif defined(__ICCRISCV__)
00608 #  define ARCHITECTURE_ID "RISCV"
00609
00610 # elif defined(__ICCAVR__)
00611 #  define ARCHITECTURE_ID "AVR"
00612
00613 # elif defined(__ICC430__)
00614 #  define ARCHITECTURE_ID "MSP430"
00615
00616 # elif defined(__ICCV850__)
00617 #  define ARCHITECTURE_ID "V850"
00618
00619 # elif defined(__ICC8051__)
00620 #  define ARCHITECTURE_ID "8051"
00621
00622 # elif defined(__ICCSTM8__)
00623 #  define ARCHITECTURE_ID "STM8"
00624
00625 # else /* unknown architecture */
00626 #  define ARCHITECTURE_ID ""
00627 # endif
```

```

00628
00629 #elif defined(__ghs__)
00630 # if defined(__PPC64__)
00631 # define ARCHITECTURE_ID "PPC64"
00632
00633 # elif defined(__ppc__)
00634 # define ARCHITECTURE_ID "PPC"
00635
00636 # elif defined(__ARM__)
00637 # define ARCHITECTURE_ID "ARM"
00638
00639 # elif defined(__x86_64__)
00640 # define ARCHITECTURE_ID "x64"
00641
00642 # elif defined(__i386__)
00643 # define ARCHITECTURE_ID "X86"
00644
00645 # else /* unknown architecture */
00646 # define ARCHITECTURE_ID ""
00647 # endif
00648
00649 #elif defined(__TI_COMPILER_VERSION__)
00650 # if defined(__TI_ARM__)
00651 # define ARCHITECTURE_ID "ARM"
00652
00653 # elif defined(__MSP430__)
00654 # define ARCHITECTURE_ID "MSP430"
00655
00656 # elif defined(__TMS320C28XX__)
00657 # define ARCHITECTURE_ID "TMS320C28x"
00658
00659 # elif defined(__TMS320C6X__)
00660 # define ARCHITECTURE_ID "TMS320C6x"
00661
00662 # else /* unknown architecture */
00663 # define ARCHITECTURE_ID ""
00664 # endif
00665
00666 # elif defined(__ADSPSHARC__)
00667 # define ARCHITECTURE_ID "SHARC"
00668
00669 # elif defined(__ADSPBLACKFIN__)
00670 # define ARCHITECTURE_ID "Blackfin"
00671
00672 #else
00673 # define ARCHITECTURE_ID
00674 #endif
00675
00676 /* Convert integer to decimal digit literals. */
00677 #define DEC(n) \
00678     ('0' + (((n) / 10000000)%10)), \
00679     ('0' + (((n) / 1000000)%10)), \
00680     ('0' + (((n) / 100000)%10)), \
00681     ('0' + (((n) / 10000)%10)), \
00682     ('0' + (((n) / 1000)%10)), \
00683     ('0' + (((n) / 100)%10)), \
00684     ('0' + (((n) / 10)%10)), \
00685     ('0' + ((n) % 10))
00686
00687 /* Convert integer to hex digit literals. */
00688 #define HEX(n) \
00689     ('0' + ((n)>>28 & 0xF)), \
00690     ('0' + ((n)>>24 & 0xF)), \
00691     ('0' + ((n)>>20 & 0xF)), \
00692     ('0' + ((n)>>16 & 0xF)), \
00693     ('0' + ((n)>>12 & 0xF)), \
00694     ('0' + ((n)>>8 & 0xF)), \
00695     ('0' + ((n)>>4 & 0xF)), \
00696     ('0' + ((n) & 0xF))
00697
00698 /* Construct a string literal encoding the version number. */
00699 #ifdef COMPILER_VERSION
00700 char const* info_version = "INFO" ":" "compiler_version[" COMPILER_VERSION "]";
00701
00702 /* Construct a string literal encoding the version number components. */
00703 #elif defined(COMPILER_VERSION_MAJOR)
00704 char const info_version[] = {
00705     'I', 'N', 'F', 'O', ':',
00706     'c', 'o', 'm', 'p', 'i', 'l', 'e', 'r', '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '[',
00707     COMPILER_VERSION_MAJOR,
00708     '#ifdef COMPILER_VERSION_MINOR',
00709     '.', COMPILER_VERSION_MINOR,
00710     '# ifdef COMPILER_VERSION_PATCH',
00711     '.', COMPILER_VERSION_PATCH,
00712     '# ifdef COMPILER_VERSION_TWEAK',
00713     '.', COMPILER_VERSION_TWEAK,
00714     '# endif'

```

```

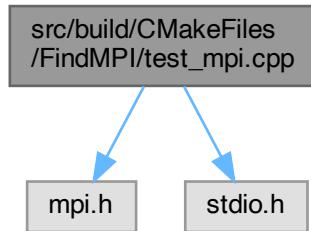
00715 # endif
00716 # endif
00717 ']' , '\0' );
00718 #endif
00719
00720 /* Construct a string literal encoding the internal version number. */
00721 #ifdef COMPILER_VERSION_INTERNAL
00722 char const info_version_internal[] = {
00723   'I', 'N', 'F', '0', ':',
00724   'c', 'o', 'm', 'p', 'i', 'l', 'e', 'r', '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '_',
00725   'i', 'n', 't', 'e', 'r', 'n', 'a', 'l', '[',
00726   COMPILER_VERSION_INTERNAL, ']' , '\0' );
00727 #elif defined(COMPILER_VERSION_INTERNAL_STR)
00728 char const* info_version_internal = "INFO ":" compiler_version_internal[" COMPILER_VERSION_INTERNAL_STR "]";
00729#endif
00730
00731 /* Construct a string literal encoding the version number components. */
00732 #ifdef SIMULATE_VERSION_MAJOR
00733 char const info_simulate_version[] = {
00734   'I', 'N', 'F', '0', ':',
00735   's', 'i', 'm', 'u', 'l', 'a', 't', 'e', '_', 'v', 'e', 'r', 's', 'i', 'o', 'n', '[',
00736   SIMULATE_VERSION_MAJOR,
00737 # ifdef SIMULATE_VERSION_MINOR
00738   '.', SIMULATE_VERSION_MINOR,
00739 # ifdef SIMULATE_VERSION_PATCH
00740   '.', SIMULATE_VERSION_PATCH,
00741 # ifdef SIMULATE_VERSION_TWEAK
00742   '.', SIMULATE_VERSION_TWEAK,
00743 # endif
00744 # endif
00745 # endif
00746 ']' , '\0' );
00747#endif
00748
00749 /* Construct the string literal in pieces to prevent the source from
00750   getting matched. Store it in a pointer rather than an array
00751   because some compilers will just produce instructions to fill the
00752   array rather than assigning a pointer to a static array. */
00753 char const* info_platform = "INFO ":" platform[" PLATFORM_ID "]";
00754 char const* info_arch = "INFO ":" arch[" ARCHITECTURE_ID "]";
00755
00756
00757
00758 #if defined(__INTEL_COMPILER) && defined(_MSVC_LANG) && _MSVC_LANG < 201403L
00759 # if defined(__INTEL_CXX11_MODE__)
00760 #   if defined(__cpp_aggregate_nsdmi)
00761 #     define CXX_STD 201402L
00762 #   else
00763 #     define CXX_STD 201103L
00764 #   endif
00765 # else
00766 #   define CXX_STD 199711L
00767 # endif
00768 #elif defined(_MSC_VER) && defined(_MSVC_LANG)
00769 # define CXX_STD _MSVC_LANG
00770 #else
00771 # define CXX_STD __cplusplus
00772#endif
00773
00774 const char* info_language_standard_default = "INFO ":" standard_default["
00775 #if CXX_STD > 202002L
00776   "23"
00777 #elif CXX_STD > 201703L
00778   "20"
00779 #elif CXX_STD >= 201703L
00780   "17"
00781 #elif CXX_STD >= 201402L
00782   "14"
00783 #elif CXX_STD >= 201103L
00784   "11"
00785 #else
00786   "98"
00787 #endif
00788 "];
00789
00790 const char* info_language_extensions_default = "INFO ":" extensions_default["
00791 #if (defined(__clang__)) || defined(__GNUC__) || defined(__xlc__) ||
00792   defined(__TI_COMPILER_VERSION__) &&
00793   !defined(__STRICT_ANSI__)
00794   "ON"
00795 #else
00796   "OFF"
00797 #endif
00798 "];
00799
00800 /*-----*/

```

```
00801 int main(int argc, char* argv[])
00802 {
00803     int require = 0;
00804     require += info_compiler[argc];
00805     require += info_platform[argc];
00806     require += COMPILER_VERSION_MAJOR;
00807     require += info_version[argc];
00808     #endif
00809     #endif
00810     #ifdef COMPILER_VERSION_INTERNAL
00811         require += info_version_internal[argc];
00812     #endif
00813     #ifdef SIMULATE_ID
00814         require += info_simulate[argc];
00815     #endif
00816     #ifdef SIMULATE_VERSION_MAJOR
00817         require += info_simulate_version[argc];
00818     #endif
00819     #if defined(__CRAYXT_COMPUTE_LINUX_TARGET)
00820         require += info_cray[argc];
00821     #endif
00822     require += info_language_standard_default[argc];
00823     require += info_language_extensions_default[argc];
00824     (void)argv;
00825     return require;
00826 }
```

6.6 src/build/CMakeFiles/FindMPI/test_mpi.cpp File Reference

```
#include <mpi.h>
#include <stdio.h>
Include dependency graph for test_mpi.cpp:
```



Functions

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

6.6.1 Function Documentation

6.6.1.1 main()

```

int main (
    int argc,
    char * argv[] )

```

Definition at line 21 of file [test_mpi.cpp](#).

```

00022 {
00023 #if defined(MPI_VERSION) && defined(MPI_SUBVERSION)
00024 # ifdef __cplusplus
00025     std::puts(mpiver_str);
00026 # else
00027     puts(mpiver_str);
00028 # endif
00029 #endif
00030 #ifdef TEST_MPI_MPICXX
00031     MPI::MPI_Init(&argc, &argv);
00032     MPI::MPI_Finalize();
00033 #else
00034     MPI_Init(&argc, &argv);
00035     MPI_Finalize();
00036 #endif
00037     return 0;
00038 }

```

6.7 test_mpi.cpp

[Go to the documentation of this file.](#)

```

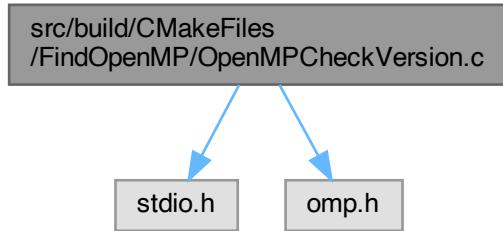
00001 #include <mpi.h>
00002
00003 #ifdef __cplusplus
00004 # include <cstdio>
00005 #else
00006 # include <stdio.h>
00007 #endif
00008
00009 #if defined(MPI_VERSION) && defined(MPI_SUBVERSION)
00010 static const char mpiver_str[] = { 'I', 'N',
00011                         'F', 'O',
00012                         ':', 'M',
00013                         'P', 'I',
00014                         '-', 'V',
00015                         'E', 'R',
00016                         '[', ('0' + MPI_VERSION),
00017                         '.', ('0' + MPI_SUBVERSION),
00018                         ']', '\0' };
00019 #endif
00020
00021 int main(int argc, char* argv[])
00022 {
00023 #if defined(MPI_VERSION) && defined(MPI_SUBVERSION)
00024 # ifdef __cplusplus
00025     std::puts(mpiver_str);
00026 # else
00027     puts(mpiver_str);
00028 # endif
00029 #endif
00030 #ifdef TEST_MPI_MPICXX
00031     MPI::MPI_Init(&argc, &argv);
00032     MPI::MPI_Finalize();
00033 #else
00034     MPI_Init(&argc, &argv);
00035     MPI_Finalize();
00036 #endif
00037     return 0;
00038 }

```

6.8 src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c File Reference

```
#include <stdio.h>
#include <omp.h>
```

Include dependency graph for OpenMPCheckVersion.c:



Functions

- int [main](#) (void)

Variables

- const char [ompver_str](#) []

6.8.1 Function Documentation

6.8.1.1 main()

```
int main (
    void )
```

Definition at line 13 of file [OpenMPCheckVersion.c](#).

```
00014 {
00015     puts(ompver_str);
00016     return 0;
00017 }
```

References [ompver_str](#).

6.8.2 Variable Documentation

6.8.2.1 ompver_str

```
const char ompver_str[ ]
```

Initial value:

```
= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'P', 'r', 'd', 'a', 't', 'e', '[',
    ('0' + ((OPENMP/100000)%10)),
    ('0' + ((OPENMP/10000)%10)),
    ('0' + ((OPENMP/1000)%10)),
    ('0' + ((OPENMP/100)%10)),
    ('0' + ((OPENMP/10)%10)),
    ('0' + ((OPENMP/1)%10)),
    ']' , '\0' }
```

Definition at line 4 of file OpenMPCheckVersion.c.

Referenced by [main\(\)](#).

6.9 OpenMPCheckVersion.c

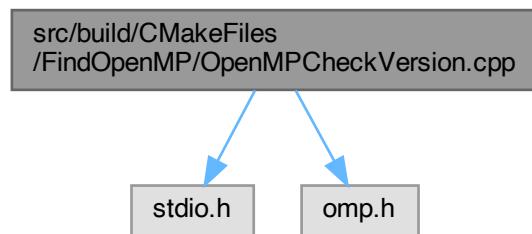
[Go to the documentation of this file.](#)

```
00001
00002 #include <stdio.h>
00003 #include <omp.h>
00004 const char ompver_str[] = { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
00005     'P', 'r', 'd', 'a', 't', 'e', '[',
00006     ('0' + ((OPENMP/100000)%10)),
00007     ('0' + ((OPENMP/10000)%10)),
00008     ('0' + ((OPENMP/1000)%10)),
00009     ('0' + ((OPENMP/100)%10)),
00010     ('0' + ((OPENMP/10)%10)),
00011     ('0' + ((OPENMP/1)%10)),
00012     ']' , '\0' };
00013 int main(void)
00014 {
00015     puts(ompver_str);
00016     return 0;
00017 }
```

6.10 src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp File Reference

```
#include <stdio.h>
#include <omp.h>
```

Include dependency graph for OpenMPCheckVersion.cpp:



Functions

- int `main` (void)

Variables

- const char `ompver_str` []

6.10.1 Function Documentation

6.10.1.1 `main()`

```
int main (
    void )
```

Definition at line 13 of file [OpenMPCheckVersion.cpp](#).

```
00014 {
00015     puts(ompver_str);
00016     return 0;
00017 }
```

References [ompver_str](#).

6.10.2 Variable Documentation

6.10.2.1 `ompver_str`

```
const char ompver_str[]
```

Initial value:

```
= { 'I', 'N', 'F', 'O', ':', 'O', 'p', 'e', 'n', 'M',
    'p', 'r', 'd', 'a', 't', 'e', '[',
    ('0' + ((_OPENMP/100000)%10)),
    ('0' + ((_OPENMP/10000)%10)),
    ('0' + ((_OPENMP/1000)%10)),
    ('0' + ((_OPENMP/100)%10)),
    ('0' + ((_OPENMP/10)%10)),
    ('0' + ((_OPENMP/1)%10)),
    ']', '\0' }
```

Definition at line 4 of file [OpenMPCheckVersion.cpp](#).

Referenced by [main\(\)](#).

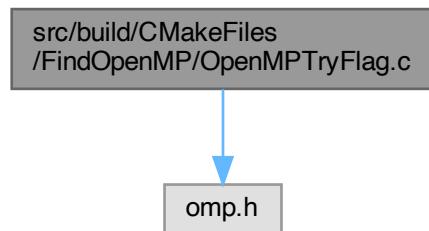
6.11 OpenMPCheckVersion.cpp

[Go to the documentation of this file.](#)

```
00001
00002 #include <stdio.h>
00003 #include <omp.h>
00004 const char ompver_str[] = { 'I', 'N', 'F', 'O', ';', 'O', 'p', 'e', 'n', 'M',
00005     'P', ' ', 'd', 'a', 't', 'e', '[', '0' + ((OPENMP/100000)%10),
00006     ('0' + ((OPENMP/10000)%10)),
00007     ('0' + ((OPENMP/1000)%10)),
00008     ('0' + ((OPENMP/100)%10)),
00009     ('0' + ((OPENMP/10)%10)),
00010     ('0' + ((OPENMP/1)%10)),
00011     ']', '\0' };
00012
00013 int main(void)
00014 {
00015     puts(ompver_str);
00016     return 0;
00017 }
```

6.12 src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c File Reference

```
#include <omp.h>
Include dependency graph for OpenMPTryFlag.c:
```



Functions

- int `main` (void)

6.12.1 Function Documentation

6.12.1.1 main()

```
int main (
    void )
```

Definition at line 3 of file OpenMPTryFlag.c.

```
00003     {
00004 #ifdef _OPENMP
00005     omp_get_max_threads();
00006     return 0;
00007 #elif defined(__HIP_DEVICE_COMPILE__)
00008     return 0;
00009 #else
00010     breaks_on_purpose
00011 #endif
00012 }
```

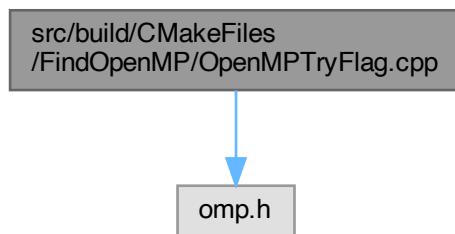
6.13 OpenMPTryFlag.c

[Go to the documentation of this file.](#)

```
00001
00002 #include <omp.h>
00003 int main(void) {
00004 #ifdef _OPENMP
00005     omp_get_max_threads();
00006     return 0;
00007 #elif defined(__HIP_DEVICE_COMPILE__)
00008     return 0;
00009 #else
00010     breaks_on_purpose
00011 #endif
00012 }
```

6.14 src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp File Reference

```
#include <omp.h>
Include dependency graph for OpenMPTryFlag.cpp:
```



Functions

- int [main \(void\)](#)

6.14.1 Function Documentation

6.14.1.1 main()

```
int main (
    void )
```

[Definition at line 3 of file OpenMPTryFlag.cpp.](#)

```
00003     {
00004 #ifdef __OPENMP
00005     omp_get_max_threads();
00006     return 0;
00007 #elif defined(__HIP_DEVICE_COMPILE__)
00008     return 0;
00009 #else
00010     breaks_on_purpose
00011 #endif
00012 }
```

6.15 OpenMPTryFlag.cpp

[Go to the documentation of this file.](#)

```
00001
00002 #include <omp.h>
00003 int main(void) {
00004 #ifdef __OPENMP
00005     omp_get_max_threads();
00006     return 0;
00007 #elif defined(__HIP_DEVICE_COMPILE__)
00008     return 0;
00009 #else
00010     breaks_on_purpose
00011 #endif
00012 }
```

6.16 src/build/CMakeFiles/hewes.dir/DerivationStencils.cpp.o.d File Reference

6.17 DerivationStencils.cpp.o.d

[Go to the documentation of this file.](#)

```
00001 CMakeFiles/hewes.dir/DerivationStencils.cpp.o: \
00002   /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.cpp \
00003   /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.h \
00004   /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00005   /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00006   /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00007   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00008   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00009   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config_site \
00010   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00011   /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/float.h \
00012   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00013   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00014   /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stddef.h \
00015   /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/__stddef_max_align_t.h \
00016   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_nullptr \
00017   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00018   /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00019   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00020   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
```

```

00021 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \
00022 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00023 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00024 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00025 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00026 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00027 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00028 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00029 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/cdefs.h \
00030 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00031 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00032 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00033 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00034 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00035 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \
00036 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00037 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00038 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00039 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00040 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00041 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00042 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00043 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00044 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h

```

6.18 src/build/CMakeFiles/hewes.dir/ICSetters.cpp.o.d File Reference

6.19 ICSetters.cpp.o.d

[Go to the documentation of this file.](#)

```

00001 CMakeFiles/hewes.dir/ICSetters.cpp.o: \
00002 /Users/andi/Documents/ruhl_gits/hewes/src/ICSetters.cpp \
00003 /Users/andi/Documents/ruhl_gits/hewes/src/ICSetters.h \
00004 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cmath \
00005 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00006 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config_site \
00007 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00008 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/math.h \
00009 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/math.h \
00010 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/cdefs.h \
00011 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00012 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00013 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/Availability.h \
00014 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityVersions.h \
00015 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityInternal.h \
00016 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdlib.h \
00017 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h \
00018 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types.h \
00019 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00020 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00021 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00022 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00023 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/wait.h \
00024 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_pid_t.h \
00025 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_id_t.h \
00026 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/signal.h \
00027 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/appleciops.h \
00028 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/signal.h \
00029 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/signal.h \
00030 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_mcontext.h \
00031 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_mcontext.h \
00032 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/machine/_structs.h \
00033 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/arm/_structs.h \
00034 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00035 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00036 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
00037 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \
00038 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00039 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00040 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00041 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00042 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00043 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00044 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \
00045 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00046 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_attr_t.h \
00047 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_sigaltstack.h \
00048 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_ucontext.h

```

```

00049 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigset_t.h \
00050 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_size_t.h \
00051 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uid_t.h \
00052 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/resource.h \
00053 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00054 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00055 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00056 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00057 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00058 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00059 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00060 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00061 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h \
00062 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timeval.h \
00063 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/endian.h \
00064 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/endian.h \
00065 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_endian.h \
00066 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/_OSByteOrder.h \
00067 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/arm/OSByteOrder.h \
00068 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/arch.h \
00069 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/alloca.h \
00070 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ct_rune_t.h \
00071 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rune_t.h \
00072 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wchar_t.h \
00073 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_null.h \
00074 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/malloc/_malloc.h \
00075 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_dev_t.h \
00076 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mode_t.h \
00077 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/type_traits \
00078 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00079 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/version \
00080 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stddef.h \
00081 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/_stddef_max_align_t.h \
00082 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_nullptr \
00083 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/forward.h \
00084 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/move.h \
00085 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_undef_macros \
00086 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits \
00087 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/numbers \
00088 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/concepts \
00089 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/invoke.h \
00090
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/weak_result_type.h \
\
00091
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binary_function.h \
\
00092
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/unary_function.h \
\
00093
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional_base \
00094
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/operations.h \
\
00095
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/reference_wrapper.h \
\
00096
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/addressof.h \
00097
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/allocator_arg_t.h \
\
00098
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/uses_allocator.h \
\
00099
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/exception \
00100
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_availability \
00101
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdlib \
00102
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/new \
00103
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/typeinfo \
00104
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00105
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility \
00106
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_debug \
00107
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iosfwd \
00108
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mbstate_t.h \
00109
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wchar.h \
00110
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00111
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wchar.h \
00112
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mbstate_t.h \
00113
/ Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdarg.h \
00114
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdio.h \
00115
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdio.h \
00116
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_stdio.h \
00117
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_va_list.h \
00118
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_stdio.h \
00119
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctermid.h \
00120
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_off_t.h \
00121
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ssize_t.h \
00122
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/time.h \
00123
/ Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_clock_t.h \

```

```
00124 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_time_t.h \
00125 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timespec.h \
00126 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00127 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00128 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wint_t.h \
00129 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctype_t.h \
00130 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctype.h \
00131 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/ctype.h \
00132 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctype.h \
00133 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/runetype.h \
00134 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_tuple \
00135 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/as_const.h \
00136 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/cmp.h \
00137 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/declval.h \
00138 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/exchange.h \
00139 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/in_place.h \
00140 \
00141 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/integer_sequence.h \
00142 \
00143 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/pair.h \
00144 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/unwrap_ref.h \
00145 \
00146 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/piecewise_construct.h \
00147 \
00148 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/rel_ops.h \
00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/swap.h \
00150 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/to_underlying.h \
00151 \
00152 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bits \
00153 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_cstring \
00154 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_string.h \
00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/string.h \
00156 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/functional \
00157 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/search.h \
00158 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/comp.h \
00159 \
00160 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/iterator_traits.h \
00161 \
00162 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/incrementable_traits.h \
00163 \
00164 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/readable_traits.h \
00165 \
00166 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binary_negate.h \
00167 \
00168 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/bind_front.h \
00169 \
00170 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/perfect_forward.h \
00171 \
00172 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \
00173 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/bind.h \
00174 \
00175 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binder1st.h \
00176 \
00177 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binder2nd.h \
00178 \
00179 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/default_searcher.h \
00180 \
00181 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/function.h \
00182 \
00183 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/allocator_traits.h \
00184 \
00185 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/construct_at.h \
00186 \
00187 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/pointer_traits.h \
00188 \
00189 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/compressed_pair.h \
00190 \
00191 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/shared_ptr.h \
00192 \
00193 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/allocation_guard.h \
00194 \
00195 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/allocator.h \
00196 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdexcept \
00197 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/unique_ptr.h \
00198 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/hash.h \
00199 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/atomic \
00200
```

```

/Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__thread/poll_with_backoff.h
00183 \
00184 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/chrono \
00185 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctime \
00186 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ratio \
00187 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/climits \
00188 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/limits.h \
00189 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/limits.h \
00190 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/limits.h \
00191 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/limits.h \
00192 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_limits.h \
00193 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/syslimits.h \
00194 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__threading_support \
00195 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/errno.h \
00196 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/errno.h \
00197 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/errno.h \
00198 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/sched.h \
00199 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/pthread_impl.h \
00200 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_cond_t.h \
00201 \
00202 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_condattr_t.h \
00203 \
00204 \
00205 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_key_t.h \
00206 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutex_t.h \
00207 \
00208 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutexattr_t.h \
00209 \
00210 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/qos.h \
00211 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/qos.h \
00212 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sched.h \
00213 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory \
00214 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_safety.h \
00215 \
00216 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/raw_storage_iterator.h \
00217 \
00218 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator \
00219 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator/access.h \
00220 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator/advance.h \
00221 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator/concepts.h \
00222 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator/move.h \
00223 \
00224 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/back_insert_iterator.h \
00225 \
00226 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator/iterator.h \
00227 \
00228 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/common_iterator.h \
00229 \
00230 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_swap.h \
00231 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/access.h \
00232 \
00233 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_borrowed_range.h \
00234 \
00235 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/__decay_copy.h \
00236 \
00237 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/variant \
00238 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__variant/monostate.h \
00239 \
00240 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/counted_iterator.h \
00241 \
00242 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/default_sentinel.h \
00243 \
00244 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/data.h \
00245 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/distance.h \
00246 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/empty.h \
00247 \
00248 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/erase_if_container.h \
00249 \
00250 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/front_insert_iterator.h \
00251 \
00252 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/insert_iterator.h \
00253 \

```

```
00240 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istreambuf_iterator.h \
00241 \
00242 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istream_iterator.h \
00243 \
00244 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/move_iterator.h \
00245 \
00246 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/next.h \
00247 \
00248 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostreambuf_iterator.h \
00249 \
00250 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/prev.h \
00251 \
00252 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/projected.h \
00253 \
00254 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_access.h \
00255 \
00256 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_iterator.h \
00257 \
00258 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/temporary_buffer.h \
00259 \
00260 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/uninitialized_algorithms.h \
00261 \
00262 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cassert \
00263 \
00264 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/assert.h \
00265 \
00266 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/identity.h \
00267 \
00268 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fn.h \
00269 \
00270 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fun_ref.h \
00271 \
00272 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/not_fn.h \
00273 \
00274 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_binary_function.h \
00275 \
00276 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_unary_function.h \
00277 \
00278 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/ranges_operations.h \
00279 \
00280 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_negate.h \
00281 \
00282 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/adjacent_find.h \
00283 \
00284 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/all_of.h \
00285 \
00286 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/any_of.h \
00287 \
00288 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/binary_search.h \
00289 \
00290 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lower_bound.h \
00291 \
00292 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/half_positive.h \
00293 \
00294 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp_ref_type.h \
00295 \
00296 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/clamp.h \
00297 \
00298 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy.h \
00299 \
00300 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unwrap_iter.h \
00301 \
00302 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_backward.h \
00303 \
00304 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_if.h \
00305 \
00306 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_n.h \
00307 \
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count.h \
00309 \
00310 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count_if.h \
00311 \
00312 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal.h \
00313 \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal_range.h \
00315 \
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/upper_bound.h \
00317 \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill_n.h \
00319 \
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill.h \
00321 \
00322 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00323 \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_end.h \
00325 
```

```

00285 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_first_of.h
00286 \
00287 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if.h \
00288 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if_not.h \
00289 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each.h \
00290 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each_n.h \
00291 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate.h \
00292 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/includes.h \
00293 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/inplace_merge.h \
00294 \
00295 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min.h \
00296 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min_element.h \
00297 \
00298 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00299 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate.h \
00300 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move_backward.h \
00301 \
00302 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/swap_ranges.h \
00303 \
00304 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap.h \
00305 \
00306 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap_until.h \
00307 \
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_partitioned.h \
00309 \
00310 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_permutation.h \
00311 \
00312 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted.h \
00313 \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted_until.h \
00315 \
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/iter_swap.h \
00317 \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lexicographical_compare.h \
00319 \
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max.h \
00321 \
00322 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max_element.h \
00323 \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00325 \
00326 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00327 \
00328 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/mismatch.h \
00329 \
00330 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00331 \
00332 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/nth_element.h \
00333 \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/none_of.h \
00335 \
00336 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h \
00337 \
00338 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_element.h \
00339 \
00340 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h \
00341 \
00342 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h \
00343 \
00344 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h \
00345 \
00346 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h \
00347 \
00348 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy_if.h

```

```

00333   \
00334   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_if.h \
00335   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace.h \
00336   \
00337   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy.h \
00338   \
00339   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_if.h \
00340   \
00341   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse_copy.h \
00342   \
00343   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate_copy.h \
00344   \
00345   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sample.h \
00346   \
00347   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__random/uniform_int_distribution.h \
00348   \
00349   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search_n.h \
00350   \
00351   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_difference.h \
00352   \
00353   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_intersection.h \
00354   \
00355   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_symmetric_difference.h \
00356   \
00357   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_union.h \
00358   \
00359   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_left.h \
00360   \
00361   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_right.h \
00362   \
00363   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shuffle.h \
00364   \
00365   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_partition.h \
00366   \
00367   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_sort.h \
00368   \
00369   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/transform.h \
00370   \
00371   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unique_copy.h \
00372   \
00373   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unique.h \
00374   \
00375   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/vector \
00376   \
00377   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bit_reference \
00378   \
00379   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__split_buffer \
00380   \
00381   /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00382   \
00383   /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00384   \
00385   /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00386   \
00387   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00388   \
00389   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00390   \
00391   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h

```

6.20 src/build/CMakeFiles/hewes.dir/LatticePatch.cpp.o.d File Reference

6.21 LatticePatch.cpp.o.d

[Go to the documentation of this file.](#)

```

00001 CMakeFiles/hewes.dir/LatticePatch.cpp.o: \
00002   /Users/andi/Documents/ruhl_gits/hewes/src/LatticePatch.cpp \
00003   /Users/andi/Documents/ruhl_gits/hewes/src/LatticePatch.h \
00004   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iomanip \
00005   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00006   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config_site \
00007   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00008   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__string \
00009   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy.h \
00010   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unwrap_iter.h \
00011   \
00012   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iterator \
00013   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__debug \
00014   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iosfwd \
00015   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__mbstate_t.h \
00016   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wchar.h \
00017   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00018   /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stddef.h \

```

```

00018 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/__stddef_max_align_t.h \
00019 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_nullptr \
00020 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wchar.h \
00021 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types.h \
00022 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00023 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/cdefs.h \
00024 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00025 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00026 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00027 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00028 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00029 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/Availability.h \
00030 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityVersions.h \
00031 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityInternal.h \
00032 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_null.h \
00033 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_size_t.h \
00034 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mbstate_t.h \
00035 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00036 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00037 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
00038 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \
00039 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00040 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00041 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00042 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00043 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00044 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00045 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \
00046 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00047 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ct_rune_t.h \
00048 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rune_t.h \
00049 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wchar_t.h \
00050 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdarg.h \
00051 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdio.h \
00052 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdio.h \
00053 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_stdio.h \
00054 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_va_list.h \
00055 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/stdio.h \
00056 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctermid.h \
00057 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_off_t.h \
00058 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ssize_t.h \
00059 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/time.h \
00060 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_clock_t.h \
00061 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_time_t.h \
00062 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timespec.h \
00063 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00064 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00065 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wint_t.h \
00066 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctype_t.h \
00067 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctype.h \
00068 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/ctype.h \
00069 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctype.h \
00070 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/runetype.h \
00071 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional_base \
00072 \
00073 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binary_function.h \
00074 \
00075 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/invoke.h \
00076 \
00077 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/weak_result_type.h \
00078 \
00079 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/unary_function.h \
00080 \
00081 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/_undef_macros \
00082 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/type_traits \
00083 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00084 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/version \
00085 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility/forward.h \
00086 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility/move.h \
00087 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/operations.h \
00088 \
00089 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/reference_wrapper.h \
00090 \
00091 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/addressof.h \
00092 \
00093 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/allocator_arg_t.h \
00094 \
00095 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/uses_allocator.h \
00096 \
00097 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_exception \
00098 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_availability \
00099 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdlib \
00100 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdlib.h \
00101 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h \
00102 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/wait.h \

```

```
00093 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_pid_t.h \
00094 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_id_t.h \
00095 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_signal.h \
00096 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/appleapiopts.h \
00097 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_signal.h \
00098 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_signal.h \
00099 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_mcontext.h \
00100 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_mcontext.h \
00101 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/machine/_structs.h \
00102 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/arm/_structs.h \
00103 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_attr_t.h \
00104 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigaltstack.h \
00105 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ucontext.h \
00106 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigset_t.h \
00107 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uid_t.h \
00108 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/resource.h \
00109 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00110 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00111 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00112 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00113 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00114 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00115 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00116 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00117 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h \
00118 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timeval.h \
00119 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/endian.h \
00120 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_endian.h \
00121 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_endian.h \
00122 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/_OSByteOrder.h \
00123 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/arm/OSByteOrder.h \
00124 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/arch.h \
00125 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/alloc.h \
00126 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/malloc/_malloc.h \
00127 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_dev_t.h \
00128 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mode_t.h \
00129 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_new \
00130 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/typeinfo \
00131 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00132 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility \
00133 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_tuple \
00134 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/as_const.h \
00135 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/cmp.h \
00136 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_limits \
00137 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/declval.h \
00138 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/exchange.h \
00139 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/in_place.h \
00140 \
00141 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/integer_sequence.h \
00142 \
00143 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/piecewise_construct.h \
00144 \
00145 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/rel_ops.h \
00146 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/swap.h \
00147 \
00148 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_compare \
00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_initializer_list \
00150 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/access.h \
00151 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/advance.h \
00152 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_function_like.h \
00153 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/concepts.h \
00154 \
00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/incrementable_traits.h \
00156 \
00157 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/readable_traits.h \
00158 \
00159 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_memory/pointer_traits.h \
00160 \
00161 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/back_insert_iterator.h \
00162 \
00163 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/iterator.h \
00164 \
00165 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/common_iterator.h \
00166 \
00167 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_iterator/iter_swap.h \
```

```

00163 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/access.h \
00164 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_borrowed_range.h \
\
00165 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/__decay_copy.h \
\
00166 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/variant \
00167 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/hash.h \
00168 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstring \
00169 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string.h \
00170 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/string.h \
00171 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/strings.h \
00172 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__variant/monostate.h \
00173 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \
00174
00175 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/counted_iterator.h \
\
00176 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/default_sentinel.h \
\
00177 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/data.h \
00178 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/distance.h \
00179 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/empty.h \
00180
00181 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/erase_if_container.h \
\
00182 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/front_insert_iterator.h \
\
00183 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/insert_iterator.h \
\
00184 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istreambuf_iterator.h \
\
00185 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/next.h \
00186
00187 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostreambuf_iterator.h \
\
00188 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostream_iterator.h \
00189 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/projected.h \
00190
00191 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_access.h \
\
00192 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_iterator.h \
\
00193 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/size.h \
00194
00195 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_backward.h \
\
00196 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_n.h \
00197 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill_n.h \
00198
00199 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_first_of.h \
\
00200 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp.h \
00201 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_end.h \
00202 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min.h \
00203 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min_element.h \
\
00204 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdio \
00205 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwchar \
00206 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwctype \
00207 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wctype.h \
00208 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types_wctrans_t.h \
00209 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/istream \
00210 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ostream \
00211 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/bitset \
00212 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bit_reference \
00213 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bits \
00214 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/algorithm \
00215 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/functional \
00216 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search.h \
00217

```

```
00218 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_negate.h
00219 \
00220 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/bind_front.h
00221 \
00222 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/perfect_forward.h
00223 \
00224 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/bind.h \
00225 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binder1st.h
00226 \
00227 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binder2nd.h
00228 \
00229 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/default_searcher.h
00230 \
00231 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/function.h \
00232 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_traits.h
00233 \
00234 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/construct_at.h \
00235 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/compressed_pair.h
00236 \
00237 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/shared_ptr.h \
00238 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocation_guard.h
00239 \
00240 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator.h \
00241 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdexcept \
00242 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/unique_ptr.h \
00243 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/atomic \
00244 \
00245 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/thread/poll_with_backoff.h
00246 \
00247 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/chrono \
00248 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctime \
00249 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ratio \
00250 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/climits \
00251 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits.h \
00252 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/include/limits.h \
00253 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/include/machine/limits.h \
00254 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/limits.h \
00255 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_limits.h \
00256 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/syslimits.h \
00257 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__threading_support \
00258 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/errno.h \
00259 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/errno.h \
00260 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/sched.h \
00261 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/pthread_impl.h \
00262 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_cond_t.h \
00263 \
00264 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_condattr_t.h
00265 \
00266 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_key_t.h \
00267 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutex_t.h \
00268 \
00269 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutexattr_t.h
00270 \
00271 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory/raw_storage_iterator.h
00272 \
00273 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory/temporary_buffer.h
00274 \
00275 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory/uninitialized_algorithms.h
00276 \
00277 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cassert \
00278 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/assert.h \
00279 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/identity.h \
```

```
00273 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fn.h \
00274 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fun_ref.h \
\
00275 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/not_fn.h \
00276 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_binary_function.h \
\
00277 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_unary_function.h \
\
00278 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/ranges_operations.h \
\
00279 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_negate.h \
\
00280 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/adjacent_find.h \
\
00281 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/all_of.h \
00282 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/any_of.h \
00283 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/binary_search.h \
\
00284 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lower_bound.h \
\
00285 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/half_positive.h \
\
00286 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp_ref_type.h \
\
00287 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/clamp.h \
00288 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_if.h \
00289 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count.h \
00290 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count_if.h \
00291 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal.h \
00292 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal_range.h \
\
00293 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/upper_bound.h \
\
00294 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill.h \
00295 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00296 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if.h \
00297 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if_not.h \
\
00298 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each.h \
00299 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each_n.h \
\
00300 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate_n.h \
\
00301 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate.h \
00302 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/includes.h \
00303 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/inplace_merge.h \
\
00304 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00305 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate.h \
00306 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move_backward.h \
\
00307 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/swap_ranges.h \
\
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap.h \
00309 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap_until.h \
\
00310 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_partitioned.h \
\
00311 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_permutation.h \
\
00312 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted.h \
00313 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted_until.h \
\
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/iter_swap.h \
00315 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lexicographical_compare.h \
\
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00317 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sift_down.h \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max.h \
00319 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max_element.h \
\
```

```
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/merge.h \
00321 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax.h \
00322 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax_element.h \
\
00323 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/mismatch.h \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/next_permutation.h \
\
00325 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse.h \
00326 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/none_of.h \
00327 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/nth_element.h \
\
00328 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort.h \
00329 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort.h \
\
00330 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort_heap.h \
00331 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/pop_heap.h \
00332 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort_copy.h \
\
00333 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h \
\
00335 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h \
\
00336 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/prev_permutation.h \
\
00337 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/push_heap.h \
00338 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove.h \
00339 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy.h \
\
00340 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy_if.h \
\
00341 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_if.h \
00342 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace.h \
00343 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy.h \
\
00344 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy_if.h \
\
00345 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_if.h \
\
00346 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse_copy.h \
\
00347 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate_copy.h \
\
00348 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sample.h \
00349 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__random/uniform_int_distribution.h \
\
00350 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search_n.h \
00351 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_difference.h \
\
00352 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_intersection.h \
\
00353 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_symmetric_difference.h \
\
00354 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_union.h \
00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_left.h \
\
00356 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_right.h \
\
00357 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shuffle.h \
00358 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_partition.h \
\
00359 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_sort.h \
\
00360 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/transform.h \
00361 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unique_copy.h \
\
00362 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unique.h \
00363 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string \
00364 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string_view \
00365 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_view.h \
00366 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ios \
```

```

00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__locale \
00368 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mutex \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mutex_base \
00370 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/system_error \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_errc \
00372 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cerrno \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale.h \
00374 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/locale.h \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_locale.h \
00376 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale.h \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_xlocale.h \
00378 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_ctype.h \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00380 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdio.h \
00381 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdlib.h \
00382 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_string.h \
00383 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_time.h \
00384 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wchar.h \
00385 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00386 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale \
00387 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/streambuf \
00388 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/nl_types.h \
00389 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/types.h \
00390 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_char.h \
00391 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_short.h \
00392 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int.h \
00393 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_caddr_t.h \
00394 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blkcnt_t.h \
00395 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blksize_t.h \
00396 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_gid_t.h \
00397 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_addr_t.h \
00398 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_port_t.h \
00399 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino_t.h \
00400 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino64_t.h \
00401 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_key_t.h \
00402 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_nlink_t.h \
00403 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_useconds_t.h \
00404 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_suseconds_t.h \
00405 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rsize_t.h \
00406 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_errno_t.h \
00407 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_def.h \
00408 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_setsize.h \
00409 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_set.h \
00410 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_clr.h \
00411 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_zero.h \
00412 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_isset.h \
00413 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_copy.h \
00414 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsblkcnt_t.h \
00415 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsfilcnt_t.h \
00416 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_nl_item.h \
00417 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bsd_locale_defaults.h \
00418 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iostream \
00419 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/sstream \
00420 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/array \
00421 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/vector \
00422 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__split_buffer \
00423 /opt/homebrew/include/mpi.h \
00424 /opt/homebrew/include/mpi_portable_platform.h \
00425 /opt/homebrew/include/omp.h \
00426 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode.h \
00427 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_context.h \
00428 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_logger.h \
00429 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00430 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00431 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00432 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00433 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00434 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00435 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_profiler.h \
00436 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nvector.h \
00437 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nonlinear solver.h \
00438 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_ls.h \
00439 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_direct.h \
00440 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_iterative.h \
00441 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_linear solver.h \
00442 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_matrix.h \
00443 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_proj.h \
00444 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_openmp.h \
00445 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpiplus.h \
00446 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpimanyvector.h \
00447 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_mpi_types.h \
00448 /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.h \
00449 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/math.h \
00450 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/math.h

```

6.22 src/build/CMakeFiles/hewes.dir/main.cpp.o.d File Reference

6.23 main.cpp.o.d

[Go to the documentation of this file.](#)

```
00001 CMakeFiles/hewes.dir/main.cpp.o: \
00002   /Users/andi/Documents/ruhl_gits/hewes/src/main.cpp \
00003   /Users/andi/Documents/ruhl_gits/hewes/src/SimulationFunctions.h \
00004   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cmath \
00005   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00006   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config_site \
00007   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00008   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/math.h \
00009   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/math.h \
00010   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/cdefs.h \
00011   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00012   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00013   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/Availability.h \
00014   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityVersions.h \
00015   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityInternal.h \
00016   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdlib.h \
00017   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h \
00018   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types.h \
00019   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00020   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00021   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00022   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00023   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/wait.h \
00024   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_pid_t.h \
00025   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_id_t.h \
00026   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/signal.h \
00027   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/applepiopts.h \
00028   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/signal.h \
00029   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/signal.h \
00030   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_mcontext.h \
00031   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_mcontext.h \
00032   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/machine/_structs.h \
00033   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/arm/_structs.h \
00034   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00035   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00036   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
00037   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \
00038   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00039   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00040   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00041   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00042   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00043   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00044   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \
00045   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00046   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_attr.h \
00047   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigaltstack.h \
00048   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ucontext.h \
00049   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigset.h \
00050   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_size_t.h \
00051   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uid_t.h \
00052   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/resource.h \
00053   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00054   /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00055   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00056   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00057   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00058   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00059   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00060   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00061   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h \
00062   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timeval.h \
00063   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/endian.h \
00064   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/endian.h \
00065   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_endian.h \
00066   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/_OSByteOrder.h \
00067   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/arm/OSByteOrder.h \
00068   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/arch.h \
00069   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/alloca.h \
00070   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ct_rune_t.h \
00071   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rune_t.h \
00072   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wchar_t.h \
00073   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_null.h \
00074   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/malloc/_malloc.h \
00075   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_dev_t.h \
00076   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mode_t.h \
00077   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/type_traits \
```

```

00078 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstddef \
00079 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/version \
00080 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stddef.h \
00081 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/_stddef_max_align_t.h \
00082 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__nullptr \
00083 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/forward.h \
00084 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/move.h \
00085 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__undef_macros \
00086 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits \
00087 /Users/andi/Documents/ruhl_gits/hewes/src/LatticePatch.h \
00088 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iomanip \
00089 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string \
00090 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/algorithm/copy.h \
00091 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/algorithm/unwrap_iter.h \
\\
00092 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iterator \
00093 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__debug \
00094 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iosfwd \
00095 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mbstate_t.h \
00096 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wchar.h \
00097 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00098 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wchar.h \
00099 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mbstate_t.h \
00100 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdarg.h \
00101 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdio.h \
00102 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdio.h \
00103 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_stdio.h \
00104 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_va_list.h \
00105 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/stdio.h \
00106 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctermid.h \
00107 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_off_t.h \
00108 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ssize_t.h \
00109 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/time.h \
00110 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_clock_t.h \
00111 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_time_t.h \
00112 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timespec.h \
00113 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00114 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/__wctype.h \
00115 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wint_t.h \
00116 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctype_t.h \
00117 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctype.h \
00118 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/ctype.h \
00119 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctype.h \
00120 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/runetype.h \
00121 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional_base \
00122 \
\\
00123 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_function.h \
\\
00124 \
\\
00125 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/invoke.h \
\\
00126 \
\\
00127 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/weak_result_type.h \
\\
00128 \
\\
00129 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_function.h \
\\
00130 \
\\
00131 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/reference_wrapper.h \
\\
00132 \
\\
00133 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/addressof.h \
\\
00134 \
\\
00135 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_arg_t.h \
\\
00136 \
\\
00137 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/uses_allocator.h \
\\
00138 \
\\
00139 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/exception \
00140 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/cmp.h \
00141 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/decval.h \
00142 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/exchange.h \
00143 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/in_place.h \
00144 \
\\
00145 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/integer_sequence.h \
\\
00146 \
\\
00147 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/pair.h \
\\
00148 \
\\
00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unwrap_ref.h \
\\
\\
00150 \
\\
00151 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/piecewise_construct.h

```

```
\00148 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/relops.h \
00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/swap.h \
00150 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/to_underlying.h \
\00151 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/compare \
00152 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/initializer_list \
00153 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/access.h \
00154 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/advance.h \
00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__function_like.h \
00156 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/concepts.h \
00157 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/incrementable_traits.h \
\00158 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/concepts \
00159 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_move.h \
00160 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator_traits.h \
\00161 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/readable_traits.h \
\00162 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_traits.h \
\00163 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/back_insert_iterator.h \
\00164 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator.h \
00165 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/common_iterator.h \
\00166 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_swap.h \
00167 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/access.h \
00168 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_borrowed_range.h \
\00169 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/__decay_copy.h \
\00170 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/variant \
00171 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/hash.h \
00172 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstring \
00173 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string.h \
00174 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/string.h \
00175 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/strings.h \
00176 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__variant/monostate.h \
00177 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \
00178 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/counted_iterator.h \
\00179 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/default_sentinel.h \
\00180 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/data.h \
00181 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/distance.h \
00182 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/empty.h \
00183 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/erase_if_container.h \
\00184 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/front_insert_iterator.h \
\00185 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/insert_iterator.h \
\00186 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istreambuf_iterator.h \
\00187 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istream_iterator.h \
\00188 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/move_iterator.h \
\00189 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/next.h \
00190 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostreambuf_iterator.h \
\00191 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostream_iterator.h \
\00192 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/prev.h \
00193 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/projected.h \
00194 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_access.h \
\00195 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_iterator.h
```

```

00196 \
00197 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/size.h \
00198 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/wrap_iter.h \
00199 \
00200 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_backward.h \
00201 \
00202 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_n.h \
00203 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_n.h \
00204 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min.h \
00205 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min_element.h \
00206 \
00207 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdio \
00208 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwchar \
00209 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwctype \
00210 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cctype \
00211 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wctype.h \
00212 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctrans_t.h \
00213 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/istream \
00214 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ostream \
00215 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/bitset \
00216 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bit_reference \
00217 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bits \
00218 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/algorithm \
00219 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/functional \
00220 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search.h \
00221 \
00222 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_negate.h \
00223 \
00224 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/bind_front.h \
00225 \
00226 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binder1st.h \
00227 \
00228 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/default_searcher.h \
00229 \
00230 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_traits.h \
00231 \
00232 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/construct_at.h \
00233 \
00234 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/compressed_pair.h \
00235 \
00236 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/shared_ptr.h \
00237 \
00238 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocation_guard.h \
00239 \
00240 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/chrono \
00241 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctime \
00242 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ratio \
00243 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits.h \
00244 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/limits.h \
00245 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/limits.h \
00246 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/limits.h \
00247 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/limits.h \
00248 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_limits.h \
00249 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/syslimits.h \
00250 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__threading_support \
00251 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/errno.h \
00252 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/errno.h \
00253 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/errno.h \
00254 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/sched.h \
00255 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/pthread_impl.h \
00256 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_cond_t.h \
00257 \
00258 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_condattr_t.h \
00259 \

```

```
00258 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_key_t.h \
00259 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutex_t.h \
00260 \
00261 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutexattr_t.h \
00262 \
00263 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_once_t.h \
00264 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_rwlock_t.h \
00265 \
00266 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/qos.h \
00267 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mach_port_t.h \
00268 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sched.h \
00269 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory \
00270 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_safety.h \
00271 \
00272 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/raw_storage_iterator.h \
00273 \
00274 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/temporary_buffer.h \
00275 \
00276 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/identity.h \
00277 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fn.h \
00278 \
00279 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fun_ref.h \
00280 \
00281 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/not_fn.h \
00282 \
00283 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_binary_function.h \
00284 \
00285 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_unary_function.h \
00286 \
00287 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/ranges_operations.h \
00288 \
00289 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_negate.h \
00290 \
00291 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/adjacent_find.h \
00292 \
00293 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/all_of.h \
00294 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/any_of.h \
00295 \
00296 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/binary_search.h \
00297 \
00298 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lower_bound.h \
00299 \
00300 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/half_positive.h \
00301 \
00302 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp_ref_type.h \
00303 \
00304 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/clamp.h \
00305 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_if.h \
00306 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count.h \
00307 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count_if.h \
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal.h \
00309 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal_range.h \
00310 \
00311 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/upper_bound.h \
00312 \
00313 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill.h \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00315 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if.h \
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if_not.h \
00317 \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each.h \
00319 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each_n.h \
00320 \
00321 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate_n.h \
00322 \
00323 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate.h \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/includes.h \
00325
```

```
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/inplace_merge.h
00309 \
00310 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00311 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate.h \
00312 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move_backward.h
00313 \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/swap_ranges.h
00315 \
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap.h \
00317 \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap_until.h
00319 \
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_permutation.h
00321 \
00322 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted.h \
00323 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted_until.h
00324 \
00325 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/iter_swap.h \
00326 \
00327 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lexicographical_compare.h
00328 \
00329 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00330 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sift_down.h \
00331 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max.h \
00332 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max_element.h
00333 \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/merge.h \
00335 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax.h \
00336 \
00337 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax_element.h
00338 \
00339 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/mismatch.h \
00340 \
00341 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/next_permutation.h
00342 \
00343 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse.h \
00344 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/none_of.h \
00345 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/nth_element.h
00346 \
00347 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sift_up.h \
00348 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort.h \
00349 \
00350 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h \
00351 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h
00352 \
00353 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h
00354 \
00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/prev_permutation.h
00356 \
00357 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort.h \
00358 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort_copy.h
00359 \
00360 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h
00361 \
00362 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h
00363 \
00364 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point_if.h
00365 \
00366 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_if.h \
00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace.h \
00368 \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy.h
00370 \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy_if.h
00372 \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_if.h
00374 \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse_copy.h
00376 \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate_copy.h
00378 \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sample.h \
```

```
00353 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_random/uniform_int_distribution.h
00354 \
00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/search_n.h \
00356 \
00357 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/set_difference.h
00358 \
00359 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/set_symmetric_difference.h
00360 \
00361 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/shuffle.h \
00362 \
00363 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/stable_partition.h
00364 \
00365 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/stable_sort.h
00366 \
00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string \
00368 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string_view \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_ranges/enable_view.h \
00370 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ios \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_locale \
00372 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/mutex \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mutex_base \
00374 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/system_error \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_errc \
00376 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cerrno \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale.h \
00378 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/locale.h \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_locale.h \
00380 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale.h \
00381 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_xlocale.h \
00382 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_ctype.h \
00383 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00384 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdio.h \
00385 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdlib.h \
00386 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_string.h \
00387 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_time.h \
00388 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wchar.h \
00389 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00390 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale \
00391 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/streambuf \
00392 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/nl_types.h \
00393 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/types.h \
00394 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_char.h \
00395 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_short.h \
00396 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int.h \
00397 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_caddr_t.h \
00398 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blkcnt_t.h \
00399 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blksize_t.h \
00400 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_gid_t.h \
00401 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_addr_t.h \
00402 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_port_t.h \
00403 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino_t.h \
00404 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino64_t.h \
00405 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_key_t.h \
00406 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mlink_t.h \
00407 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_useconds_t.h \
00408 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_suseconds_t.h \
00409 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rsize_t.h \
00410 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_errno_t.h \
00411 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_def.h \
00412 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_setsize.h \
00413 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_set.h \
00414 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_clr.h \
00415 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_zero.h \
00416 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_isset.h \
00417 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_copy.h \
00418 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsblkcnt_t.h \
00419 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsfilcnt_t.h \
00420 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_nl_item.h \
00421 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bsd_locale_defaults.h \
00422 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iostream \
00423 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/sstream \
00424 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/array \
00425 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/vector \
```

```

00426 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__split_buffer \
00427 /opt/homebrew/include/mpi.h \
00428 /opt/homebrew/include/mpi_portable_platform.h \
00429 /opt/homebrew/include/omp.h \
00430 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode.h \
00431 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_context.h \
00432 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_logger.h \
00433 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00434 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00435 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00436 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00437 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/float.h \
00438 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00439 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_profiler.h \
00440 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nvector.h \
00441 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nonsquaresolver.h \
00442 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_ls.h \
00443 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_direct.h \
00444 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_iterative.h \
00445 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_linearsolver.h \
00446 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_matrix.h \
00447 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_proj.h \
00448 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_openmp.h \
00449 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpiplusx.h \
00450 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpimanyvector.h \
00451 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_mpi_types.h \
00452 /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.h \
00453 /Users/andi/Documents/ruhl_gits/hewes/src/SimulationClass.h \
00454 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sunnonlinsol/sunnonlinsol_fixedpoint.h \
00455 /Users/andi/Documents/ruhl_gits/hewes/src/ICSetters.h \
00456 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/numbers \
00457 /Users/andi/Documents/ruhl_gits/hewes/src/Outputters.h \
00458 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/filesystem \
00459 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stack \
00460 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/deque \
00461 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/fstream \
00462 /Users/andi/Documents/ruhl_gits/hewes/src/TimeEvolutionFunctions.h

```

6.24 src/build/CMakeFiles/hewes.dir/Outputters.cpp.o.d File Reference

6.25 Outputters.cpp.o.d

[Go to the documentation of this file.](#)

```

00001 CMakeFiles/hewes.dir/Outputters.cpp.o: \
00002 /Users/andi/Documents/ruhl_gits/hewes/src/Outputters.cpp \
00003 /Users/andi/Documents/ruhl_gits/hewes/src/Outputters.h \
00004 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/filesystem \
00005 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__availability \
00006 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00007 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config_site \
00008 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00009 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__debug \
00010 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iosfwd \
00011 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__mbstate_t.h \
00012 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wchar.h \
00013 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00014 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stddef.h \
00015 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/__stddef_max_align_t.h \
00016 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__nullptr \
00017 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wchar.h \
00018 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types.h \
00019 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00020 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_cdefs.h \
00021 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00022 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00023 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00024 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00025 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00026 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/Availability.h \
00027 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityVersions.h \
00028 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityInternal.h \
00029 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_null.h \
00030 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_size_t.h \
00031 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mbstate_t.h \
00032 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00033 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00034 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
00035 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \

```

```

00036 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00037 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00038 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00039 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00040 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00041 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00042 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \
00043 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00044 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ct_rune_t.h \
00045 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rune_t.h \
00046 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wchar_t.h \
00047 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdarg.h \
00048 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdio.h \
00049 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdio.h \
00050 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_stdio.h \
00051 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_va_list.h \
00052 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/stdio.h \
00053 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctermid.h \
00054 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_off_t.h \
00055 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ssize_t.h \
00056 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/time.h \
00057 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_clock_t.h \
00058 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_time_t.h \
00059 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timespec.h \
00060 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00061 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00062 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wint_t.h \
00063 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctype_t.h \
00064 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctype.h \
00065 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/ctype.h \
00066 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctype.h \
00067 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/runetype.h \
00068 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/forward.h \
00069 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/type_traits \
00070 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstddef \
00071 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/version \
00072 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/move.h \
00073 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_undef_macros \
00074 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/chrono \
00075 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/compare \
00076 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctime \
00077 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits \
00078 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ratio \
00079 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/climits \
00080 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits.h \
00081 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/limits.h \
00082 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/limits.h \
00083 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/limits.h \
00084 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/limits.h \
00085 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_limits.h \
00086 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/syslimits.h \
00087 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00088 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00089 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00090 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00091 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00092 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00093 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00094 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00095 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00096 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h \
00097 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdlib \
00098 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdlib.h \
00099 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h \
00100 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/wait.h \
00101 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_pid_t.h \
00102 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_id_t.h \
00103 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/signal.h \
00104 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/appleapiopts.h \
00105 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/signal.h \
00106 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/signal.h \
00107 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_mcontext.h \
00108 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_mcontext.h \
00109 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/machine/_structs.h \
00110 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/arm/_structs.h \
00111 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_attr_t.h \
00112 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigaltstack.h \
00113 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ucontext.h \
00114 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigset_t.h \
00115 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uid_t.h \
00116 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/resource.h \
00117 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timeval.h \
00118 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/endian.h \
00119 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/endian.h \
00120 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_endian.h \
00121 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/_OSByteOrder.h \
00122 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/arm/OSByteOrder.h \

```

```

00123 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/arch.h \
00124 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/alloca.h \
00125 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/malloc_malloc.h \
00126 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_dev_t.h \
00127 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mode_t.h \
00128 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iterator \
00129 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional_base \
00130 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_function.h \
\
00131 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/invoke.h \
00132 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/weak_result_type.h \
\
00133 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_function.h \
\
00134 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/operations.h \
\
00135 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/reference_wrapper.h \
\
00136 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/addressof.h \
00137 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_arg_t.h \
\
00138 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/uses_allocator.h \
\
00139 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/exception \
00140 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/new \
00141 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/typeinfo \
00142 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility \
00143 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__tuple \
00144 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/as_const.h \
00145 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/cmp.h \
00146 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/declval.h \
00147 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/exchange.h \
00148 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/in_place.h \
00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/integer_sequence.h \
\
00150 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/pair.h \
00151 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unwrap_ref.h \
\
00152 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/piecewise_construct.h \
\
00153 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/rel_ops.h \
00154 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/swap.h \
00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/to_underlying.h \
\
00156 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/initializer_list \
00157 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/access.h \
00158 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/advance.h \
00159 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__function_like.h \
00160 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/concepts.h \
00161 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/incrementable_traits.h \
\
00162 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/concepts \
00163 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_move.h \
00164 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator_traits.h \
\
00165 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/readable_traits.h \
\
00166 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_traits.h \
\
00167 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/back_insert_iterator.h \
\
00168 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator.h \
00169 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/common_iterator.h \
\
00170 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_swap.h \
00171 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/access.h \
00172 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_borrowed_range.h \
\
00173 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/__decay_copy.h \
\
00174 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/variant \
00175 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/hash.h \
00176 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstring \
00177 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string.h \

```

```
00178 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/string.h \
00179 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/strings.h \
00180 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__variant/monostate.h \
00181 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \
00182 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/counted_iterator.h \
\\
00183 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/default_sentinel.h \
\\
00184 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/data.h \
00185 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/distance.h \
00186 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/empty.h \
00187 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/erase_if_container.h \
\\
00188 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/front_insert_iterator.h \
\\
00189 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/insert_iterator.h \
\\
00190 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istreambuf_iterator.h \
\\
00191 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istream_iterator.h \
\\
00192 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/move_iterator.h \
\\
00193 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/next.h \
00194 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostreambuf_iterator.h \
\\
00195 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostream_iterator.h \
\\
00196 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/prev.h \
00197 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/projected.h \
00198 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_access.h \
\\
00199 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_iterator.h \
\\
00200 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/size.h \
00201 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/wrap_iter.h \
00202 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory \
00203 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocation_guard.h \
\\
00204 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_traits.h \
\\
00205 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/construct_at.h \
00206 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator.h \
00207 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdexcept \
00208 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/compressed_pair.h \
\\
00209 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_safety.h \
\\
00210 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/raw_storage_iterator.h \
\\
00211 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/shared_ptr.h \
00212 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/unique_ptr.h \
00213 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/atomic \
00214 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__thread/poll_with_backoff.h \
\\
00215 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__threading_support \
00216 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/errno.h \
00217 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/errno.h \
00218 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/errno.h \
00219 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/sched.h \
00220 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/pthread_impl.h \
00221 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_cond_t.h \
00222 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_condattr_t.h \
\\
00223 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_key_t.h \
00224 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutex_t.h \
\\
00225 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutexattr_t.h
```

```

00226 \
00227 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_once_t.h \
00228 \
00229 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_rwlock_t.h \
00230 \
00231 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_rwlockattr_t.h \
00232 \
00233 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mach_port_t.h \
00234 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sched.h \
00235 \
00236 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/temporary_buffer.h \
00237 \
00238 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/uninitialized_algorithms.h \
00239 \
00240 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_cassert \
00241 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/assert.h \
00242 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_stack \
00243 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_deque \
00244 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_split_buffer \
00245 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm \
00246 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/search.h \
00247 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/comp.h \
00248 \
00249 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional \
00250 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/bind_front.h \
00251 \
00252 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/perfect_forward.h \
00253 \
00254 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/bind.h \
00255 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binder1st.h \
00256 \
00257 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binder2nd.h \
00258 \
00259 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/default_searcher.h \
00260 \
00261 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/function.h \
00262 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/identity.h \
00263 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/mem_fn.h \
00264 \
00265 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/mem_fun_ref.h \
00266 \
00267 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/not_fn.h \
00268 \
00269 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional(pointer_to_binary_function.h \
00270 \
00271 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional(pointer_to_unary_function.h \
00272 \
00273 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/ranges_operations.h \
00274 \
00275 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/unary_negate.h \
00276 \
00277 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/adjacent_find.h \
00278 \
00279 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/all_of.h \
00280 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/any_of.h \
00281 \
00282 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/binary_search.h \
00283 \
00284 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/lower_bound.h \
00285 \
00286 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/half_positive.h \
00287 \
00288 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/comp_ref_type.h \
00289 \
00290 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/clamp.h \
00291 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/copy.h \
00292 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/unwrap_iter.h \
00293 \
00294 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/copy_backward.h \
00295 \
00296 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/copy_if.h \
00297

```

```
00274 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_n.h \
00275 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count.h \
00276 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count_if.h \
00277 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal.h \
00278 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal_range.h \
00279 \
00280 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/upper_bound.h \
00281 \
00282 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill_n.h \
00283 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill.h \
00284 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00285 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_end.h \
00286 \
00287 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_first_of.h \
00288 \
00289 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if.h \
00290 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if_not.h \
00291 \
00292 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each.h \
00293 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each_n.h \
00294 \
00295 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate_n.h \
00296 \
00297 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate.h \
00298 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/includes.h \
00299 \
00300 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/inplace_merge.h \
00301 \
00302 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min.h \
00303 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min_element.h \
00304 \
00305 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00306 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate.h \
00307 \
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move_backward.h \
00309 \
00310 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/swap_ranges.h \
00311 \
00312 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap.h \
00313 \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap_until.h \
00315 \
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_partitioned.h \
00317 \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_permutation.h \
00319 \
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted.h \
00321 \
00322 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted_until.h \
00323 \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/iter_swap.h \
00325 \
00326 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lexicographical_compare.h \
00327 \
00328 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00329 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sift_down.h \
00330 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max.h \
00331 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max_element.h \
00332 \
00333 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/merge.h \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax.h \
00335 \
00336 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax_element.h \
00337 \
00338 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/mismatch.h \
00339 \
00340 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/next_permutation.h \
00341 \
00342 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse.h \
00343 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/none_of.h \
00344 \
00345 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/nth_element.h \
00346 \
00347 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort.h \
00348 \
00349 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort.h \
00350 \
00351 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort_heap.h \
00352 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/pop_heap.h \
00353 \
00354 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort_copy.h \
00355 \
00356 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h \
00357 \
00358 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h \
00359 \
00360
```

```

00326 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h
00327 \
00328 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/prev_permutation.h
00329 \
00330 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/push_heap.h \
00331 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy.h
00332 \
00333 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_if.h \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace.h \
00335 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy.h
00336 \
00337 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse_copy.h
00338 \
00339 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate_copy.h
00340 \
00341 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__random/uniform_int_distribution.h
00342 \
00343 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search_n.h \
00344 \
00345 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_difference.h
00346 \
00347 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_left.h
00348 \
00349 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_right.h
00350 \
00351 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_partition.h
00352 \
00353 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_sort.h
00354 \
00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/transform.h \
00356 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string \
00357 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdio \
00358 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string_view \
00359 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ranges/enable_view.h \
00360 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwchar \
00361 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwctype \
00362 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wctype \
00363 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wctype.h \
00364 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctrans_t.h \
00365 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/system_error \
00366 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_errc \
00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cerrno \
00368 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_locale \
00370 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mutex \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mutex_base \
00372 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale.h \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/locale.h \
00374 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_locale.h \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale.h \
00376 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_xlocale.h \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_ctype.h \
00378 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdio.h \
00380 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdlib.h \
00381 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_string.h \
00382 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_time.h \
00383 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wchar.h \

```

```

00384 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00385 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ios \
00386 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/streambuf \
00387 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/nl_types.h \
00388 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/types.h \
00389 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_char.h \
00390 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_short.h \
00391 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int.h \
00392 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_caddr_t.h \
00393 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blkcnt_t.h \
00394 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blksize_t.h \
00395 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_gid_t.h \
00396 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_addr_t.h \
00397 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_port_t.h \
00398 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino_t.h \
00399 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino64_t.h \
00400 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_key_t.h \
00401 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_nlink_t.h \
00402 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_useconds_t.h \
00403 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_suseconds_t.h \
00404 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rsize_t.h \
00405 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_errno_t.h \
00406 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_def.h \
00407 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_setsize.h \
00408 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_set.h \
00409 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_clr.h \
00410 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_zero.h \
00411 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_isset.h \
00412 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_copy.h \
00413 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsblkcnt_t.h \
00414 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsfilcnt_t.h \
00415 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_nl_item.h \
00416 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bsd_locale_defaults.h \
00417 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iomanip \
00418 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/istream \
00419 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ostream \
00420 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/bitset \
00421 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bit_reference \
00422 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/fstream \
00423 /Users/andi/Documents/ruhl_gits/hewes/src/LatticePatch.h \
00424 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iostream \
00425 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/sstream \
00426 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/array \
00427 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/vector \
00428 /opt/homebrew/include/mpi.h \
00429 /opt/homebrew/include/mpi_portable_platform.h \
00430 /opt/homebrew/include/omp.h \
00431 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode.h \
00432 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_context.h \
00433 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_logger.h \
00434 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00435 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00436 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00437 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00438 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/float.h \
00439 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00440 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_profiler.h \
00441 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nvector.h \
00442 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nonlinear solver.h \
00443 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_ls.h \
00444 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_direct.h \
00445 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_iterative.h \
00446 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_linear solver.h \
00447 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_matrix.h \
00448 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_proj.h \
00449 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_openmp.h \
00450 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpiplusx.h \
00451 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpimanyvector.h \
00452 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_mpi_types.h \
00453 /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.h

```

6.26 src/build/CMakeFiles/hewes.dir/SimulationClass.cpp.o.d File Reference

6.27 SimulationClass.cpp.o.d

[Go to the documentation of this file.](#)

```

00001 CMakeFiles/hewes.dir/SimulationClass.cpp.o: \
00002 /Users/andi/Documents/ruhl_gits/hewes/src/SimulationClass.cpp \
00003 /Users/andi/Documents/ruhl_gits/hewes/src/SimulationClass.h \
00004 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sunnonlinsol/sunnonlinsol_fixedpoint.h \
00005 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00006 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00007 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00008 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00009 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00010 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config_site \
00011 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00012 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/float.h \
00013 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00014 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00015 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stddef.h \
00016 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/_stddef_max_align_t.h \
00017 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/nullptr \
00018 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00019 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00020 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00021 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
00022 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \
00023 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00024 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00025 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00026 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00027 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00028 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00029 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00030 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/cdefs.h \
00031 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00032 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00033 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00034 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00035 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00036 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \
00037 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00038 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00039 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00040 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00041 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00042 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00043 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00044 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00045 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h \
00046 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nvvector.h \
00047 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdio.h \
00048 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdio.h \
00049 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_stdio.h \
00050 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/Availability.h \
00051 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityVersions.h \
00052 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityInternal.h \
00053 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types.h \
00054 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_va_list.h \
00055 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_size_t.h \
00056 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_null.h \
00057 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/stdio.h \
00058 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctermid.h \
00059 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_off_t.h \
00060 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ssize_t.h \
00061 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdlib.h \
00062 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h \
00063 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/wait.h \
00064 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_pid_t.h \
00065 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_id_t.h \
00066 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/signal.h \
00067 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/appleapiopts.h \
00068 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/signal.h \
00069 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/signal.h \
00070 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_mcontext.h \
00071 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_mcontext.h \
00072 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/machine/_structs.h \
00073 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/arm/_structs.h \
00074 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_attr_t.h \
00075 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigaltstack.h \
00076 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ucontext.h \
00077 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigset_t.h \
00078 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uid_t.h \
00079 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/resource.h \
00080 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timeval.h \
00081 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/endian.h \
00082 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/endian.h \
00083 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_endian.h \
00084 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/_OSByteOrder.h \
00085 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/arm/OSByteOrder.h \
00086 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/arch.h \
00087 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/alloca.h \

```

```
00088 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ct_rune_t.h \
00089 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rune_t.h \
00090 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wchar_t.h \
00091 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/malloc/_malloc.h \
00092 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_dev_t.h \
00093 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mode_t.h \
00094 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_context.h \
00095 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_logger.h \
00096 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_profiler.h \
00097 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory \
00098 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional_base \
00099 \
00100 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_function.h \
00101 \
00102 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/invoke.h \
00103 \
00104 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/type_traits \
00105 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstddef \
00106 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/version \
00107 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/forward.h \
00108 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/move.h \
00109 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/operations.h \
00110 \
00111 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/reference_wrapper.h \
00112 \
00113 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/addressof.h \
00114 \
00115 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__availability \
00116 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdlib \
00117 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/new \
00118 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/typeinfo \
00119 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00120 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility \
00121 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__debug \
00122 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iosfwd \
00123 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__mbstate_t.h \
00124 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wchar.h \
00125 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wchar.h \
00126 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mbstate_t.h \
00127 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdarg.h \
00128 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/time.h \
00129 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_clock_t.h \
00130 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_time_t.h \
00131 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timespec.h \
00132 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00133 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/__wctype.h \
00134 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wint_t.h \
00135 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctype_t.h \
00136 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctype.h \
00137 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/ctype.h \
00138 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctype.h \
00139 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/runetype.h \
00140 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__tuple \
00141 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/as_const.h \
00142 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/cmp.h \
00143 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits \
00144 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/decval.h \
00145 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/exchange.h \
00146 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/in_place.h \
00147 \
00148 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/integer_sequence.h \
00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/pair.h \
00150 \
00151 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/piecewise_construct.h \
00152 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/rel_ops.h \
00153 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/swap.h \
00154 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/to_underlying.h \
00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/compare \
00156 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/initializer_list \
00157
```

```

00157   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocation_guard.h
00158   \_
00159   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_traits.h
00160   \_
00161   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/construct_at.h \
00162   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_traits.h
00163   \_
00164   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator.h \
00165   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/stdexcept \
00166   \_
00167   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/compressed_pair.h
00168   \_
00169   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \
00170   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_safety.h
00171   \_
00172   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iterator \
00173   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/access.h \
00174   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/advance.h \
00175   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__function_like.h \
00176   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/concepts.h \
00177   \_
00178   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/incrementable_traits.h
00179   \_
00180   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/concepts \
00181   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_move.h \
00182   \_
00183   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator_traits.h
00184   \_
00185   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/functional/hash.h \
00186   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstring \
00187   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string.h \
00188   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/string.h \
00189   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__variant/monostate.h \
00190   \_
00191   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/counted_iterator.h
00192   \_
00193   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/default_sentinel.h
00194   \_
00195   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/data.h \
00196   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/distance.h \
00197   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/empty.h \
00198   \_
00199   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/erase_if_container.h
00200   \_
00201   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/front_insert_iterator.h
00202   \_
00203   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/insert_iterator.h
00204   \_
00205   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istreambuf_iterator.h
00206   \_
00207   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istream_iterator.h
00208   \_
00209   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/move_iterator.h
00210   \_
00211   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/next.h \
00212   \_
00213   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostreambuf_iterator.h
00214   \_

```

```

00203 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostream_iterator.h \
00204 \
00205 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/prev.h \
00206 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/projected.h \
00207 \
00208 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_access.h \
00209 \
00210 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/size.h \
00211 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/shared_ptr.h \
00212 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/unique_ptr.h \
00213 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/atomic \
00214 \
00215 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/chrono \
00216 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctime \
00217 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ratio \
00218 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/climits \
00219 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/limits.h \
00220 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/limits.h \
00221 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/limits.h \
00222 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/limits.h \
00223 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_limits.h \
00224 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/syslimits.h \
00225 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_threading_support \
00226 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/errno.h \
00227 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/errno.h \
00228 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/errno.h \
00229 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/sched.h \
00230 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/pthread_impl.h \
00231 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_cond_t.h \
00232 \
00233 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_condattr_t.h \
00234 \
00235 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_key_t.h \
00236 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutex_t.h \
00237 \
00238 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutexattr_t.h \
00239 \
00240 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_t.h \
00241 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/qos.h \
00242 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/qos.h \
00243 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mach_port_t.h \
00244 \
00245 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/temporary_buffer.h \
00246 \
00247 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/uninitialized_algorithms.h \
00248 \
00249 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cassert \
00250 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/assert.h \
00251 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nonlinear solver.h \
00252 /Users/andi/Documents/ruhl_gits/hewes/src/ICSetters.h \
00253 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cmath \
00254 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/math.h \
00255 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/math.h \
00256 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/numbers \
00257 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/array \
00258 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/algorithm \
00259 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bits \
00260 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/functional \
00261 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search.h \
00262 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp.h \
00263 \
00264 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_negate.h \
00265 \
00266 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/bind_front.h \
00267 \
00268 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/perfect_forward.h \
00269 \
00270 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/bind.h \
00271 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binderlist.h

```

```

00265 \Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binder2nd.h
00266 \
00267 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/default_searcher.h
00268 \
00269 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/function.h \
00270 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/identity.h \
00271 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fn.h \
00272 \
00273 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fun_ref.h
00274 \
00275 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/not_fn.h \
00276 \
00277 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_binary_function.h
00278 \
00279 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_unary_function.h
00280 \
00281 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/ranges_operations.h
00282 \
00283 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_negate.h
00284 \
00285 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/adjacent_find.h
00286 \
00287 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/all_of.h \
00288 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/any_of.h \
00289 \
00290 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/binary_search.h
00291 \
00292 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lower_bound.h
00293 \
00294 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/half_positive.h
00295 \
00296 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp_ref_type.h
00297 \
00298 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/clamp.h \
00299 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy.h \
00300 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unwrap_iter.h
00301 \
00302 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_backward.h
00303 \
00304 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_if.h \
00305 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_n.h \
00306 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count.h \
00307 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count_if.h \
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal.h \
00309 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal_range.h
00310 \
00311 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/upper_bound.h
00312 \
00313 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_n.h \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00315 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_end.h \
00317 \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_first_of.h
00319 \
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if.h \
00321 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if_not.h
00322 \
00323 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each.h \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each_n.h
00325 \
00326 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate_n.h
00327 \
00328 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate.h \
00329 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/includes.h \
00330 \
00331 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/inplace_merge.h
00332 \
00333 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min.h \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min_element.h
00335 \
00336 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00337 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate.h \
00338 \
00339 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move_backward.h
00340 \
00341 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/swap_ranges.h
00342 \

```

```
00313 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap.h \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap_until.h \
\
00315 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_partitioned.h \
\
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_permutation.h \
\
00317 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted.h \
00318 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted_until.h \
\
00319 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/iter_swap.h \
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lexicographical_compare.h \
\
00321 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00322 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sift_down.h \
00323 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max.h \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max_element.h \
\
00325 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/merge.h \
00326 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax.h \
00327 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax_element.h \
\
00328 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/mismatch.h \
00329 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/next_permutation.h \
\
00330 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse.h \
00331 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/none_of.h \
00332 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/nth_element.h \
\
00333 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort.h \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort.h \
\
00335 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort_heap.h \
00336 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/pop_heap.h \
00337 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort_copy.h \
\
00338 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h \
00339 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h \
\
00340 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h \
\
00341 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/prev_permutation.h \
\
00342 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/push_heap.h \
00343 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove.h \
00344 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy.h \
\
00345 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy_if.h \
\
00346 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_if.h \
00347 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace.h \
00348 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy.h \
\
00349 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy_if.h \
\
00350 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_if.h \
\
00351 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse_copy.h \
\
00352 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate_copy.h \
\
00353 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sample.h \
00354 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__random/uniform_int_distribution.h \
\
00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search_n.h \
00356 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_difference.h \
\
00357 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_intersection.h
```

```

00358 \
00359 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_symmetric_difference.h \
00360 \
00361 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_union.h \
00362 \
00363 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_left.h \
00364 \
00365 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_right.h \
00366 \
00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shuffle.h \
00368 \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_partition.h \
00370 \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_sort.h \
00372 \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/vector \
00374 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/bit_reference \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwchar \
00376 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwctype \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cctype \
00378 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wctype.h \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wctype.h \
00380 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types_wctrans_t.h \
00381 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/istream \
00382 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ostream \
00383 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/bitset \
00384 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string \
00385 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string_view \
00386 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_view.h \
00387 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ios \
00388 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__locale \
00389 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/mutex \
00390 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__mutex_base \
00391 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/system_error \
00392 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_errc \
00393 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cerrno \
00394 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale.h \
00395 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/locale.h \
00396 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_locale.h \
00397 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale.h \
00398 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_xlocale.h \
00399 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_ctype.h \
00400 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00401 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdio.h \
00402 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdlib.h \
00403 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_string.h \
00404 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_time.h \
00405 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wchar.h \
00406 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00407 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale \
00408 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/streambuf \
00409 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/nl_types.h \
00410 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/types.h \
00411 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_char.h \
00412 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_short.h \
00413 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int.h \
00414 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_caddr_t.h \
00415 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blkcnt_t.h \
00416 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blksize_t.h \
00417 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_gid_t.h \
00418 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_addr_t.h \
00419 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_port_t.h \
00420 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino_t.h \
00421 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino64_t.h \
00422 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_key_t.h \
00423 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_nlink_t.h \
00424 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_useconds_t.h \
00425 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_suseconds_t.h \
00426 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rsize_t.h \
00427 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_errno_t.h \
00428 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_def.h \
00429 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_setsize.h \
00430 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_set.h \
00431 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_clr.h \
00432 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_zero.h \
00433 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_isset.h \
00434 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_copy.h \
00435 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsblkcnt_t.h \

```

```

00436 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsfilcnt_t.h \
00437 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_nl_item.h \
00438 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bsd_locale_defaults.h \
00439 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iostream \
00440 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/sstream \
00441 /opt/homebrew/include/mpi.h \
00442 /opt/homebrew/include/mpi_portable_platform.h \
00443 /opt/homebrew/include/omp.h \
00444 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode.h \
00445 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_ls.h \
00446 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_direct.h \
00447 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_iterative.h \
00448 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_linearsolver.h \
00449 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_matrix.h \
00450 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_proj.h \
00451 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_openmp.h \
00452 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpiplusx.h \
00453 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpimanyvector.h \
00454 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_mpi_types.h \
00455 /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.h \
00456 /Users/andi/Documents/ruhl_gits/hewes/src/Outputters.h \
00457 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/filesystem \
00458 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stack \
00459 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/deque \
00460 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/fstream \
00461 /Users/andi/Documents/ruhl_gits/hewes/src/TimeEvolutionFunctions.h

```

6.28 src/build/CMakeFiles/hewes.dir/SimulationFunctions.cpp.o.d File Reference

6.29 SimulationFunctions.cpp.o.d

[Go to the documentation of this file.](#)

```

00001 CMakeFiles/hewes.dir/SimulationFunctions.cpp.o: \
00002 /Users/andi/Documents/ruhl_gits/hewes/src/SimulationFunctions.cpp \
00003 /Users/andi/Documents/ruhl_gits/hewes/src/SimulationFunctions.h \
00004 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cmath \
00005 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00006 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_config_site \
00007 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00008 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/math.h \
00009 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/math.h \
00010 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/cdefs.h \
00011 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00012 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00013 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/Availability.h \
00014 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityVersions.h \
00015 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityInternal.h \
00016 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdlib.h \
00017 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h \
00018 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types.h \
00019 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00020 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00021 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00022 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00023 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/wait.h \
00024 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_pid_t.h \
00025 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_id_t.h \
00026 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/signal.h \
00027 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/appleapiopts.h \
00028 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/signal.h \
00029 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/signal.h \
00030 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_mcontext.h \
00031 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_mcontext.h \
00032 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/machine/_structs.h \
00033 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/arm/_structs.h \
00034 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00035 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00036 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
00037 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \
00038 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00039 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00040 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00041 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00042 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00043 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00044 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \

```

```

00045 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00046 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_attr_t.h \
00047 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigaltstack.h \
00048 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ucontext.h \
00049 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigset_t.h \
00050 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_size_t.h \
00051 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uid_t.h \
00052 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/resource.h \
00053 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00054 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00055 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00056 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00057 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00058 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00059 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00060 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00061 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h \
00062 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timeval.h \
00063 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/endian.h \
00064 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/endian.h \
00065 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_endian.h \
00066 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/_OSByteOrder.h \
00067 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/arm/OSByteOrder.h \
00068 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/arch.h \
00069 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/alloca.h \
00070 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ct_rune_t.h \
00071 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rune_t.h \
00072 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wchar_t.h \
00073 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_null.h \
00074 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/malloc/_malloc.h \
00075 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_dev_t.h \
00076 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mode_t.h \
00077 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/type_traits \
00078 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstddef \
00079 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/version \
00080 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stddef.h \
00081 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/_stddef_max_align_t.h \
00082 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_nullptr \
00083 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/forward.h \
00084 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_utility/move.h \
00085 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_undef_macros \
00086 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits \
00087 /Users/andi/Documents/ruhl_gits/hewes/src/LatticePatch.h \
00088 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iomanip \
00089 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_string \
00090 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/copy.h \
00091 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_algorithm/unwrap_iter.h \
\
00092 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iterator \
00093 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_debug \
00094 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iosfwd \
00095 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_mbstate_t.h \
00096 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wchar.h \
00097 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00098 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wchar.h \
00099 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mbstate_t.h \
00100 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdarg.h \
00101 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdio.h \
00102 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdio.h \
00103 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_stdio.h \
00104 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_va_list.h \
00105 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/stdio.h \
00106 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctermid.h \
00107 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_off_t.h \
00108 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ssize_t.h \
00109 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/time.h \
00110 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_clock_t.h \
00111 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_time_t.h \
00112 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timespec.h \
00113 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00114 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00115 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wint_t.h \
00116 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctype_t.h \
00117 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctype.h \
00118 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/ctype.h \
00119 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctype.h \
00120 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/runetype.h \
00121 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional_base \
00122 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/binary_function.h \
\
00123 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/invoke.h \
00124 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/weak_result_type.h \
\
00125 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_functional/unary_function.h

```

```
\00126 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/operations.h\n00127 \\\n00128 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/reference_wrapper.h\n00129 \\\n00130 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/addressof.h \\n00131 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_arg_t.h\n00132 \\\n00133 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdlib \\n00134 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/new \\n00135 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/typeinfo \\n00136 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \\n00137 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility \\n00138 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \\n00139 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/as_const.h \\n00140 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/cmp.h \\n00141 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/decinal.h \\n00142 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/exchange.h \\n00143 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/in_place.h \\n00144 \\\n00145 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/integer_sequence.h\n00146 \\\n00147 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/piecewise_construct.h\n00148 \\\n00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/rel_ops.h \\n00150 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/swap.h \\n00151 \\\n00152 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/initializer_list \\n00153 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/access.h \\n00154 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/advance.h \\n00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__function_like.h \\n00156 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/concepts.h \\n00157 \\\n00158 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/incrementable_traits.h\n00159 \\\n00160 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_move.h \\n00161 \\\n00162 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator_traits.h\n00163 \\\n00164 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/readable_traits.h\n00165 \\\n00166 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_traits.h\n00167 \\\n00168 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/back_insert_iterator.h\n00169 \\\n00170 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator.h \\n00171 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/hash.h \\n00172 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstring \\n00173 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string.h \\n00174 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/string.h \\n00175 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/strings.h \\n00176 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__variant/monostate.h \\n00177 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \\n00178 \\\n00179 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/counted_iterator.h\n00180 \\\n00181 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/default_sentinel.h\n00182 \\\n00183 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/data.h \\n00184 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/distance.h \\n00185
```

```
00182 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/empty.h \
00183 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/erase_if_container.h \
00184 \
00185 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/front_insert_iterator.h \
00186 \
00187 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/insert_iterator.h \
00188 \
00189 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istreambuf_iterator.h \
00190 \
00191 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istream_iterator.h \
00192 \
00193 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/next.h \
00194 \
00195 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostreambuf_iterator.h \
00196 \
00197 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/prev.h \
00198 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/projected.h \
00199 \
00200 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_access.h \
00201 \
00202 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_iterator.h \
00203 \
00204 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_backward.h \
00205 \
00206 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp.h \
00207 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_end.h \
00208 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min.h \
00209 \
00210 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min_element.h \
00211 \
00212 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types_wctrans_t.h \
00213 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iostream \
00214 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ostream \
00215 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/bitset \
00216 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bit_reference \
00217 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_bits \
00218 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/algorithm \
00219 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/functional \
00220 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search.h \
00221 \
00222 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_negate.h \
00223 \
00224 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/bind_front.h \
00225 \
00226 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binderlist.h \
00227 \
00228 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/default_searcher.h \
00229 \
00230 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/function.h \
00231 \
00232 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_traits.h \
00233 \
00234 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/construct_at.h \
00235 \
00236 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/compressed_pair.h
```

```

00232   \
00233   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/shared_ptr.h \
00234   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator.h \
00235   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdexcept \
00236   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/unique_ptr.h \
00237   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/atomic \
00238   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__thread/poll_with_backoff.h \
00239   \
00240   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/chrono \
00241   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctime \
00242   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ratio \
00243   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/climits \
00244   /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/limits.h \
00245   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/limits.h \
00246   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/limits.h \
00247   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/limits.h \
00248   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_limits.h \
00249   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/syslimits.h \
00250   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__threading_support \
00251   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/errno.h \
00252   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/errno.h \
00253   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/errno.h \
00254   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/sched.h \
00255   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/pthread_impl.h \
00256   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_cond_t.h \
00257   \
00258   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_condattr_t.h \
00259   \
00260   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_key_t.h \
00261   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutex_t.h \
00262   \
00263   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutexattr_t.h \
00264   \
00265   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_t.h \
00266   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/qos.h \
00267   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/qos.h \
00268   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mach_port_t.h \
00269   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sched.h \
00270   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory \
00271   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_safety.h \
00272   \
00273   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/raw_storage_iterator.h \
00274   \
00275   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/temporary_buffer.h \
00276   \
00277   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/uninitialized_algorithms.h \
00278   \
00279   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cassert \
00280   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/assert.h \
00281   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/identity.h \
00282   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fn.h \
00283   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fun_ref.h \
00284   \
00285   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/not_fn.h \
00286   \
00287   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_binary_function.h \
00288   \
00289   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_unary_function.h \
00290   \
00291   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/ranges_operations.h \
00292   \
00293   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_negate.h \
00294   \
00295   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/adjacent_find.h \
00296   \
00297   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/all_of.h \
00298   /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/any_of.h \

```

```

00287     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/binary_search.h
00288     \
00289     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lower_bound.h
00290     \
00291     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/half_positive.h
00292     \
00293     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp_ref_type.h
00294     \
00295     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/clamp.h \
00296     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_if.h \
00297     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count.h \
00298     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count_if.h \
00299     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal.h \
00300     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal_range.h
00301     \
00302     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/upper_bound.h
00303     \
00304     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill.h \
00305     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00306     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if.h \
00307     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if_not.h
00308     \
00309     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each.h \
00310     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each_n.h
00311     \
00312     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate_n.h
00313     \
00314     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate.h \
00315     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/includes.h \
00316     \
00317     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/inplace_merge.h
00318     \
00319     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00320     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate.h \
00321     \
00322     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move_backward.h
00323     \
00324     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/swap_ranges.h
00325     \
00326     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap.h \
00327     \
00328     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap_until.h
00329     \
00330     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_partitioned.h
00331     \
00332     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_permutation.h
00333     \
00334     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted.h \
00335     \
00336     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted_until.h
00337     \
00338     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/iter_swap.h \
00339     \
00340     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lexicographical_compare.h
00341     \
00342     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00343     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sift_down.h \
00344     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max.h \
00345     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max_element.h
00346     \
00347     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/merge.h \
00348     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax.h \
00349     \
00350     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax_element.h
00351     \
00352     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/mismatch.h \
00353     \
00354     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/next_permutation.h
00355     \
00356     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse.h \
00357     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/none_of.h \
00358     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/nth_element.h
00359     \
00360     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort.h \
00361     \
00362     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort.h
00363     \
00364     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort_copy.h
00365     \
00366     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/pop_heap.h \
00367     \
00368     /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h

```

```
00337 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h \
00338 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h \
\
00339 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h \
\
00340 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/prev_permutation.h \
\
00341 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/push_heap.h \
00342 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove.h \
00343 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy.h \
\
00344 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy_if.h \
\
00345 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_if.h \
00346 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace.h \
00347 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy.h \
\
00348 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy_if.h \
\
00349 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_if.h \
\
00350 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse_copy.h \
\
00351 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate_copy.h \
\
00352 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sample.h \
00353 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__random/uniform_int_distribution.h \
\
00354 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search_n.h \
00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_difference.h \
\
00356 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_intersection.h \
\
00357 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_symmetric_difference.h \
\
00358 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_union.h \
00359 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_left.h \
\
00360 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_right.h \
\
00361 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shuffle.h \
00362 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_partition.h \
\
00363 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_sort.h \
\
00364 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/transform.h \
00365 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unique_copy.h \
\
00366 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unique.h \
00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string \
00368 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string_view \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_view.h \
00370 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ios \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__locale \
00372 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/mutex \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__mutex_base \
00374 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/system_error \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__errc \
00376 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cerrno \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale.h \
00378 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/locale.h \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_locale.h \
00380 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale.h \
00381 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_xlocale.h \
00382 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_ctype.h \
00383 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00384 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdio.h \
00385 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdlib.h \
00386 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_string.h \
00387 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_time.h \
00388 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wchar.h \
00389 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00390 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale \
00391 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/streambuf \
00392 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/nl_types.h \
```

```

00393 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/types.h \
00394 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_char.h \
00395 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_short.h \
00396 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int.h \
00397 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_caddr_t.h \
00398 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blkcnt_t.h \
00399 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blksize_t.h \
00400 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_gid_t.h \
00401 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_addr_t.h \
00402 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_port_t.h \
00403 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino_t.h \
00404 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino64_t.h \
00405 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_key_t.h \
00406 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_nlink_t.h \
00407 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_useconds_t.h \
00408 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_suseconds_t.h \
00409 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rsize_t.h \
00410 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_errno_t.h \
00411 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_def.h \
00412 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_setsize.h \
00413 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_set.h \
00414 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_clr.h \
00415 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_zero.h \
00416 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_isset.h \
00417 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_copy.h \
00418 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsblkcnt_t.h \
00419 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsfilcnt_t.h \
00420 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_nl_item.h \
00421 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bsd_locale_defaults.h \
00422 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iostream \
00423 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/sstream \
00424 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/array \
00425 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/vector \
00426 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__split_buffer \
00427 /opt/homebrew/include/mpi.h \
00428 /opt/homebrew/include/mpi_portable_platform.h \
00429 /opt/homebrew/include/omp.h \
00430 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode.h \
00431 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_context.h \
00432 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_logger.h \
00433 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00434 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00435 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00436 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00437 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/float.h \
00438 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \
00439 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_profiler.h \
00440 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nvvector.h \
00441 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nondirectivesolver.h \
00442 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_ls.h \
00443 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_direct.h \
00444 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_iterative.h \
00445 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_linearsolver.h \
00446 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_matrix.h \
00447 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_proj.h \
00448 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_openmp.h \
00449 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpiplusx.h \
00450 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpmimanyvector.h \
00451 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_mpi_types.h \
00452 /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.h \
00453 /Users/andi/Documents/ruhl_gits/hewes/src/SimulationClass.h \
00454 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sunnonlinsol/sunnonlinsol_fixedpoint.h \
00455 /Users/andi/Documents/ruhl_gits/hewes/src/ICSetters.h \
00456 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/numbers \
00457 /Users/andi/Documents/ruhl_gits/hewes/src/Outputters.h \
00458 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/filesystem \
00459 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stack \
00460 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/deque \
00461 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/fstream \
00462 /Users/andi/Documents/ruhl_gits/hewes/src/TimeEvolutionFunctions.h

```

6.30 src/build/CMakeFiles/hewes.dir/TimeEvolutionFunctions.cpp.o.d File Reference

6.31 TimeEvolutionFunctions.cpp.o.d

[Go to the documentation of this file.](#)

```

00001 CMakeFiles/hewes.dir/TimeEvolutionFunctions.cpp.o: \
00002 /Users/andi/Documents/ruhl_gits/hewes/src/TimeEvolutionFunctions.cpp \
00003 /Users/andi/Documents/ruhl_gits/hewes/src/TimeEvolutionFunctions.h \
00004 /Users/andi/Documents/ruhl_gits/hewes/src/LatticePatch.h \
00005 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iomanip \
00006 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config \
00007 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__config_site \
00008 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread.h \
00009 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__string \
00010 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy.h \
00011 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/unwrap_iter.h \
00012 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iterator \
00013 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__debug \
00014 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iosfwd \
00015 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__mbstate_t.h \
00016 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/wchar.h \
00017 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stddef.h \
00018 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/_stddef_max_align_t.h \
00019 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__nullptr \
00020 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wchar.h \
00021 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types.h \
00022 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types.h \
00023 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_cdefs.h \
00024 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_symbol_aliasing.h \
00025 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_posix_availability.h \
00026 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_types.h \
00027 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00028 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_types.h \
00029 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_types.h \
00030 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/Availability.h \
00031 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityVersions.h \
00032 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/AvailabilityInternal.h \
00033 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_null.h \
00034 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_size_t.h \
00035 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mbstate_t.h \
00036 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/types.h \
00037 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/types.h \
00038 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int8_t.h \
00039 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int16_t.h \
00040 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int32_t.h \
00041 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_int64_t.h \
00042 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int8_t.h \
00043 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int16_t.h \
00044 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int32_t.h \
00045 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int64_t.h \
00046 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_intptr_t.h \
00047 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uintptr_t.h \
00048 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ct_rune_t.h \
00049 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rune_t.h \
00050 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wchar_t.h \
00051 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdarg.h \
00052 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdio.h \
00053 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdio.h \
00054 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_stdio.h \
00055 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_va_list.h \
00056 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_stdio.h \
00057 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctermid.h \
00058 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_off_t.h \
00059 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ssize_t.h \
00060 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/time.h \
00061 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_clock_t.h \
00062 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_time_t.h \
00063 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timespec.h \
00064 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00065 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_wctype.h \
00066 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_wint_t.h \
00067 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctype_t.h \
00068 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctype.h \
00069 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/ctype.h \
00070 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_ctype.h \
00071 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/runetype.h \
00072 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional_base \
00073 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_function.h \
00074 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/invoke.h \
00075 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/weak_result_type.h \
00076 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_function.h \
00077 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__undef_macros \
00078 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/type_traits \
00079 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00080 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/version \

```

```

00081 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/forward.h \
00082 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/move.h \
00083 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/operations.h \
00084 \
00085 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/reference_wrapper.h \
00086 \
00087 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/addressof.h \
00088 \
00089 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_arg_t.h \
00090 \
00091 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdlib \
00092 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdlib.h \
00093 \
00094 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_pid_t.h \
00095 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_id_t.h \
00096 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/signal.h \
00097 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/applecioppts.h \
00098 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/signal.h \
00099 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/signal.h \
00100 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/_mcontext.h \
00101 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_mcontext.h \
00102 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/machine/_structs.h \
00103 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/mach/arm/_structs.h \
00104 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_attr_t.h \
00105 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigaltstack.h \
00106 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ucontext.h \
00107 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_sigset_t.h \
00108 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_uid_t.h \
00109 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/resource.h \
00110 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stdint.h \
00111 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/stdint.h \
00112 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/stdint.h \
00113 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint8_t.h \
00114 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint16_t.h \
00115 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint32_t.h \
00116 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uint64_t.h \
00117 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_intmax_t.h \
00118 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_uintmax_t.h \
00119 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_timeval.h \
00120 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/machine/endian.h \
00121 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/endian.h \
00122 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_endian.h \
00123 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/_OSByteOrder.h \
00124 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/libkern/arm/OSByteOrder.h \
00125 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/arch.h \
00126 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/alloca.h \
00127 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/malloc/_malloc.h \
00128 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_dev_t.h \
00129 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mode_t.h \
00130 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/new \
00131 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/typeinfo \
00132 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdint \
00133 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/utility \
00134 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__tuple \
00135 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/as_const.h \
00136 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/cmp.h \
00137 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits \
00138 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/declval.h \
00139 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/exchange.h \
00140 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/in_place.h \
00141 \
00142 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/integer_sequence.h \
00143 \
00144 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unwrap_ref.h \
00145 \
00146 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/piecewise_construct.h \
00147 \
00148 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/compare \
00149 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/initializer_list \
00150 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/access.h \
00151 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/advance.h \
00152 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__function_like.h \
00153 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/concepts.h \
00154 \
00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/incrementable_traits.h

```

```
\00155 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/concepts \
00156 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_move.h \
00157 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator_traits.h \
\00158 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/readable_traits.h \
\00159 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_traits.h \
\00160 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/back_insert_iterator.h \
\00161 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iterator.h \
00162 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/common_iterator.h \
\00163 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/iter_swap.h \
00164 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/access.h \
00165 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_borrowed_range.h \
\00166 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__utility/__decay_copy.h \
\00167 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/variant \
00168 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/hash.h \
00169 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstring \
00170 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string.h \
00171 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/string.h \
00172 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/strings.h \
00173 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__variant/monostate.h \
00174 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/tuple \
00175 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/counted_iterator.h \
\00176 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/default_sentinel.h \
\00177 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/data.h \
00178 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/distance.h \
00179 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/empty.h \
00180 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/erase_if_container.h \
\00181 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/front_insert_iterator.h \
\00182 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/insert_iterator.h \
\00183 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istreambuf_iterator.h \
\00184 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/istream_iterator.h \
\00185 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/move_iterator.h \
\00186 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/next.h \
00187 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostreambuf_iterator.h \
\00188 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/ostream_iterator.h \
\00189 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/prev.h \
00190 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/projected.h \
00191 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_access.h \
\00192 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/reverse_iterator.h \
\00193 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/size.h \
00194 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__iterator/wrap_iter.h \
00195 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_backward.h \
\00196 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_n.h \
00197 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill_n.h \
00198 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_first_of.h \
\00199 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp.h \
00200 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_end.h \\\
```

```

00201 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min.h \
00202 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/min_element.h \
00203 \
00204 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cstdio \
00205 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwchar \
00206 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cwctype \
00207 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cctype \
00208 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/wctype.h \
00209 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_wctrans_t.h \
00210 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/istream \
00211 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ostream \
00212 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/bitset \
00213 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bit_reference \
00214 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bits \
00215 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/algorithm \
00216 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/functional \
00217 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search.h \
00218 \
00219 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binary_negate.h \
00220 \
00221 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/perfect_forward.h \
00222 \
00223 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/bind.h \
00224 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binder1st.h \
00225 \
00226 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/binder2nd.h \
00227 \
00228 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/default_searcher.h \
00229 \
00230 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/function.h \
00231 \
00232 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/allocator_traits.h \
00233 \
00234 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/construct_at.h \
00235 \
00236 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/compressed_pair.h \
00237 \
00238 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/shared_ptr.h \
00239 \
00240 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/unique_ptr.h \
00241 \
00242 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__thread/poll_with_backoff.h \
00243 \
00244 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/chrono \
00245 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ctime \
00246 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ratio \
00247 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/climits \
00248 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits.h \
00249 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/limits.h \
00250 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/limits.h \
00251 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/arm/_limits.h \
00252 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/syslimits.h \
00253 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/_threading_support \
00254 \
00255 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/errno.h \
00256 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/errno.h \
00257 \
00258 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/sched.h \
00259 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/thread_attr.h \
00260 \
00261 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_cond_t.h \
00262 \
00263 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_condattr_t.h \
00264 \
00265 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_key_t.h \
00266 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutex_t.h \
00267 \
00268 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_mutexattr_t.h \
00269 \
00270 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_once_t.h \
00271 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_rwlock_t.h \
00272 \
00273 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_rwlockattr_t.h \
00274 \
00275 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_pthread/_pthread_t.h \

```

```
00262 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/pthread/qos.h \
00263 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/qos.h \
00264 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_mach_port_t.h \
00265 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sched.h \
00266 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/memory \
00267 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/pointer_safety.h \
00268 \
00269 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/raw_storage_iterator.h \
00270 \
00271 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/temporary_buffer.h \
00272 \
00273 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__memory/uninitialized_algorithms.h \
00274 \
00275 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cassert \
00276 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/assert.h \
00277 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/identity.h \
00278 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fn.h \
00279 \
00280 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/mem_fun_ref.h \
00281 \
00282 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/not_fn.h \
00283 \
00284 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_binary_function.h \
00285 \
00286 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/pointer_to_unary_function.h \
00287 \
00288 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/ranges_operations.h \
00289 \
00290 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__functional/unary_negate.h \
00291 \
00292 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/adjacent_find.h \
00293 \
00294 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/all_of.h \
00295 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/any_of.h \
00296 \
00297 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/binary_search.h \
00298 \
00299 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lower_bound.h \
00300 \
00301 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/half_positive.h \
00302 \
00303 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/comp_ref_type.h \
00304 \
00305 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/clamp.h \
00306 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/copy_if.h \
00307 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count.h \
00308 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/count_if.h \
00309 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal.h \
00310 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/equal_range.h \
00311 \
00312 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/upper_bound.h \
00313 \
00314 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/fill.h \
00315 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find.h \
00316 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if.h \
00317 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/find_if_not.h \
00318 \
00319 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each.h \
00320 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/for_each_n.h \
00321 \
00322 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate_n.h \
00323 \
00324 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/generate.h \
00325 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/includes.h \
00326 \
00327 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/inplace_merge.h \
00328 \
00329 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move.h \
00330 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate.h \
00331 \
00332 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/move_backward.h \
00333 \
00334 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/swap_ranges.h \
00335 \
00336 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap.h \
00337 \
00338 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_heap_until.h \
00339 \
00340
```

```

00311 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_partitioned.h \
00312 \
00313 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_permutation.h \
00314 \
00315 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted.h \
00316 \
00317 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/is_sorted_until.h \
00318 \
00319 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/iter_swap.h \
00320 \
00321 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/lexicographical_compare.h \
00322 \
00323 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/make_heap.h \
00324 \
00325 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sift_down.h \
00326 \
00327 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max.h \
00328 \
00329 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/max_element.h \
00330 \
00331 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax.h \
00332 \
00333 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/minmax_element.h \
00334 \
00335 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/mismatch.h \
00336 \
00337 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/next_permutation.h \
00338 \
00339 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse.h \
00340 \
00341 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/none_of.h \
00342 \
00343 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/nth_element.h \
00344 \
00345 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort.h \
00346 \
00347 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort.h \
00348 \
00349 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sort_heap.h \
00350 \
00351 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/pop_heap.h \
00352 \
00353 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partial_sort_copy.h \
00354 \
00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition.h \
00356 \
00357 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_copy.h \
00358 \
00359 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/partition_point.h \
00360 \
00361 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/prev_permutation.h \
00362 \
00363 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/push_heap.h \
00364 \
00365 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove.h \
00366 \
00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy.h \
00368 \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_copy_if.h \
00370 \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/remove_if.h \
00372 \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace.h \
00374 \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy.h \
00376 \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_copy_if.h \
00378 \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/replace_if.h \
00380 \
00381 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/reverse_copy.h \
00382 \
00383 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/rotate_copy.h \
00384 \
00385 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/sample.h \
00386 \
00387 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__random/uniform_int_distribution.h \
00388 \
00389 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/search_n.h \
00390 \
00391 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_difference.h \
00392 \
00393 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_intersection.h \
00394 \
00395 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_symmetric_difference.h \
00396 \

```

```

00355 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/set_union.h \
00356 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_left.h \
00357 \
00358 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/shift_right.h \
00359 \
00360 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_partition.h \
00361 \
00362 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__algorithm/stable_sort.h \
00363 \
00364 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string \
00365 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/string_view \
00366 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__ranges/enable_view.h \
00367 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/ios \
00368 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__locale \
00369 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/mutex \
00370 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__mutex_base \
00371 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/system_error \
00372 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__errc \
00373 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cerrno \
00374 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale.h \
00375 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/locale.h \
00376 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_locale.h \
00377 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale.h \
00378 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_xlocale.h \
00379 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_ctype.h \
00380 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00381 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdio.h \
00382 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_stdlib.h \
00383 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_string.h \
00384 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_time.h \
00385 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wchar.h \
00386 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/xlocale/_wctype.h \
00387 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/locale \
00388 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/streambuf \
00389 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/nl_types.h \
00390 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/types.h \
00391 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_char.h \
00392 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_short.h \
00393 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_u_int.h \
00394 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_caddr_t.h \
00395 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blkcnt_t.h \
00396 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_blksize_t.h \
00397 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_gid_t.h \
00398 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_addr_t.h \
00399 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_in_port_t.h \
00400 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino_t.h \
00401 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_ino64_t.h \
00402 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_key_t.h \
00403 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_nlink_t.h \
00404 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_useconds_t.h \
00405 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_suseconds_t.h \
00406 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_rsize_t.h \
00407 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_errno_t.h \
00408 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_def.h \
00409 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_setsize.h \
00410 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_set.h \
00411 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_clr.h \
00412 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_zero.h \
00413 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_isset.h \
00414 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fd_copy.h \
00415 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsblkcnt_t.h \
00416 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/sys/_types/_fsfilcnt_t.h \
00417 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/_types/_nl_item.h \
00418 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__bsd_locale_defaults.h \
00419 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/iostream \
00420 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/sstream \
00421 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/array \
00422 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/vector \
00423 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/__split_buffer \
00424 /opt/homebrew/include/mpi.h \
00425 /opt/homebrew/include/mpi_portable_platform.h \
00426 /opt/homebrew/include/omp.h \
00427 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode.h \
00428 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_context.h \
00429 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_logger.h \
00430 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_config.h \
00431 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_export.h \
00432 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_types.h \
00433 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/float.h \
00434 /Library/Developer/CommandLineTools/usr/lib/clang/14.0.0/include/float.h \
00435 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/float.h \

```

```

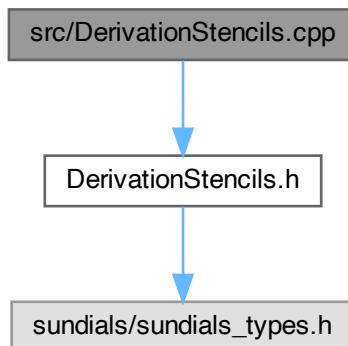
00436 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_profiler.h \
00437 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nvvector.h \
00438 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_nonlinear solver.h \
00439 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_ls.h \
00440 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_direct.h \
00441 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_iterative.h \
00442 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_linearsolver.h \
00443 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_matrix.h \
00444 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/cvode/cvode_proj.h \
00445 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_openmp.h \
00446 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpirus.h \
00447 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/nvector/nvector_mpianyvector.h \
00448 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sundials/sundials_mpi_types.h \
00449 /Users/andi/Documents/ruhl_gits/hewes/src/DerivationStencils.h \
00450 /Users/andi/Documents/ruhl_gits/hewes/src/SimulationClass.h \
00451 /Users/andi/Documents/sundials-6.3.0/INSTDIR_Rel/include/sunnonlinsol/sunnonlinsol_fixedpoint.h \
00452 /Users/andi/Documents/ruhl_gits/hewes/src/ICSetters.h \
00453 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/cmath \
00454 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/math.h \
00455 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/math.h \
00456 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/numbers \
00457 /Users/andi/Documents/ruhl_gits/hewes/src/Outputters.h \
00458 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/filesystem \
00459 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/stack \
00460 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/deque \
00461 /Library/Developer/CommandLineTools/SDKs/MacOSX12.3.sdk/usr/include/c++/v1/fstream

```

6.32 src/DerivationStencils.cpp File Reference

Empty. All definitions in the header.

```
#include "DerivationStencils.h"
Include dependency graph for DerivationStencils.cpp:
```



6.32.1 Detailed Description

Empty. All definitions in the header.

Definition in file [DerivationStencils.cpp](#).

6.33 DerivationStencils.cpp

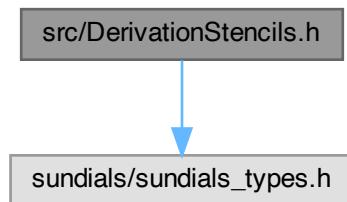
[Go to the documentation of this file.](#)

```
00001 ///////////////////////////////////////////////////////////////////
00002 /// @file DerivationStencils.cpp
00003 /// @brief Empty. All definitions in the header.
00004 ///////////////////////////////////////////////////////////////////
00005 #include "DerivationStencils.h"
```

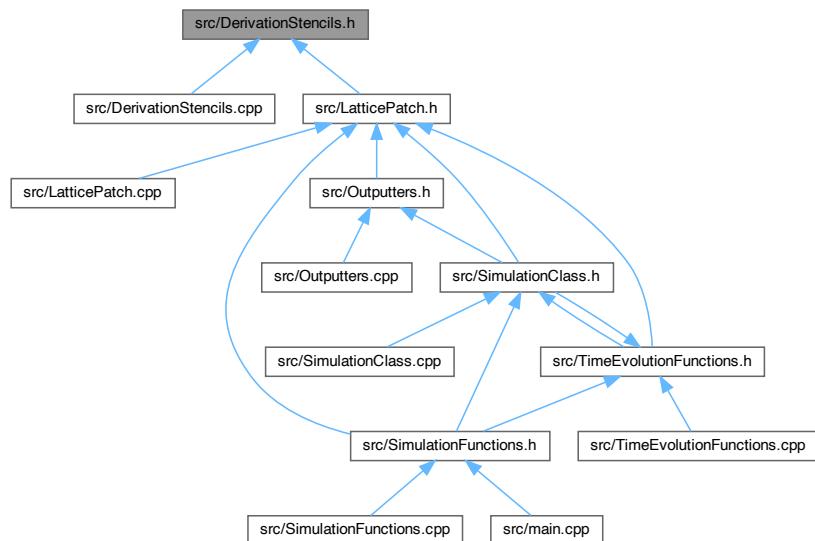
6.34 src/DerivationStencils.h File Reference

Definition of derivation stencils from order 1 to 13.

```
#include <sundials/sundials_types.h>
Include dependency graph for DerivationStencils.h:
```



This graph shows which files directly or indirectly include this file:



Functions

- sunrealtype [s9b](#) (sunrealtype const *udata)
- sunrealtype [s10f](#) (sunrealtype const *udata)
- sunrealtype [s10c](#) (sunrealtype const *udata)
- sunrealtype [s10b](#) (sunrealtype const *udata)
- sunrealtype [s11f](#) (sunrealtype const *udata)
- sunrealtype [s11b](#) (sunrealtype const *udata)
- sunrealtype [s12f](#) (sunrealtype const *udata)
- sunrealtype [s12c](#) (sunrealtype const *udata)
- sunrealtype [s12b](#) (sunrealtype const *udata)
- sunrealtype [s13f](#) (sunrealtype const *udata)
- sunrealtype [s13b](#) (sunrealtype const *udata)

6.34.1 Detailed Description

Definition of derivation stencils from order 1 to 13.

Definition in file [DerivationStencils.h](#).

6.34.2 Function Documentation

6.34.2.1 [s10b\(\)](#) [1/2]

```
sunrealtype s10b (
    unrealtype const * udata ) [inline]
```

Definition at line 275 of file [DerivationStencils.h](#).
00276 { [return](#) [s10b](#)(udata, 6); }

References [s10b\(\)](#).

Here is the call graph for this function:



6.34.2.2 s10b() [2/2]

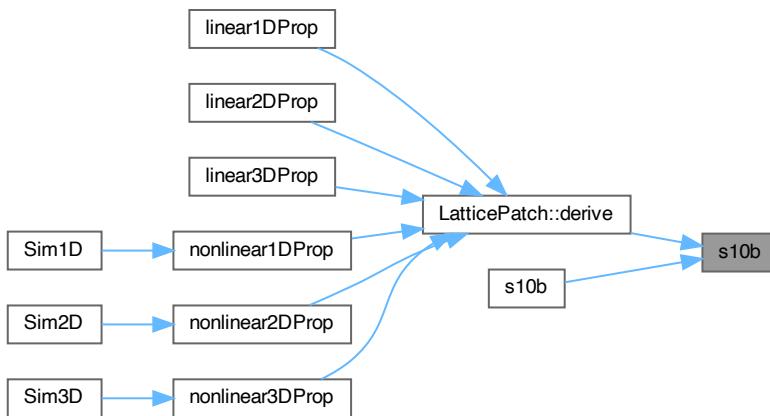
```
sunrealtype s10b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 169 of file [DerivationStencils.h](#).

```
00169
00170     return 1.0 / 840.0 * udata[-4 * dPD] - 1.0 / 63.0 * udata[-3 * dPD] +
00171         3.0 / 28.0 * udata[-2 * dPD] - 4.0 / 7.0 * udata[-1 * dPD] -
00172         11.0 / 30.0 * udata[0] + 6.0 / 5.0 * udata[1 * dPD] -
00173         1.0 / 2.0 * udata[2 * dPD] + 4.0 / 21.0 * udata[3 * dPD] -
00174         3.0 / 56.0 * udata[4 * dPD] + 1.0 / 105.0 * udata[5 * dPD] -
00175         1.0 / 1260.0 * udata[6 * dPD];
00176 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s10b\(\)](#).

Here is the caller graph for this function:



6.34.2.3 s10c() [1/2]

```
sunrealtype s10c (
    unrealtype const * udata ) [inline]
```

Definition at line 273 of file [DerivationStencils.h](#).

```
00274 { return s10c(udata, 6); }
```

References [s10c\(\)](#).

Here is the call graph for this function:



6.34.2.4 s10c() [2/2]

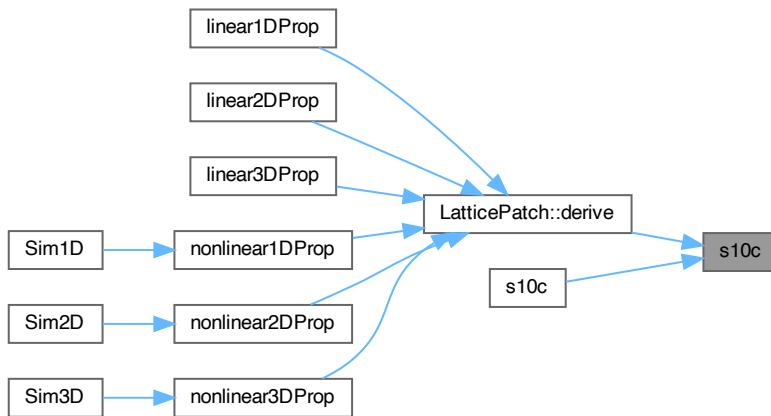
```
sunrealtype s10c (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 161 of file [DerivationStencils.h](#).

```
00161
00162     return -1.0 / 1260.0 * udata[-5 * dPD] + 5.0 / 504.0 * udata[-4 * dPD] -
00163         5.0 / 84.0 * udata[-3 * dPD] + 5.0 / 21.0 * udata[-2 * dPD] -
00164         5.0 / 6.0 * udata[-1 * dPD] + 0 + 5.0 / 6.0 * udata[1 * dPD] -
00165         5.0 / 21.0 * udata[2 * dPD] + 5.0 / 84.0 * udata[3 * dPD] -
00166         5.0 / 504.0 * udata[4 * dPD] + 1.0 / 1260.0 * udata[5 * dPD];
00167 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s10c\(\)](#).

Here is the caller graph for this function:



6.34.2.5 s10f() [1/2]

```
sunrealtype s10f (
    unrealtype const * udata ) [inline]
```

Definition at line 271 of file [DerivationStencils.h](#).

```
00272 { return s10f(udata, 6); }
```

References [s10f\(\)](#).

Here is the call graph for this function:



6.34.2.6 s10f() [2/2]

```

sunrealtype s10f (
    unrealtype const * udata,
    const int dPD ) [inline]
  
```

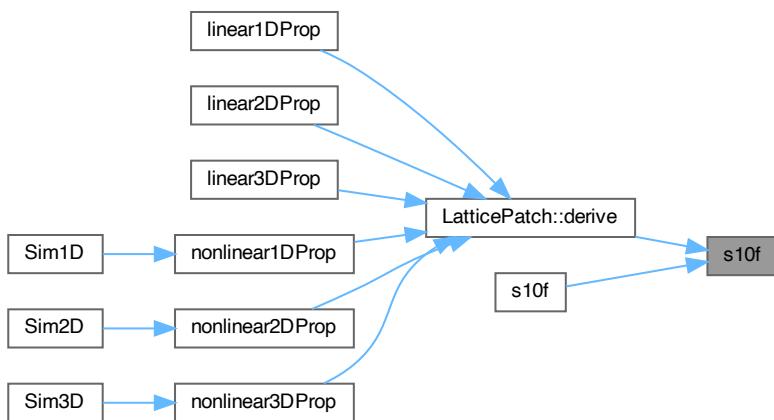
Definition at line 152 of file [DerivationStencils.h](#).

```

00152     {
00153     return 1.0 / 1260.0 * udata[-6 * dPD] - 1.0 / 105.0 * udata[-5 * dPD] +
00154         3.0 / 56.0 * udata[-4 * dPD] - 4.0 / 21.0 * udata[-3 * dPD] +
00155         1.0 / 2.0 * udata[-2 * dPD] - 6.0 / 5.0 * udata[-1 * dPD] +
00156         11.0 / 30.0 * udata[0] + 4.0 / 7.0 * udata[1 * dPD] -
00157         3.0 / 28.0 * udata[2 * dPD] + 1.0 / 63.0 * udata[3 * dPD] -
00158         1.0 / 840.0 * udata[4 * dPD];
00159 }
  
```

Referenced by [LatticePatch::derive\(\)](#), and [s10f\(\)](#).

Here is the caller graph for this function:



6.34.2.7 s11b() [1/2]

```
sunrealtype s11b (
    unrealtype const * udata ) [inline]
```

Definition at line 279 of file [DerivationStencils.h](#).

```
00280 { return s11b(udata, 6); }
```

References [s11b\(\)](#).

Here is the call graph for this function:



6.34.2.8 s11b() [2/2]

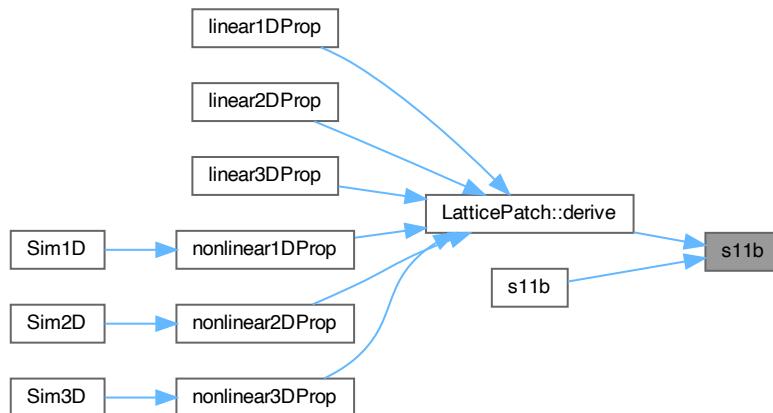
```
sunrealtype s11b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 187 of file [DerivationStencils.h](#).

```
00187
00188     return -1.0 / 2310.0 * udata[-5 * dPD] + 1.0 / 168.0 * udata[-4 * dPD] -
00189         5.0 / 126.0 * udata[-3 * dPD] + 5.0 / 28.0 * udata[-2 * dPD] -
00190         5.0 / 7.0 * udata[-1 * dPD] - 1.0 / 6.0 * udata[0] + udata[1 * dPD] -
00191         5.0 / 14.0 * udata[2 * dPD] + 5.0 / 42.0 * udata[3 * dPD] -
00192         5.0 / 168.0 * udata[4 * dPD] + 1.0 / 210.0 * udata[5 * dPD] -
00193         1.0 / 2772.0 * udata[6 * dPD];
00194 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s11b\(\)](#).

Here is the caller graph for this function:



6.34.2.9 s11f() [1/2]

```
sunrealtype s11f (
    unrealtype const * udata ) [inline]
```

Definition at line 277 of file [DerivationStencils.h](#).
 00278 { **return** s11f(udata, 6); }

References [s11f\(\)](#).

Here is the call graph for this function:



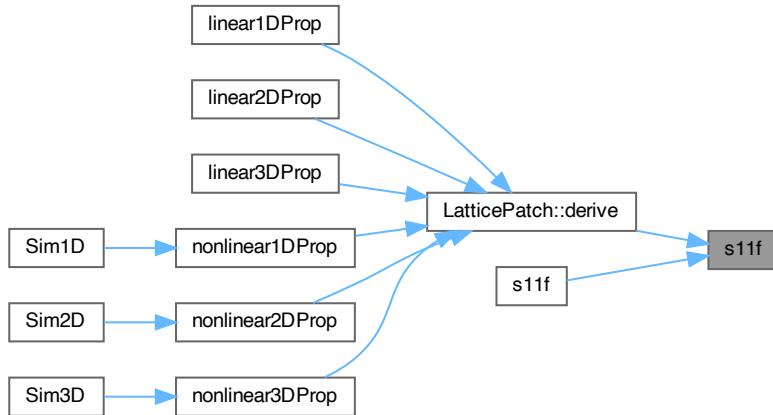
6.34.2.10 s11f() [2/2]

```
sunrealtype s11f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 178 of file [DerivationStencils.h](#).
 00178 {
 00179 **return** 1.0 / 2772.0 * udata[-6 * dPD] - 1.0 / 210.0 * udata[-5 * dPD] +
 00180 5.0 / 168.0 * udata[-4 * dPD] - 5.0 / 42.0 * udata[-3 * dPD] +
 00181 5.0 / 14.0 * udata[-2 * dPD] - 1.0 / 1.0 * udata[-1 * dPD] +
 00182 1.0 / 6.0 * udata[0] + 5.0 / 7.0 * udata[1 * dPD] -
 00183 5.0 / 28.0 * udata[2 * dPD] + 5.0 / 126.0 * udata[3 * dPD] -
 00184 1.0 / 168.0 * udata[4 * dPD] + 1.0 / 2310.0 * udata[5 * dPD];
 00185 }

Referenced by [LatticePatch::derive\(\)](#), and [s11f\(\)](#).

Here is the caller graph for this function:



6.34.2.11 `s12b()` [1/2]

```
sunrealtype s12b (
    unrealtype const * udata ) [inline]
```

Definition at line 285 of file [DerivationStencils.h](#).
00286 { **return** s12b(udata, 6); }

References [s12b\(\)](#).

Here is the call graph for this function:



6.34.2.12 s12b() [2/2]

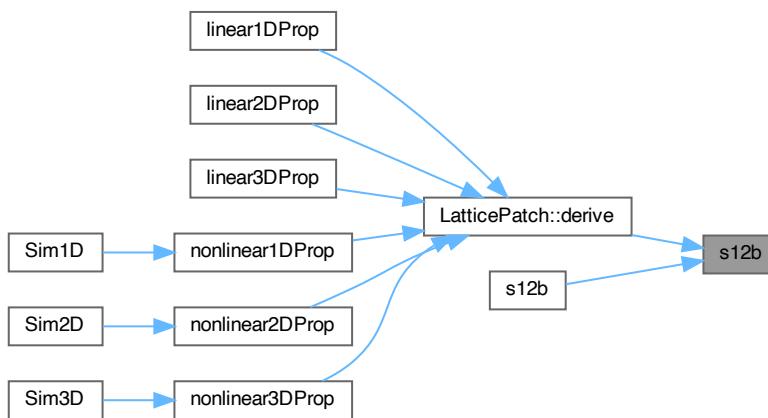
```
sunrealtype s12b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 215 of file [DerivationStencils.h](#).

```
00215
00216     return -1.0 / 3960.0 * udata[-5 * dPD] + 1.0 / 264.0 * udata[-4 * dPD] -
00217         1.0 / 36.0 * udata[-3 * dPD] + 5.0 / 36.0 * udata[-2 * dPD] -
00218         5.0 / 8.0 * udata[-1 * dPD] - 13.0 / 42.0 * udata[0] +
00219         7.0 / 6.0 * udata[1 * dPD] - 1.0 / 2.0 * udata[2 * dPD] +
00220         5.0 / 24.0 * udata[3 * dPD] - 5.0 / 72.0 * udata[4 * dPD] +
00221         1.0 / 60.0 * udata[5 * dPD] - 1.0 / 396.0 * udata[6 * dPD] +
00222         1.0 / 5544.0 * udata[7 * dPD];
00223 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s12b\(\)](#).

Here is the caller graph for this function:



6.34.2.13 s12c() [1/2]

```
sunrealtype s12c (
    unrealtype const * udata ) [inline]
```

Definition at line 283 of file [DerivationStencils.h](#).

```
00284 { return s12c(udata, 6); }
```

References [s12c\(\)](#).

Here is the call graph for this function:



6.34.2.14 s12c() [2/2]

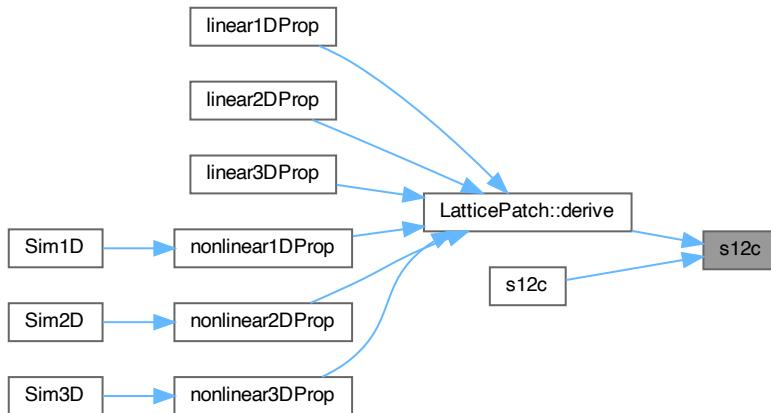
```
sunrealtype s12c (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 206 of file [DerivationStencils.h](#).

```
00206
00207     return 1.0 / 5544.0 * udata[-6 * dPD] - 1.0 / 385.0 * udata[-5 * dPD] +
00208         1.0 / 56.0 * udata[-4 * dPD] - 5.0 / 63.0 * udata[-3 * dPD] +
00209         15.0 / 56.0 * udata[-2 * dPD] - 6.0 / 7.0 * udata[-1 * dPD] + 0 +
00210         6.0 / 7.0 * udata[1 * dPD] - 15.0 / 56.0 * udata[2 * dPD] +
00211         5.0 / 63.0 * udata[3 * dPD] - 1.0 / 56.0 * udata[4 * dPD] +
00212         1.0 / 385.0 * udata[5 * dPD] - 1.0 / 5544.0 * udata[6 * dPD];
00213 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s12c\(\)](#).

Here is the caller graph for this function:



6.34.2.15 s12f() [1/2]

```
sunrealtype s12f (
    unrealtype const * udata ) [inline]
```

Definition at line 281 of file [DerivationStencils.h](#).

```
00282 { return s12f(udata, 6); }
```

References [s12f\(\)](#).

Here is the call graph for this function:



6.34.2.16 s12f() [2/2]

```

sunrealtype s12f (
    unrealtype const * udata,
    const int dPD ) [inline]
  
```

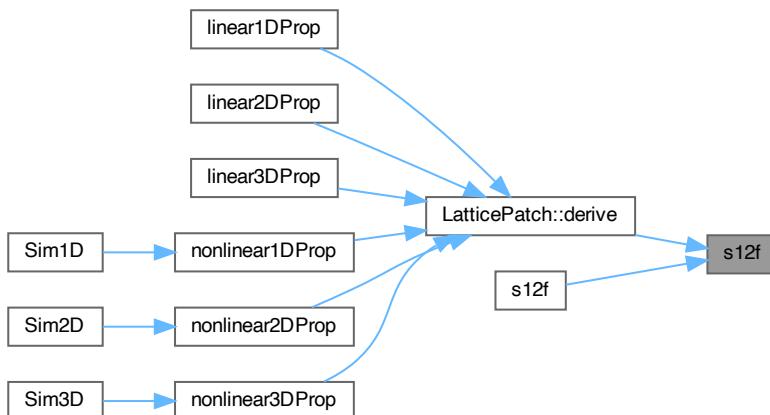
Definition at line 196 of file [DerivationStencils.h](#).

```

00196
00197     return -1.0 / 5544.0 * udata[-7 * dPD] + 1.0 / 396.0 * udata[-6 * dPD] -
00198         1.0 / 60.0 * udata[-5 * dPD] + 5.0 / 72.0 * udata[-4 * dPD] -
00199         5.0 / 24.0 * udata[-3 * dPD] + 1.0 / 2.0 * udata[-2 * dPD] -
00200         7.0 / 6.0 * udata[-1 * dPD] + 13.0 / 42.0 * udata[0] +
00201         5.0 / 8.0 * udata[1 * dPD] - 5.0 / 36.0 * udata[2 * dPD] +
00202         1.0 / 36.0 * udata[3 * dPD] - 1.0 / 264.0 * udata[4 * dPD] +
00203         1.0 / 3960.0 * udata[5 * dPD];
00204 }
  
```

Referenced by [LatticePatch::derive\(\)](#), and [s12f\(\)](#).

Here is the caller graph for this function:



6.34.2.17 s13b() [1/2]

```
sunrealtype s13b (
    unrealtype const * udata ) [inline]
```

Definition at line 289 of file [DerivationStencils.h](#).

```
00290 { return s13b(udata, 6); }
```

References [s13b\(\)](#).

Here is the call graph for this function:



6.34.2.18 s13b() [2/2]

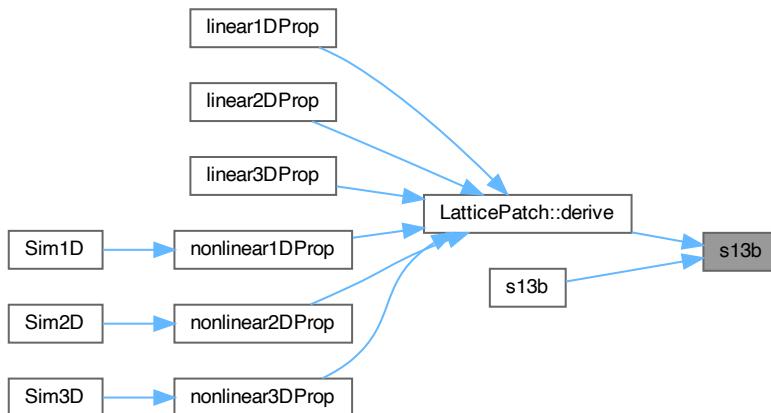
```
sunrealtype s13b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 235 of file [DerivationStencils.h](#).

```
00235 {
00236     return 1.0 / 10296.0 * udata[-6 * dPD] - 1.0 / 660.0 * udata[-5 * dPD] +
00237         1.0 / 88.0 * udata[-4 * dPD] - 1.0 / 18.0 * udata[-3 * dPD] +
00238         5.0 / 24.0 * udata[-2 * dPD] - 3.0 / 4.0 * udata[-1 * dPD] -
00239         1.0 / 7.0 * udata[0] + udata[1 * dPD] - 3.0 / 8.0 * udata[2 * dPD] +
00240         5.0 / 36.0 * udata[3 * dPD] - 1.0 / 24.0 * udata[4 * dPD] +
00241         1.0 / 110.0 * udata[5 * dPD] - 1.0 / 792.0 * udata[6 * dPD] +
00242         1.0 / 12012.0 * udata[7 * dPD];
00243 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s13b\(\)](#).

Here is the caller graph for this function:



6.34.2.19 s13f() [1/2]

```
sunrealtype s13f (
    unrealtype const * udata ) [inline]
```

Definition at line 287 of file [DerivationStencils.h](#).
 00288 { **return** s13f(udata, 6); }

References [s13f\(\)](#).

Here is the call graph for this function:



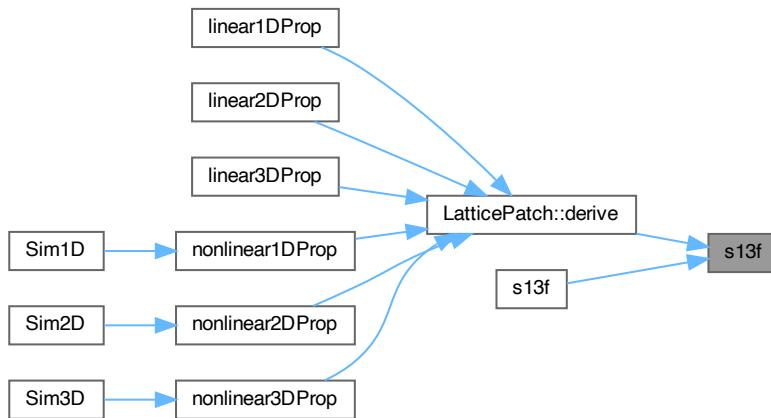
6.34.2.20 s13f() [2/2]

```
sunrealtype s13f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 225 of file [DerivationStencils.h](#).
 00225 {
 00226 **return** -1.0 / 12012.0 * udata[-7 * dPD] + 1.0 / 792.0 * udata[-6 * dPD] -
 00227 1.0 / 110.0 * udata[-5 * dPD] + 1.0 / 24.0 * udata[-4 * dPD] -
 00228 5.0 / 36.0 * udata[-3 * dPD] + 3.0 / 8.0 * udata[-2 * dPD] -
 00229 1.0 / 1.0 * udata[-1 * dPD] + 1.0 / 7.0 * udata[0] +
 00230 3.0 / 4.0 * udata[1 * dPD] - 5.0 / 24.0 * udata[2 * dPD] +
 00231 1.0 / 18.0 * udata[3 * dPD] - 1.0 / 88.0 * udata[4 * dPD] +
 00232 1.0 / 660.0 * udata[5 * dPD] - 1.0 / 10296.0 * udata[6 * dPD];
 00233 }

Referenced by [LatticePatch::derive\(\)](#), and [s13f\(\)](#).

Here is the caller graph for this function:



6.34.2.21 `s1b()` [1/2]

```
sunrealtype slb (
    unrealtype const * udata ) [inline]
```

Definition at line 250 of file [DerivationStencils.h](#).
00250 { **return** slb(udata, 6); }

References [s1b\(\)](#).

Here is the call graph for this function:



6.34.2.22 s1b() [2/2]

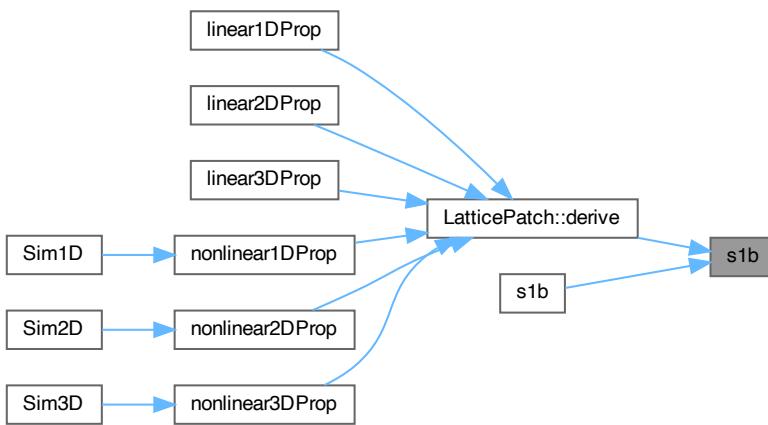
```
sunrealtype s1b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 21 of file [DerivationStencils.h](#).

```
00021
00022     return -1.0 / 1.0 * udata[0] + udata[1 * dPD];
00023 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s1b\(\)](#).

Here is the caller graph for this function:



6.34.2.23 s1f() [1/2]

```
sunrealtype s1f (
    unrealtype const * udata ) [inline]
```

Definition at line 249 of file [DerivationStencils.h](#).

```
00249 { return s1f(udata, 6); }
```

References [s1f\(\)](#).

Here is the call graph for this function:



6.34.2.24 s1f() [2/2]

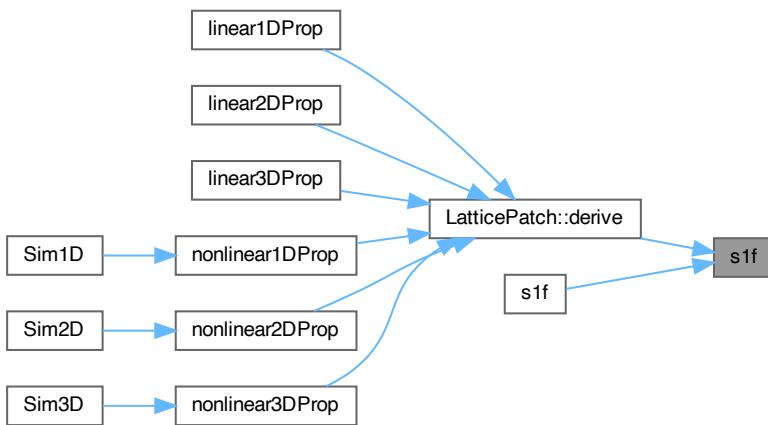
```
sunrealtype s1f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 16 of file [DerivationStencils.h](#).

```
00016     return -1.0 / 1.0 * udata[-1 * dPD] + udata[0];
00017 }
00018 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s1f\(\)](#).

Here is the caller graph for this function:



6.34.2.25 s2b() [1/2]

```
sunrealtype s2b (
    unrealtype const * udata ) [inline]
```

Definition at line 253 of file [DerivationStencils.h](#).

```
00253 { return s2b(udata, 6); }
```

References [s2b\(\)](#).

Here is the call graph for this function:



6.34.2.26 s2b() [2/2]

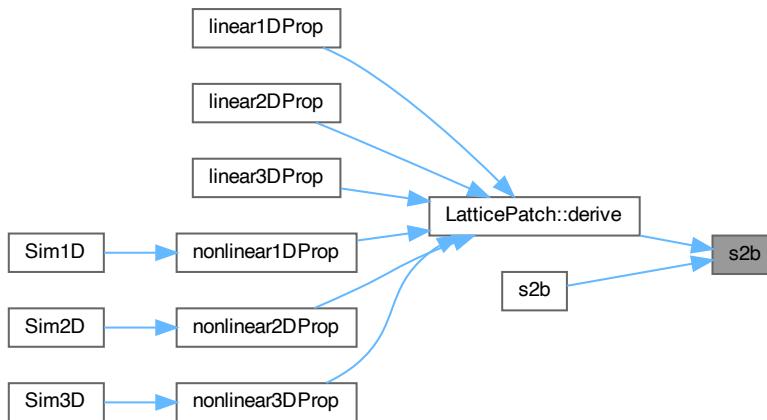
```
sunrealtype s2b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 35 of file [DerivationStencils.h](#).

```
00035 {
00036     return -3.0 / 2.0 * udata[0] + 2.0 / 1.0 * udata[1 * dPD] -
00037         1.0 / 2.0 * udata[2 * dPD];
00038 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s2b\(\)](#).

Here is the caller graph for this function:



6.34.2.27 s2c() [1/2]

```
sunrealtype s2c (
    unrealtype const * udata ) [inline]
```

Definition at line 252 of file [DerivationStencils.h](#).

```
00252 { return s2c(udata, 6); }
```

References [s2c\(\)](#).

Here is the call graph for this function:



6.34.2.28 s2c() [2/2]

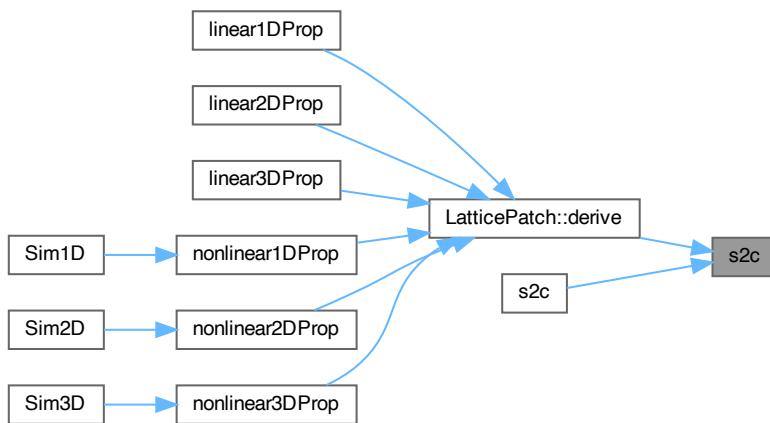
```
sunrealtype s2c (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 31 of file [DerivationStencils.h](#).

```
00031
00032     return -1.0 / 2.0 * udata[-1 * dPD] + 0 + 1.0 / 2.0 * udata[1 * dPD];
00033 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s2c\(\)](#).

Here is the caller graph for this function:



6.34.2.29 s2f() [1/2]

```
sunrealtype s2f (
    unrealtype const * udata ) [inline]
```

Definition at line 251 of file [DerivationStencils.h](#).

```
00251 { return s2f(udata, 6); }
```

References [s2f\(\)](#).

Here is the call graph for this function:



6.34.2.30 s2f() [2/2]

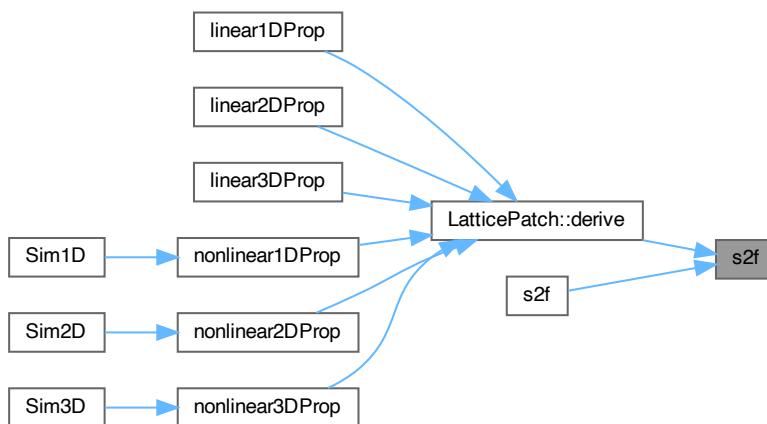
```
sunrealtype s2f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 26 of file [DerivationStencils.h](#).

```
00026
00027     return 1.0 / 2.0 * udata[-2 * dPD] - 2.0 / 1.0 * udata[-1 * dPD] +
00028         3.0 / 2.0 * udata[0];
00029 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s2f\(\)](#).

Here is the caller graph for this function:



6.34.2.31 s3b() [1/2]

```
sunrealtype s3b (
    unrealtype const * udata ) [inline]
```

Definition at line 255 of file [DerivationStencils.h](#).

```
00255 { return s3b(udata, 6); }
```

References [s3b\(\)](#).

Here is the call graph for this function:



6.34.2.32 s3b() [2/2]

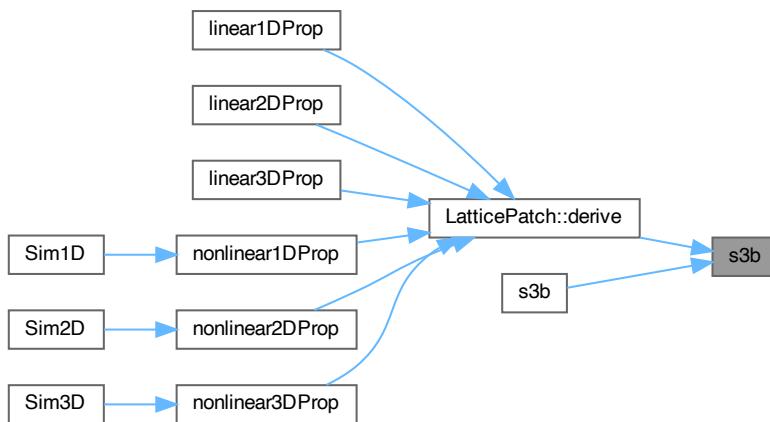
```
sunrealtype s3b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 45 of file [DerivationStencils.h](#).

```
00045 {  
00046     return -1.0 / 3.0 * udata[-1 * dPD] - 1.0 / 2.0 * udata[0] + udata[1 * dPD] -  
00047         1.0 / 6.0 * udata[2 * dPD];  
00048 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s3b\(\)](#).

Here is the caller graph for this function:



6.34.2.33 s3f() [1/2]

```
sunrealtype s3f (
    unrealtype const * udata ) [inline]
```

Definition at line 254 of file [DerivationStencils.h](#).

```
00254 { return s3f(udata, 6); }
```

References [s3f\(\)](#).

Here is the call graph for this function:



6.34.2.34 s3f() [2/2]

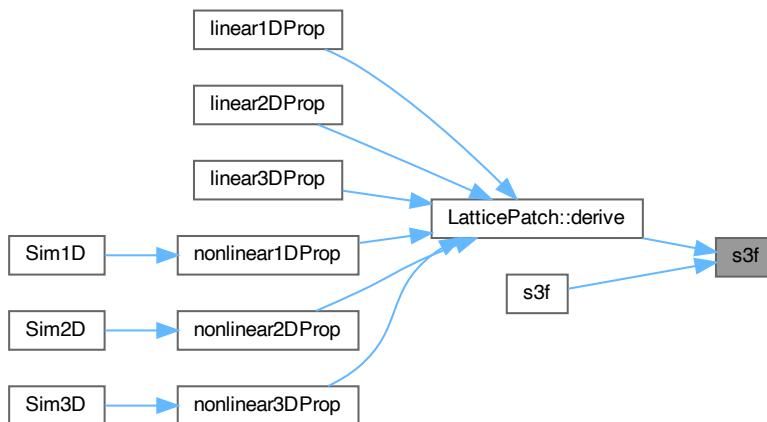
```
sunrealtype s3f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 40 of file [DerivationStencils.h](#).

```
00040
00041     return 1.0 / 6.0 * udata[-2 * dPD] - 1.0 / 1.0 * udata[-1 * dPD] +
00042         1.0 / 2.0 * udata[0] + 1.0 / 3.0 * udata[1 * dPD];
00043 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s3f\(\)](#).

Here is the caller graph for this function:



6.34.2.35 s4b() [1/2]

```
sunrealtype s4b (
    unrealtype const * udata ) [inline]
```

Definition at line 258 of file [DerivationStencils.h](#).

```
00258 { return s4b(udata, 6); }
```

References [s4b\(\)](#).

Here is the call graph for this function:



6.34.2.36 s4b() [2/2]

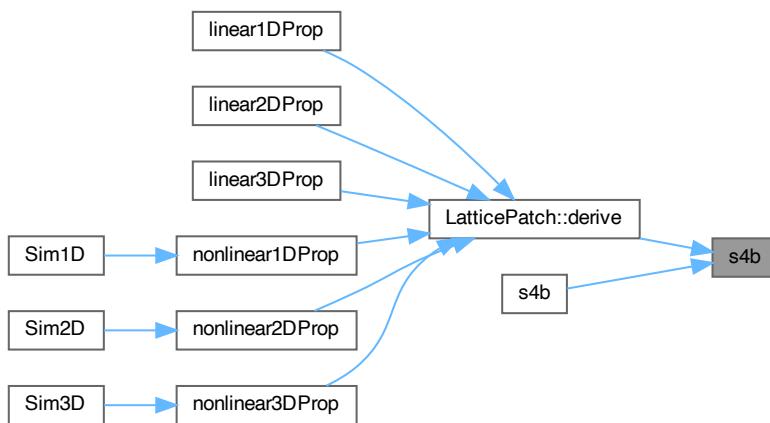
```
sunrealtype s4b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 61 of file [DerivationStencils.h](#).

```
00061
00062     return -1.0 / 4.0 * udata[-1 * dPD] - 5.0 / 6.0 * udata[0] +
00063         3.0 / 2.0 * udata[1 * dPD] - 1.0 / 2.0 * udata[2 * dPD] +
00064         1.0 / 12.0 * udata[3 * dPD];
00065 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s4b\(\)](#).

Here is the caller graph for this function:



6.34.2.37 s4c() [1/2]

```
sunrealtype s4c (
    unrealtype const * udata ) [inline]
```

Definition at line 257 of file [DerivationStencils.h](#).

```
00257 { return s4c(udata, 6); }
```

References [s4c\(\)](#).

Here is the call graph for this function:



6.34.2.38 s4c() [2/2]

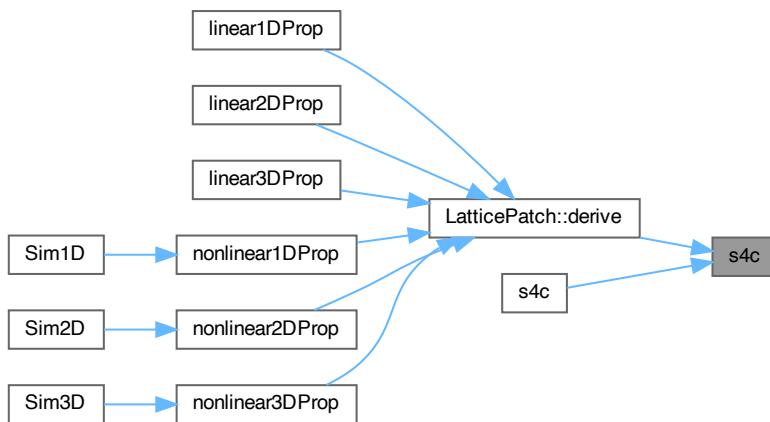
```
sunrealtype s4c (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 56 of file [DerivationStencils.h](#).

```
00056
00057     return 1.0 / 12.0 * udata[-2 * dPD] - 2.0 / 3.0 * udata[-1 * dPD] + 0 +
00058         2.0 / 3.0 * udata[1 * dPD] - 1.0 / 12.0 * udata[2 * dPD];
00059 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s4c\(\)](#).

Here is the caller graph for this function:



6.34.2.39 s4f() [1/2]

```
sunrealtype s4f (
    unrealtype const * udata ) [inline]
```

Definition at line 256 of file [DerivationStencils.h](#).

```
00256 { return s4f(udata, 6); }
```

References [s4f\(\)](#).

Here is the call graph for this function:



6.34.2.40 s4f() [2/2]

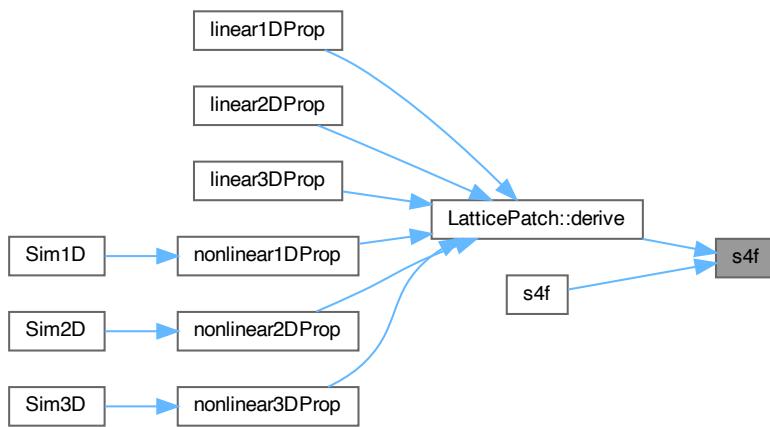
```
sunrealtype s4f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 50 of file [DerivationStencils.h](#).

```
00050
00051     return -1.0 / 12.0 * udata[-3 * dPD] + 1.0 / 2.0 * udata[-2 * dPD] -
00052         3.0 / 2.0 * udata[-1 * dPD] + 5.0 / 6.0 * udata[0] +
00053         1.0 / 4.0 * udata[1 * dPD];
00054 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s4f\(\)](#).

Here is the caller graph for this function:



6.34.2.41 s5b() [1/2]

```
sunrealtype s5b (
    unrealtype const * udata ) [inline]
```

Definition at line 260 of file [DerivationStencils.h](#).

```
00260 { return s5b(udata, 6); }
```

References [s5b\(\)](#).

Here is the call graph for this function:



6.34.2.42 s5b() [2/2]

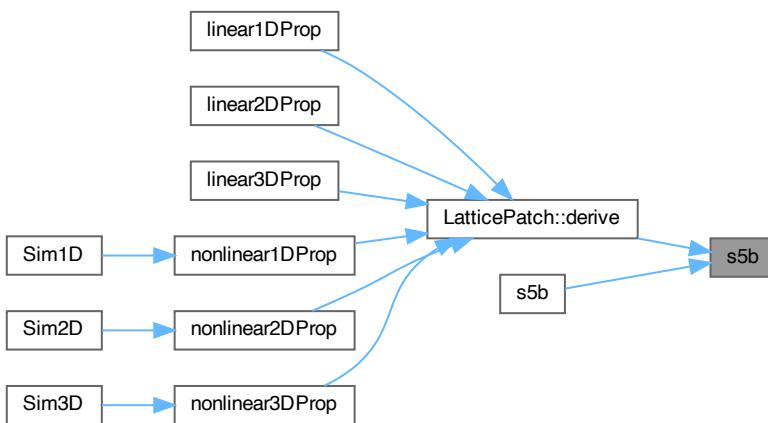
```
sunrealtype s5b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 73 of file [DerivationStencils.h](#).

```
00073
00074     return 1.0 / 20.0 * udata[-2 * dPD] - 1.0 / 2.0 * udata[-1 * dPD] -
00075         1.0 / 3.0 * udata[0] + udata[1 * dPD] - 1.0 / 4.0 * udata[2 * dPD] +
00076         1.0 / 30.0 * udata[3 * dPD];
00077 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s5b\(\)](#).

Here is the caller graph for this function:



6.34.2.43 s5f() [1/2]

```
sunrealtype s5f (
    unrealtype const * udata ) [inline]
```

Definition at line 259 of file [DerivationStencils.h](#).

```
00259 { return s5f(udata, 6); }
```

References [s5f\(\)](#).

Here is the call graph for this function:



6.34.2.44 s5f() [2/2]

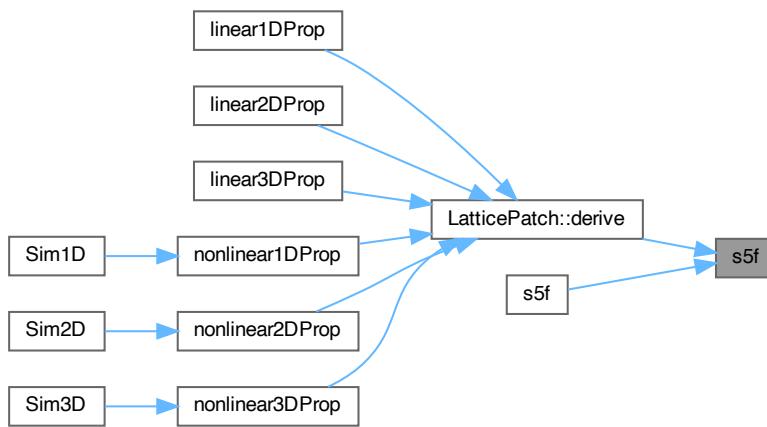
```
sunrealtype s5f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 67 of file DerivationStencils.h.

```
00067
00068     return -1.0 / 30.0 * udata[-3 * dPD] + 1.0 / 4.0 * udata[-2 * dPD] -
00069         1.0 / 1.0 * udata[-1 * dPD] + 1.0 / 3.0 * udata[0] +
00070         1.0 / 2.0 * udata[1 * dPD] - 1.0 / 20.0 * udata[2 * dPD];
00071 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s5f\(\)](#).

Here is the caller graph for this function:



6.34.2.45 s6b() [1/2]

```
sunrealtype s6b (
    unrealtype const * udata ) [inline]
```

Definition at line 263 of file DerivationStencils.h.

```
00263 { return s6b(udata, 6); }
```

References [s6b\(\)](#).

Here is the call graph for this function:



6.34.2.46 s6b() [2/2]

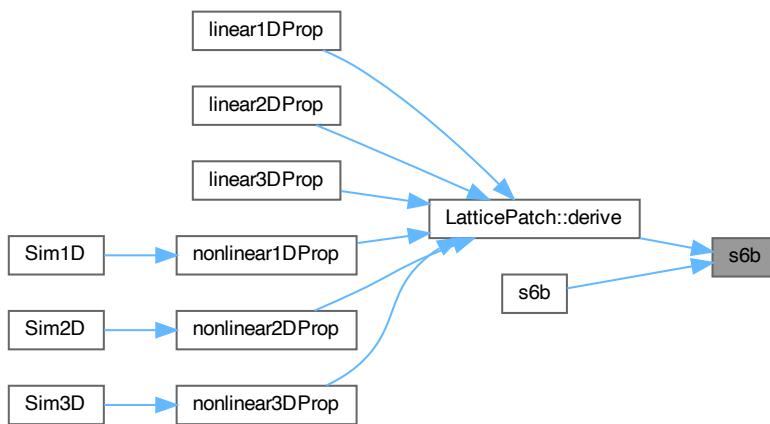
```
sunrealtype s6b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 92 of file [DerivationStencils.h](#).

```
00092
00093     return 1.0 / 30.0 * udata[-2 * dPD] - 2.0 / 5.0 * udata[-1 * dPD] -
00094         7.0 / 12.0 * udata[0] + 4.0 / 3.0 * udata[1 * dPD] -
00095         1.0 / 2.0 * udata[2 * dPD] + 2.0 / 15.0 * udata[3 * dPD] -
00096         1.0 / 60.0 * udata[4 * dPD];
00097 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s6b\(\)](#).

Here is the caller graph for this function:



6.34.2.47 s6c() [1/2]

```
sunrealtype s6c (
    unrealtype const * udata ) [inline]
```

Definition at line 262 of file [DerivationStencils.h](#).

```
00262 { return s6c(udata, 6); }
```

References [s6c\(\)](#).

Here is the call graph for this function:



6.34.2.48 s6c() [2/2]

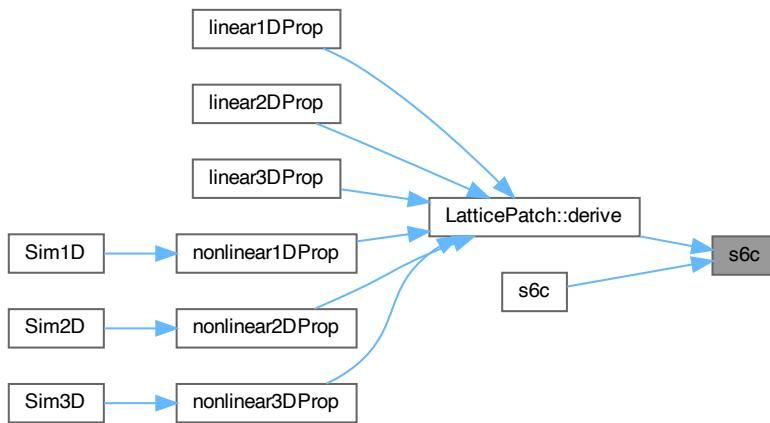
```
sunrealtype s6c (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 86 of file [DerivationStencils.h](#).

```
00086
00087     return -1.0 / 60.0 * udata[-3 * dPD] + 3.0 / 20.0 * udata[-2 * dPD] -
00088         3.0 / 4.0 * udata[-1 * dPD] + 0 + 3.0 / 4.0 * udata[1 * dPD] -
00089         3.0 / 20.0 * udata[2 * dPD] + 1.0 / 60.0 * udata[3 * dPD];
00090 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s6c\(\)](#).

Here is the caller graph for this function:



6.34.2.49 s6f() [1/2]

```
sunrealtype s6f (
    unrealtype const * udata ) [inline]
```

Definition at line 261 of file [DerivationStencils.h](#).

```
00261 { return s6f(udata, 6); }
```

References [s6f\(\)](#).

Here is the call graph for this function:



6.34.2.50 s6f() [2/2]

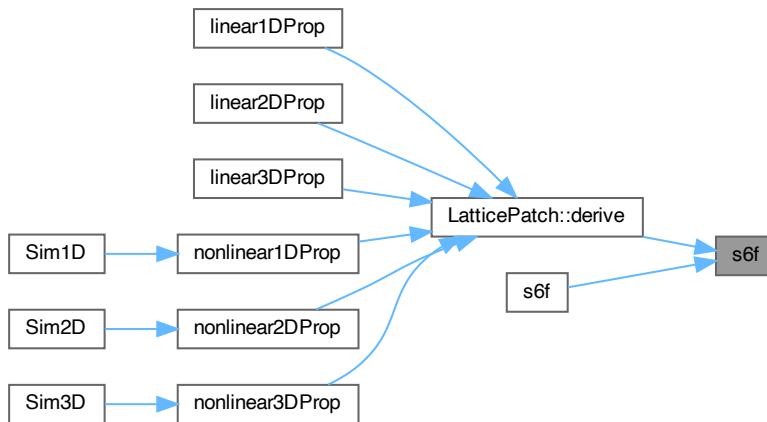
```
sunrealtype s6f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 79 of file [DerivationStencils.h](#).

```
00079
00080     return 1.0 / 60.0 * udata[-4 * dPD] - 2.0 / 15.0 * udata[-3 * dPD] +
00081         1.0 / 2.0 * udata[-2 * dPD] - 4.0 / 3.0 * udata[-1 * dPD] +
00082         7.0 / 12.0 * udata[0] + 2.0 / 5.0 * udata[1 * dPD] -
00083         1.0 / 30.0 * udata[2 * dPD];
00084 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s6f\(\)](#).

Here is the caller graph for this function:



6.34.2.51 s7b() [1/2]

```
sunrealtype s7b (
    unrealtype const * udata ) [inline]
```

Definition at line 265 of file [DerivationStencils.h](#).

```
00265 { return s7b(udata, 6); }
```

References [s7b\(\)](#).

Here is the call graph for this function:



6.34.2.52 s7b() [2/2]

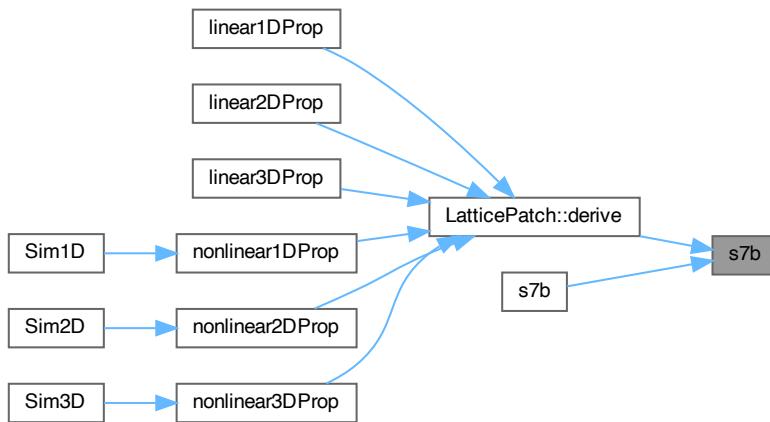
```
sunrealtype s7b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 106 of file [DerivationStencils.h](#).

```
00106      {
00107      return -1.0 / 105.0 * udata[-3 * dPD] + 1.0 / 10.0 * udata[-2 * dPD] -
00108          3.0 / 5.0 * udata[-1 * dPD] - 1.0 / 4.0 * udata[0] + udata[1 * dPD] -
00109          3.0 / 10.0 * udata[2 * dPD] + 1.0 / 15.0 * udata[3 * dPD] -
00110          1.0 / 140.0 * udata[4 * dPD];
00111 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s7b\(\)](#).

Here is the caller graph for this function:



6.34.2.53 s7f() [1/2]

```
sunrealtype s7f (
    unrealtype const * udata ) [inline]
```

Definition at line 264 of file [DerivationStencils.h](#).

```
00264 { return s7f(udata, 6); }
```

References [s7f\(\)](#).

Here is the call graph for this function:



6.34.2.54 s7f() [2/2]

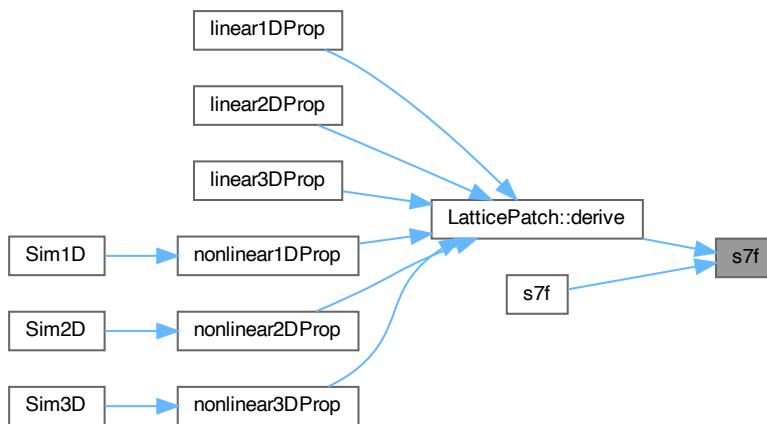
```
sunrealtype s7f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 99 of file [DerivationStencils.h](#).

```
00099
00100     return 1.0 / 140.0 * udata[-4 * dPD] - 1.0 / 15.0 * udata[-3 * dPD] +
00101         3.0 / 10.0 * udata[-2 * dPD] - 1.0 / 1.0 * udata[-1 * dPD] +
00102         1.0 / 4.0 * udata[0] + 3.0 / 5.0 * udata[1 * dPD] -
00103         1.0 / 10.0 * udata[2 * dPD] + 1.0 / 105.0 * udata[3 * dPD];
00104 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s7f\(\)](#).

Here is the caller graph for this function:



6.34.2.55 s8b() [1/2]

```
sunrealtype s8b (
    unrealtype const * udata ) [inline]
```

Definition at line 268 of file [DerivationStencils.h](#).

```
00268 { return s8b(udata, 6); }
```

References [s8b\(\)](#).

Here is the call graph for this function:



6.34.2.56 s8b() [2/2]

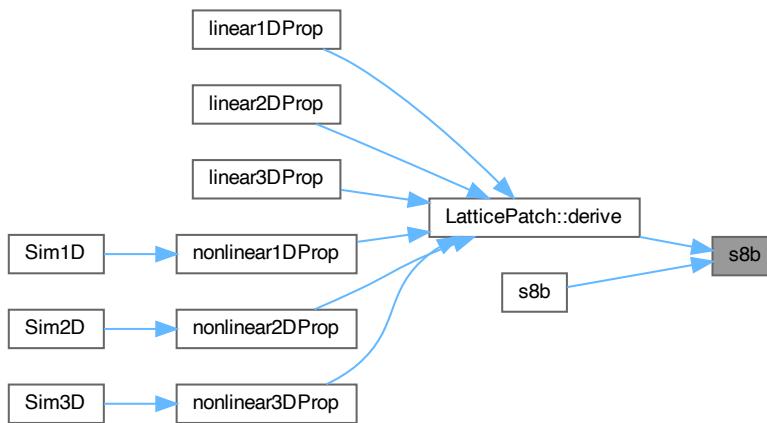
```
sunrealtype s8b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 128 of file [DerivationStencils.h](#).

```
00128     {
00129     return -1.0 / 168.0 * udata[-3 * dPD] + 1.0 / 14.0 * udata[-2 * dPD] -
00130         1.0 / 2.0 * udata[-1 * dPD] - 9.0 / 20.0 * udata[0] +
00131         5.0 / 4.0 * udata[1 * dPD] - 1.0 / 2.0 * udata[2 * dPD] +
00132         1.0 / 6.0 * udata[3 * dPD] - 1.0 / 28.0 * udata[4 * dPD] +
00133         1.0 / 280.0 * udata[5 * dPD];
00134 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s8b\(\)](#).

Here is the caller graph for this function:



6.34.2.57 s8c() [1/2]

```
sunrealtype s8c (
    unrealtype const * udata ) [inline]
```

Definition at line 267 of file [DerivationStencils.h](#).

```
00267 { return s8c(udata, 6); }
```

References [s8c\(\)](#).

Here is the call graph for this function:



6.34.2.58 s8c() [2/2]

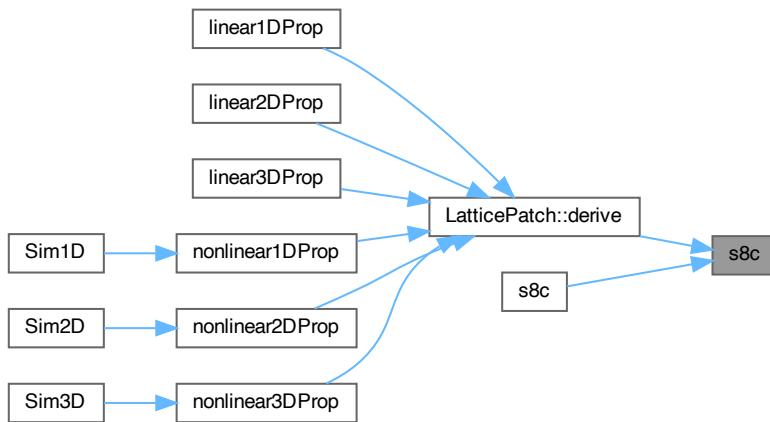
```
sunrealtype s8c (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 121 of file [DerivationStencils.h](#).

```
00121
00122     return 1.0 / 280.0 * udata[-4 * dPD] - 4.0 / 105.0 * udata[-3 * dPD] +
00123         1.0 / 5.0 * udata[-2 * dPD] - 4.0 / 5.0 * udata[-1 * dPD] + 0 +
00124         4.0 / 5.0 * udata[1 * dPD] - 1.0 / 5.0 * udata[2 * dPD] +
00125         4.0 / 105.0 * udata[3 * dPD] - 1.0 / 280.0 * udata[4 * dPD];
00126 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s8c\(\)](#).

Here is the caller graph for this function:



6.34.2.59 s8f() [1/2]

```
sunrealtype s8f (
    unrealtype const * udata ) [inline]
```

Definition at line 266 of file [DerivationStencils.h](#).

```
00266 { return s8f(udata, 6); }
```

References [s8f\(\)](#).

Here is the call graph for this function:



6.34.2.60 s8f() [2/2]

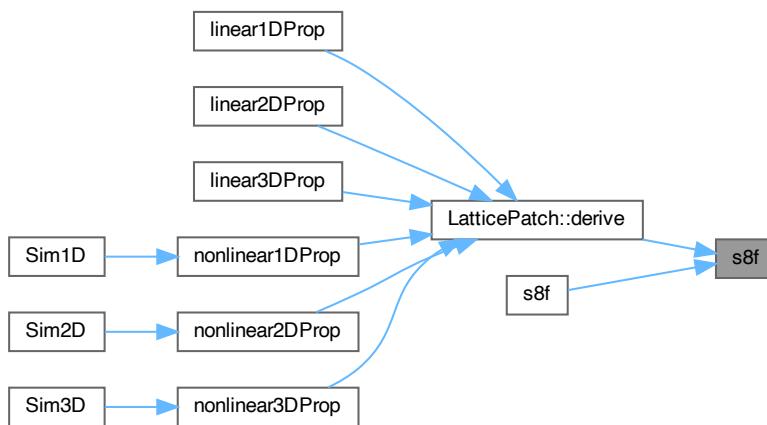
```
sunrealtype s8f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 113 of file [DerivationStencils.h](#).

```
00113
00114     return -1.0 / 280.0 * udata[-5 * dPD] + 1.0 / 28.0 * udata[-4 * dPD] -
00115         1.0 / 6.0 * udata[-3 * dPD] + 1.0 / 2.0 * udata[-2 * dPD] -
00116         5.0 / 4.0 * udata[-1 * dPD] + 9.0 / 20.0 * udata[0] +
00117         1.0 / 2.0 * udata[1 * dPD] - 1.0 / 14.0 * udata[2 * dPD] +
00118         1.0 / 168.0 * udata[3 * dPD];
00119 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s8f\(\)](#).

Here is the caller graph for this function:



6.34.2.61 s9b() [1/2]

```
sunrealtype s9b (
    unrealtype const * udata ) [inline]
```

Definition at line 270 of file [DerivationStencils.h](#).

```
00270 { return s9b(udata, 6); }
```

References [s9b\(\)](#).

Here is the call graph for this function:



6.34.2.62 s9b() [2/2]

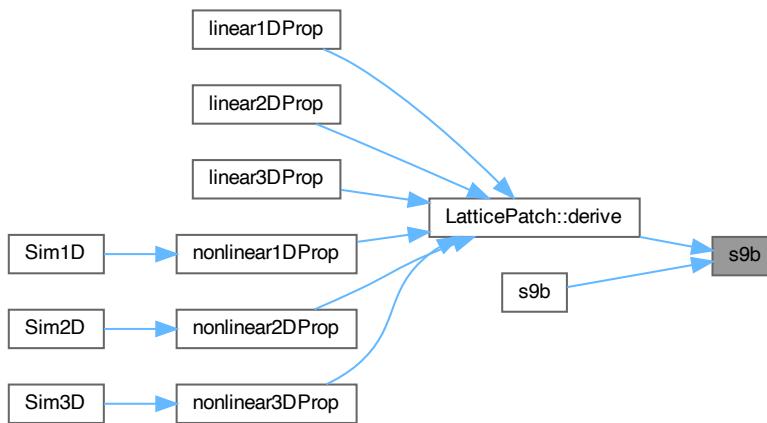
```
sunrealtype s9b (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 144 of file [DerivationStencils.h](#).

```
00144
00145     return 1.0 / 504.0 * udata[-4 * dPD] - 1.0 / 42.0 * udata[-3 * dPD] +
00146         1.0 / 7.0 * udata[-2 * dPD] - 2.0 / 3.0 * udata[-1 * dPD] -
00147         1.0 / 5.0 * udata[0] + udata[1 * dPD] - 1.0 / 3.0 * udata[2 * dPD] +
00148         2.0 / 21.0 * udata[3 * dPD] - 1.0 / 56.0 * udata[4 * dPD] +
00149         1.0 / 630.0 * udata[5 * dPD];
00150 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s9b\(\)](#).

Here is the caller graph for this function:



6.34.2.63 s9f() [1/2]

```
sunrealtype s9f (
    unrealtype const * udata ) [inline]
```

Definition at line 269 of file [DerivationStencils.h](#).

```
00269 { return s9f(udata, 6); }
```

References [s9f\(\)](#).

Here is the call graph for this function:



6.34.2.64 s9f() [2/2]

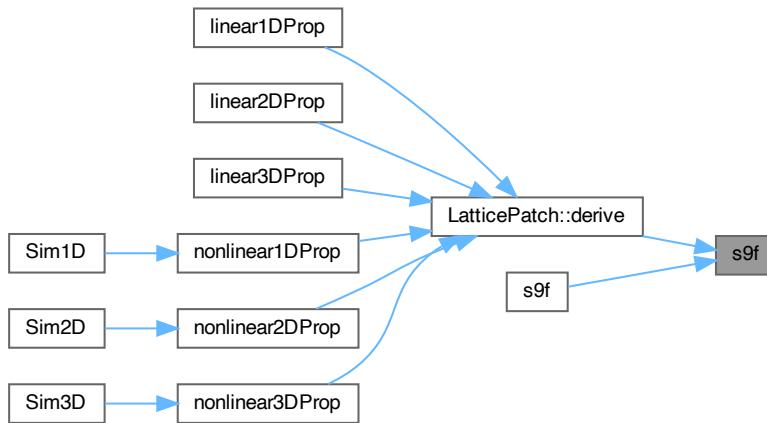
```
sunrealtype s9f (
    unrealtype const * udata,
    const int dPD ) [inline]
```

Definition at line 136 of file [DerivationStencils.h](#).

```
00136
00137     return -1.0 / 630.0 * udata[-5 * dPD] + 1.0 / 56.0 * udata[-4 * dPD] -
00138         2.0 / 21.0 * udata[-3 * dPD] + 1.0 / 3.0 * udata[-2 * dPD] -
00139         1.0 / 1.0 * udata[-1 * dPD] + 1.0 / 5.0 * udata[0] +
00140         2.0 / 3.0 * udata[1 * dPD] - 1.0 / 7.0 * udata[2 * dPD] +
00141         1.0 / 42.0 * udata[3 * dPD] - 1.0 / 504.0 * udata[4 * dPD];
00142 }
```

Referenced by [LatticePatch::derive\(\)](#), and [s9f\(\)](#).

Here is the caller graph for this function:



6.35 DerivationStencils.h

[Go to the documentation of this file.](#)

```
00001 ///////////////////////////////////////////////////////////////////
00002 /// @file DerivationStencils.h
00003 /// @brief Definition of derivation stencils from order 1 to 13
00004 ///////////////////////////////////////////////////////////////////
00005
00006 #pragma once
00007
00008 #include <sundials/sundials_types.h> /* definition of type unrealtype */
00009
00010 ///////////////////////////////////////////////////////////////////
00011 // Stencils with variable dPD -- data point dimension //
00012 ///////////////////////////////////////////////////////////////////
00013
00014 // Downwind (forward) differentiating
00015 #pragma omp declare simd uniform(dPD) notinbranch
00016 inline unrealtype slf(unrealtype const *udata, const int dPD) {
00017     return -1.0 / 1.0 * udata[-1 * dPD] + udata[0];
00018 }
00019 // Upwind (backward) differentiating
00020 #pragma omp declare simd uniform(dPD) notinbranch
00021 inline unrealtype slb(unrealtype const *udata, const int dPD) {
00022     return -1.0 / 1.0 * udata[0] + udata[1 * dPD];
00023 }
```

```

00024
00025 #pragma omp declare simd uniform(dPD) notinbranch
00026 inline sunrealtype s2f(sunrealtype const *udata, const int dPD) {
00027     return 1.0 / 2.0 * udata[-2 * dPD] - 2.0 / 1.0 * udata[-1 * dPD] +
00028         3.0 / 2.0 * udata[0];
00029 }
00030 #pragma omp declare simd uniform(dPD) notinbranch
00031 inline sunrealtype s2c(sunrealtype const *udata, const int dPD) {
00032     return -1.0 / 2.0 * udata[-1 * dPD] + 0 + 1.0 / 2.0 * udata[1 * dPD];
00033 }
00034 #pragma omp declare simd uniform(dPD) notinbranch
00035 inline sunrealtype s2b(sunrealtype const *udata, const int dPD) {
00036     return -3.0 / 2.0 * udata[0] + 2.0 / 1.0 * udata[1 * dPD] -
00037         1.0 / 2.0 * udata[2 * dPD];
00038 }
00039 #pragma omp declare simd uniform(dPD) notinbranch
00040 inline sunrealtype s3f(sunrealtype const *udata, const int dPD) {
00041     return 1.0 / 6.0 * udata[-2 * dPD] - 1.0 / 1.0 * udata[-1 * dPD] +
00042         1.0 / 2.0 * udata[0] + 1.0 / 3.0 * udata[1 * dPD];
00043 }
00044 #pragma omp declare simd uniform(dPD) notinbranch
00045 inline sunrealtype s3b(sunrealtype const *udata, const int dPD) {
00046     return -1.0 / 3.0 * udata[-1 * dPD] - 1.0 / 2.0 * udata[0] + udata[1 * dPD] -
00047         1.0 / 6.0 * udata[2 * dPD];
00048 }
00049 #pragma omp declare simd uniform(dPD) notinbranch
00050 inline sunrealtype s4f(sunrealtype const *udata, const int dPD) {
00051     return -1.0 / 12.0 * udata[-3 * dPD] + 1.0 / 2.0 * udata[-2 * dPD] -
00052         3.0 / 2.0 * udata[-1 * dPD] + 5.0 / 6.0 * udata[0] +
00053         1.0 / 4.0 * udata[1 * dPD];
00054 }
00055 #pragma omp declare simd uniform(dPD) notinbranch
00056 inline sunrealtype s4c(sunrealtype const *udata, const int dPD) {
00057     return 1.0 / 12.0 * udata[-2 * dPD] - 2.0 / 3.0 * udata[-1 * dPD] + 0 +
00058         2.0 / 3.0 * udata[1 * dPD] - 1.0 / 12.0 * udata[2 * dPD];
00059 }
00060 #pragma omp declare simd uniform(dPD) notinbranch
00061 inline sunrealtype s4b(sunrealtype const *udata, const int dPD) {
00062     return -1.0 / 4.0 * udata[-1 * dPD] - 5.0 / 6.0 * udata[0] +
00063         3.0 / 2.0 * udata[1 * dPD] - 1.0 / 2.0 * udata[2 * dPD] +
00064         1.0 / 12.0 * udata[3 * dPD];
00065 }
00066 #pragma omp declare simd uniform(dPD) notinbranch
00067 inline sunrealtype s5f(sunrealtype const *udata, const int dPD) {
00068     return -1.0 / 30.0 * udata[-3 * dPD] + 1.0 / 4.0 * udata[-2 * dPD] -
00069         1.0 / 1.0 * udata[-1 * dPD] + 1.0 / 3.0 * udata[0] +
00070         1.0 / 2.0 * udata[1 * dPD] - 1.0 / 20.0 * udata[2 * dPD];
00071 }
00072 #pragma omp declare simd uniform(dPD) notinbranch
00073 inline sunrealtype s5b(sunrealtype const *udata, const int dPD) {
00074     return 1.0 / 20.0 * udata[-2 * dPD] - 1.0 / 2.0 * udata[-1 * dPD] -
00075         1.0 / 3.0 * udata[0] + udata[1 * dPD] - 1.0 / 4.0 * udata[2 * dPD] +
00076         1.0 / 30.0 * udata[3 * dPD];
00077 }
00078 #pragma omp declare simd uniform(dPD) notinbranch
00079 inline sunrealtype s6f(sunrealtype const *udata, const int dPD) {
00080     return 1.0 / 60.0 * udata[-4 * dPD] - 2.0 / 15.0 * udata[-3 * dPD] +
00081         1.0 / 2.0 * udata[-2 * dPD] - 4.0 / 3.0 * udata[-1 * dPD] +
00082         7.0 / 12.0 * udata[0] + 2.0 / 5.0 * udata[1 * dPD] -
00083         1.0 / 30.0 * udata[2 * dPD];
00084 }
00085 #pragma omp declare simd uniform(dPD) notinbranch
00086 inline sunrealtype s6c(sunrealtype const *udata, const int dPD) {
00087     return -1.0 / 60.0 * udata[-3 * dPD] + 3.0 / 20.0 * udata[-2 * dPD] -
00088         3.0 / 4.0 * udata[-1 * dPD] + 0 + 3.0 / 4.0 * udata[1 * dPD] -
00089         3.0 / 20.0 * udata[2 * dPD] + 1.0 / 60.0 * udata[3 * dPD];
00090 }
00091 #pragma omp declare simd uniform(dPD) notinbranch
00092 inline sunrealtype s6b(sunrealtype const *udata, const int dPD) {
00093     return 1.0 / 30.0 * udata[-2 * dPD] - 2.0 / 5.0 * udata[-1 * dPD] -
00094         7.0 / 12.0 * udata[0] + 4.0 / 3.0 * udata[1 * dPD] -
00095         1.0 / 2.0 * udata[2 * dPD] + 2.0 / 15.0 * udata[3 * dPD] -
00096         1.0 / 60.0 * udata[4 * dPD];
00097 }
00098 #pragma omp declare simd uniform(dPD) notinbranch
00099 inline sunrealtype s7f(sunrealtype const *udata, const int dPD) {
00100     return 1.0 / 140.0 * udata[-4 * dPD] - 1.0 / 15.0 * udata[-3 * dPD] +
00101         3.0 / 10.0 * udata[-2 * dPD] - 1.0 / 1.0 * udata[-1 * dPD] +
00102         1.0 / 4.0 * udata[0] + 3.0 / 5.0 * udata[1 * dPD] -
00103         1.0 / 10.0 * udata[2 * dPD] + 1.0 / 105.0 * udata[3 * dPD];
00104 }
00105 #pragma omp declare simd uniform(dPD) notinbranch
00106 inline sunrealtype s7b(sunrealtype const *udata, const int dPD) {
00107     return -1.0 / 105.0 * udata[-3 * dPD] + 1.0 / 10.0 * udata[-2 * dPD] -
00108         3.0 / 5.0 * udata[-1 * dPD] - 1.0 / 4.0 * udata[0] + udata[1 * dPD] -
00109         3.0 / 10.0 * udata[2 * dPD] + 1.0 / 15.0 * udata[3 * dPD] -
00110         1.0 / 140.0 * udata[4 * dPD];

```

```

00111 }
00112 #pragma omp declare simd uniform(dPD) notinbranch
00113 inline sunrealtype s8f(sunrealtype const *udata, const int dPD) {
00114     return -1.0 / 280.0 * udata[-5 * dPD] + 1.0 / 28.0 * udata[-4 * dPD] -
00115         1.0 / 6.0 * udata[-3 * dPD] + 1.0 / 2.0 * udata[-2 * dPD] -
00116         5.0 / 4.0 * udata[-1 * dPD] + 9.0 / 20.0 * udata[0] +
00117         1.0 / 2.0 * udata[1 * dPD] - 1.0 / 14.0 * udata[2 * dPD] +
00118         1.0 / 168.0 * udata[3 * dPD];
00119 }
00120 #pragma omp declare simd uniform(dPD) notinbranch
00121 inline sunrealtype s8c(sunrealtype const *udata, const int dPD) {
00122     return 1.0 / 280.0 * udata[-4 * dPD] - 4.0 / 105.0 * udata[-3 * dPD] +
00123         1.0 / 5.0 * udata[-2 * dPD] - 4.0 / 5.0 * udata[-1 * dPD] + 0 +
00124         4.0 / 5.0 * udata[1 * dPD] - 1.0 / 5.0 * udata[2 * dPD] +
00125         4.0 / 105.0 * udata[3 * dPD] - 1.0 / 280.0 * udata[4 * dPD];
00126 }
00127 #pragma omp declare simd uniform(dPD) notinbranch
00128 inline sunrealtype s8b(sunrealtype const *udata, const int dPD) {
00129     return -1.0 / 168.0 * udata[-3 * dPD] + 1.0 / 14.0 * udata[-2 * dPD] -
00130         1.0 / 2.0 * udata[-1 * dPD] - 9.0 / 20.0 * udata[0] +
00131         5.0 / 4.0 * udata[1 * dPD] - 1.0 / 2.0 * udata[2 * dPD] +
00132         1.0 / 6.0 * udata[3 * dPD] - 1.0 / 28.0 * udata[4 * dPD] +
00133         1.0 / 280.0 * udata[5 * dPD];
00134 }
00135 #pragma omp declare simd uniform(dPD) notinbranch
00136 inline sunrealtype s9f(sunrealtype const *udata, const int dPD) {
00137     return -1.0 / 630.0 * udata[-5 * dPD] + 1.0 / 56.0 * udata[-4 * dPD] -
00138         2.0 / 21.0 * udata[-3 * dPD] + 1.0 / 3.0 * udata[-2 * dPD] -
00139         1.0 / 1.0 * udata[-1 * dPD] + 1.0 / 5.0 * udata[0] +
00140         2.0 / 3.0 * udata[1 * dPD] - 1.0 / 7.0 * udata[2 * dPD] +
00141         1.0 / 42.0 * udata[3 * dPD] - 1.0 / 504.0 * udata[4 * dPD];
00142 }
00143 #pragma omp declare simd uniform(dPD) notinbranch
00144 inline sunrealtype s9b(sunrealtype const *udata, const int dPD) {
00145     return 1.0 / 504.0 * udata[-4 * dPD] - 1.0 / 42.0 * udata[-3 * dPD] +
00146         1.0 / 7.0 * udata[-2 * dPD] - 2.0 / 3.0 * udata[-1 * dPD] -
00147         1.0 / 5.0 * udata[0] + udata[1 * dPD] - 1.0 / 3.0 * udata[2 * dPD] +
00148         2.0 / 21.0 * udata[3 * dPD] - 1.0 / 56.0 * udata[4 * dPD] +
00149         1.0 / 630.0 * udata[5 * dPD];
00150 }
00151 #pragma omp declare simd uniform(dPD) notinbranch
00152 inline sunrealtype s10f(sunrealtype const *udata, const int dPD) {
00153     return 1.0 / 1260.0 * udata[-6 * dPD] - 1.0 / 105.0 * udata[-5 * dPD] +
00154         3.0 / 56.0 * udata[-4 * dPD] - 4.0 / 21.0 * udata[-3 * dPD] +
00155         1.0 / 2.0 * udata[-2 * dPD] - 6.0 / 5.0 * udata[-1 * dPD] +
00156         11.0 / 30.0 * udata[0] + 4.0 / 7.0 * udata[1 * dPD] -
00157         3.0 / 28.0 * udata[2 * dPD] + 1.0 / 63.0 * udata[3 * dPD] -
00158         1.0 / 840.0 * udata[4 * dPD];
00159 }
00160 #pragma omp declare simd uniform(dPD) notinbranch
00161 inline sunrealtype s10c(sunrealtype const *udata, const int dPD) {
00162     return -1.0 / 1260.0 * udata[-5 * dPD] + 5.0 / 504.0 * udata[-4 * dPD] -
00163         5.0 / 84.0 * udata[-3 * dPD] + 5.0 / 21.0 * udata[-2 * dPD] -
00164         5.0 / 6.0 * udata[-1 * dPD] + 0 + 5.0 / 6.0 * udata[1 * dPD] -
00165         5.0 / 21.0 * udata[2 * dPD] + 5.0 / 84.0 * udata[3 * dPD] -
00166         5.0 / 504.0 * udata[4 * dPD] + 1.0 / 1260.0 * udata[5 * dPD];
00167 }
00168 #pragma omp declare simd uniform(dPD) notinbranch
00169 inline sunrealtype s10b(sunrealtype const *udata, const int dPD) {
00170     return 1.0 / 840.0 * udata[-4 * dPD] - 1.0 / 63.0 * udata[-3 * dPD] +
00171         3.0 / 28.0 * udata[-2 * dPD] - 4.0 / 7.0 * udata[-1 * dPD] -
00172         11.0 / 30.0 * udata[0] + 6.0 / 5.0 * udata[1 * dPD] -
00173         1.0 / 2.0 * udata[2 * dPD] + 4.0 / 21.0 * udata[3 * dPD] -
00174         3.0 / 56.0 * udata[4 * dPD] + 1.0 / 105.0 * udata[5 * dPD] -
00175         1.0 / 1260.0 * udata[6 * dPD];
00176 }
00177 #pragma omp declare simd uniform(dPD) notinbranch
00178 inline sunrealtype s11f(sunrealtype const *udata, const int dPD) {
00179     return 1.0 / 2772.0 * udata[-6 * dPD] - 1.0 / 210.0 * udata[-5 * dPD] +
00180         5.0 / 168.0 * udata[-4 * dPD] - 5.0 / 42.0 * udata[-3 * dPD] +
00181         5.0 / 14.0 * udata[-2 * dPD] - 1.0 / 1.0 * udata[-1 * dPD] +
00182         1.0 / 6.0 * udata[0] + 5.0 / 7.0 * udata[1 * dPD] -
00183         5.0 / 28.0 * udata[2 * dPD] + 5.0 / 126.0 * udata[3 * dPD] -
00184         1.0 / 168.0 * udata[4 * dPD] + 1.0 / 2310.0 * udata[5 * dPD];
00185 }
00186 #pragma omp declare simd uniform(dPD) notinbranch
00187 inline sunrealtype s11b(sunrealtype const *udata, const int dPD) {
00188     return -1.0 / 2310.0 * udata[-5 * dPD] + 1.0 / 168.0 * udata[-4 * dPD] -
00189         5.0 / 126.0 * udata[-3 * dPD] + 5.0 / 28.0 * udata[-2 * dPD] -
00190         5.0 / 7.0 * udata[-1 * dPD] - 1.0 / 6.0 * udata[0] + udata[1 * dPD] -
00191         5.0 / 14.0 * udata[2 * dPD] + 5.0 / 42.0 * udata[3 * dPD] -
00192         5.0 / 168.0 * udata[4 * dPD] + 1.0 / 210.0 * udata[5 * dPD] -
00193         1.0 / 2772.0 * udata[6 * dPD];
00194 }
00195 #pragma omp declare simd uniform(dPD) notinbranch
00196 inline sunrealtype s12f(sunrealtype const *udata, const int dPD) {
00197     return -1.0 / 5544.0 * udata[-7 * dPD] + 1.0 / 396.0 * udata[-6 * dPD] -

```

```

00198     1.0 / 60.0 * udata[-5 * dPD] + 5.0 / 72.0 * udata[-4 * dPD] -
00199     5.0 / 24.0 * udata[-3 * dPD] + 1.0 / 2.0 * udata[-2 * dPD] -
00200     7.0 / 6.0 * udata[-1 * dPD] + 13.0 / 42.0 * udata[0] +
00201     5.0 / 8.0 * udata[1 * dPD] - 5.0 / 36.0 * udata[2 * dPD] +
00202     1.0 / 36.0 * udata[3 * dPD] - 1.0 / 264.0 * udata[4 * dPD] +
00203     1.0 / 3960.0 * udata[5 * dPD];
00204 }
00205 #pragma omp declare simd uniform(dPD) notinbranch
00206 inline sunrealtype s12c(sunrealtype const *udata, const int dPD) {
00207     return 1.0 / 5544.0 * udata[-6 * dPD] - 1.0 / 385.0 * udata[-5 * dPD] +
00208     1.0 / 56.0 * udata[-4 * dPD] - 5.0 / 63.0 * udata[-3 * dPD] +
00209     15.0 / 56.0 * udata[-2 * dPD] - 6.0 / 7.0 * udata[-1 * dPD] + 0 +
00210     6.0 / 7.0 * udata[1 * dPD] - 15.0 / 56.0 * udata[2 * dPD] +
00211     5.0 / 63.0 * udata[3 * dPD] - 1.0 / 56.0 * udata[4 * dPD] +
00212     1.0 / 385.0 * udata[5 * dPD] - 1.0 / 5544.0 * udata[6 * dPD];
00213 }
00214 #pragma omp declare simd uniform(dPD) notinbranch
00215 inline sunrealtype s12b(sunrealtype const *udata, const int dPD) {
00216     return -1.0 / 3960.0 * udata[-5 * dPD] + 1.0 / 264.0 * udata[-4 * dPD] -
00217     1.0 / 36.0 * udata[-3 * dPD] + 5.0 / 36.0 * udata[-2 * dPD] -
00218     5.0 / 8.0 * udata[-1 * dPD] - 13.0 / 42.0 * udata[0] +
00219     7.0 / 6.0 * udata[1 * dPD] - 1.0 / 2.0 * udata[2 * dPD] +
00220     5.0 / 24.0 * udata[3 * dPD] - 5.0 / 72.0 * udata[4 * dPD] +
00221     1.0 / 60.0 * udata[5 * dPD] - 1.0 / 396.0 * udata[6 * dPD] +
00222     1.0 / 5544.0 * udata[7 * dPD];
00223 }
00224 #pragma omp declare simd uniform(dPD) notinbranch
00225 inline sunrealtype s13f(sunrealtype const *udata, const int dPD) {
00226     return -1.0 / 12012.0 * udata[-7 * dPD] + 1.0 / 792.0 * udata[-6 * dPD] -
00227     1.0 / 110.0 * udata[-5 * dPD] + 1.0 / 24.0 * udata[-4 * dPD] -
00228     5.0 / 36.0 * udata[-3 * dPD] + 3.0 / 8.0 * udata[-2 * dPD] -
00229     1.0 / 1.0 * udata[-1 * dPD] + 1.0 / 7.0 * udata[0] +
00230     3.0 / 4.0 * udata[1 * dPD] - 5.0 / 24.0 * udata[2 * dPD] +
00231     1.0 / 18.0 * udata[3 * dPD] - 1.0 / 88.0 * udata[4 * dPD] +
00232     1.0 / 660.0 * udata[5 * dPD] - 1.0 / 10296.0 * udata[6 * dPD];
00233 }
00234 #pragma omp declare simd uniform(dPD) notinbranch
00235 inline sunrealtype s13b(sunrealtype const *udata, const int dPD) {
00236     return 1.0 / 10296.0 * udata[-6 * dPD] - 1.0 / 660.0 * udata[-5 * dPD] +
00237     1.0 / 88.0 * udata[-4 * dPD] - 1.0 / 18.0 * udata[-3 * dPD] +
00238     5.0 / 24.0 * udata[-2 * dPD] - 3.0 / 4.0 * udata[-1 * dPD] -
00239     1.0 / 7.0 * udata[0] + udata[1 * dPD] - 3.0 / 8.0 * udata[2 * dPD] +
00240     5.0 / 36.0 * udata[3 * dPD] - 1.0 / 24.0 * udata[4 * dPD] +
00241     1.0 / 110.0 * udata[5 * dPD] - 1.0 / 792.0 * udata[6 * dPD] +
00242     1.0 / 12012.0 * udata[7 * dPD];
00243 }
00244
00245 /////////////////
00246 // Stencils with dPD fixed to 6 //
00247 /////////////////
00248
00249 inline sunrealtype s1f(sunrealtype const *udata) { return s1f(udata, 6); }
00250 inline sunrealtype s1b(sunrealtype const *udata) { return s1b(udata, 6); }
00251 inline sunrealtype s2f(sunrealtype const *udata) { return s2f(udata, 6); }
00252 inline sunrealtype s2c(sunrealtype const *udata) { return s2c(udata, 6); }
00253 inline sunrealtype s2b(sunrealtype const *udata) { return s2b(udata, 6); }
00254 inline sunrealtype s3f(sunrealtype const *udata) { return s3f(udata, 6); }
00255 inline sunrealtype s3b(sunrealtype const *udata) { return s3b(udata, 6); }
00256 inline sunrealtype s4f(sunrealtype const *udata) { return s4f(udata, 6); }
00257 inline sunrealtype s4c(sunrealtype const *udata) { return s4c(udata, 6); }
00258 inline sunrealtype s4b(sunrealtype const *udata) { return s4b(udata, 6); }
00259 inline sunrealtype s5f(sunrealtype const *udata) { return s5f(udata, 6); }
00260 inline sunrealtype s5b(sunrealtype const *udata) { return s5b(udata, 6); }
00261 inline sunrealtype s6f(sunrealtype const *udata) { return s6f(udata, 6); }
00262 inline sunrealtype s6c(sunrealtype const *udata) { return s6c(udata, 6); }
00263 inline sunrealtype s6b(sunrealtype const *udata) { return s6b(udata, 6); }
00264 inline sunrealtype s7f(sunrealtype const *udata) { return s7f(udata, 6); }
00265 inline sunrealtype s7b(sunrealtype const *udata) { return s7b(udata, 6); }
00266 inline sunrealtype s8f(sunrealtype const *udata) { return s8f(udata, 6); }
00267 inline sunrealtype s8c(sunrealtype const *udata) { return s8c(udata, 6); }
00268 inline sunrealtype s8b(sunrealtype const *udata) { return s8b(udata, 6); }
00269 inline sunrealtype s9f(sunrealtype const *udata) { return s9f(udata, 6); }
00270 inline sunrealtype s9b(sunrealtype const *udata) { return s9b(udata, 6); }
00271 inline sunrealtype s10f(sunrealtype const *udata)
00272 { return s10f(udata, 6); }
00273 inline sunrealtype s10c(sunrealtype const *udata)
00274 { return s10c(udata, 6); }
00275 inline sunrealtype s10b(sunrealtype const *udata)
00276 { return s10b(udata, 6); }
00277 inline sunrealtype s11f(sunrealtype const *udata)
00278 { return s11f(udata, 6); }
00279 inline sunrealtype s11b(sunrealtype const *udata)
00280 { return s11b(udata, 6); }
00281 inline sunrealtype s12f(sunrealtype const *udata)
00282 { return s12f(udata, 6); }
00283 inline sunrealtype s12c(sunrealtype const *udata)
00284 { return s12c(udata, 6); }

```

```

00285 inline sunrealtype s12b(sunrealtype const *udata)
00286 { return s12b(udata, 6); }
00287 inline sunrealtype s13f(sunrealtype const *udata)
00288 { return s13f(udata, 6); }
00289 inline sunrealtype s13b(sunrealtype const *udata)
00290 { return s13b(udata, 6); }
00291

```

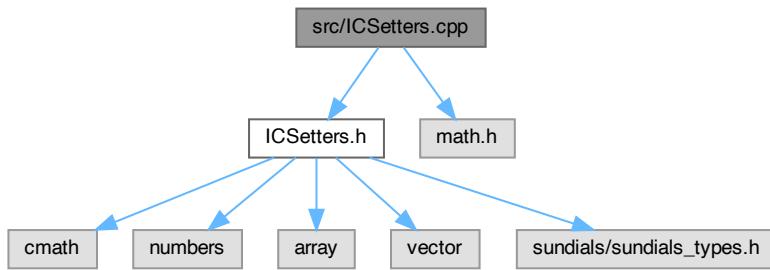
6.36 src/ICSetters.cpp File Reference

Implementation of the plane wave and Gaussian wave packets.

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

```
#include <math.h>
```

Include dependency graph for ICSetters.cpp:



6.36.1 Detailed Description

Implementation of the plane wave and Gaussian wave packets.

Definition in file [ICSetters.cpp](#).

6.37 ICSetters.cpp

[Go to the documentation of this file.](#)

```

00001 ///////////////////////////////////////////////////////////////////
00002 /// @file ICSetters.cpp
00003 /// @brief Implementation of the plane wave and Gaussian wave packets
00004 ///////////////////////////////////////////////////////////////////
00005
00006 #include "ICSetters.h"
00007
00008 #include <math.h>
00009
0010 /** PlaneWave1D construction with */
0011 PlaneWave1D::PlaneWave1D(std::array<sunrealtype, 3> k,
0012     std::array<sunrealtype, 3> p,
0013     std::array<sunrealtype, 3> phi) {
0014     kx = k[0]; /* - wavevectors \f$ k_x \f$ */
0015     ky = k[1]; /* - \f$ k_y \f$ */
0016     kz = k[2]; /* - \f$ k_z \f$ normalized to \f$ 1/\lambda \f$ */
0017     // Amplitude bug: lower by factor 3
0018     px = p[0] / 3; /* - amplitude (polarization) in x-direction \f$ p_x \f$ */
0019     py = p[1] / 3; /* - amplitude (polarization) in y-direction \f$ p_y \f$ */

```

```

00020 pz = p[2] / 3; /* - amplitude (polarization) in z-direction \f$ p_z \f$ */
00021 phix = phi[0]; /* - phase shift in x-direction \f$ \phi_x \f$ */
00022 phiy = phi[1]; /* - phase shift in y-direction \f$ \phi_y \f$ */
00023 phiz = phi[2]; /* - phase shift in z-direction \f$ \phi_z \f$ */
00024 }
00025
00026 /* PlaneWave1D implementation in space */
00027 void PlaneWave1D::addToSpace(const sunrealtype x, const sunrealtype y,
00028     const sunrealtype z,
00029     sunrealtype *pTo6Space) const {
00030     const sunrealtype wavelength =
00031         sqrt(kx * kx + ky * ky + kz * kz); /* \f$ 1/\lambda \f$ */
00032     const sunrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 *
00033         std::numbers::pi; /* \f$ 2\pi \f$ \vec{k} \cdot \vec{x} \f$ */
00034 // Plane wave definition
00035     const std::array<sunrealtype, 3> E{{
00036         px * cos(kScalarX - phix), /* \f$ E_x \f$ */
00037         py * cos(kScalarX - phiy), /* \f$ E_y \f$ */
00038         pz * cos(kScalarX - phiz)} }; /* \f$ E_z \f$ */
00039 // Put E-field into space
00040 pTo6Space[0] += E[0];
00041 pTo6Space[1] += E[1];
00042 pTo6Space[2] += E[2];
00043 // and B-field
00044 pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
00045 pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00046 pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00047 }
00048
00049 /* PlaneWave2D construction with */
00050 PlaneWave2D::PlaneWave2D(std::array<sunrealtype, 3> k,
00051     std::array<sunrealtype, 3> p,
00052     std::array<sunrealtype, 3> phi) {
00053     kx = k[0]; /* - wavevectors \f$ k_x \f$ */
00054     ky = k[1]; /* - \f$ k_y \f$ */
00055     kz = k[2]; /* - \f$ k_z \f$ normalized to \f$ 1/\lambda \f$ */
00056 // Amplitude bug: lower by factor 9
00057     px = p[0] / 9; /* - amplitude (polarization) in x-direction \f$ p_x \f$ */
00058     py = p[1] / 9; /* - amplitude (polarization) in y-direction \f$ p_y \f$ */
00059     pz = p[2] / 9; /* - amplitude (polarization) in z-direction \f$ p_z \f$ */
00060     phix = phi[0]; /* - phase shift in x-direction \f$ \phi_x \f$ */
00061     phiy = phi[1]; /* - phase shift in y-direction \f$ \phi_y \f$ */
00062     phiz = phi[2]; /* - phase shift in z-direction \f$ \phi_z \f$ */
00063 }
00064
00065 /* PlaneWave2D implementation in space */
00066 void PlaneWave2D::addToSpace(const sunrealtype x, const sunrealtype y,
00067     const sunrealtype z, sunrealtype *pTo6Space) const {
00068     const sunrealtype wavelength =
00069         sqrt(kx * kx + ky * ky + kz * kz); /* \f$ 1/\lambda \f$ */
00070     const sunrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 *
00071         std::numbers::pi; /* \f$ 2\pi \f$ \vec{k} \cdot \vec{x} \f$ */
00072 // Plane wave definition
00073     const std::array<sunrealtype, 3> E{{
00074         px * cos(kScalarX - phix), /* \f$ E_x \f$ */
00075         py * cos(kScalarX - phiy), /* \f$ E_y \f$ */
00076         pz * cos(kScalarX - phiz)} }; /* \f$ E_z \f$ */
00077 // Put E-field into space
00078 pTo6Space[0] += E[0];
00079 pTo6Space[1] += E[1];
00080 pTo6Space[2] += E[2];
00081 // and B-field
00082 pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
00083 pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00084 pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00085 }
00086
00087 /* PlaneWave3D construction with */
00088 PlaneWave3D::PlaneWave3D(std::array<sunrealtype, 3> k,
00089     std::array<sunrealtype, 3> p,
00090     std::array<sunrealtype, 3> phi) {
00091     kx = k[0]; /* - wavevectors \f$ k_x \f$ */
00092     ky = k[1]; /* - \f$ k_y \f$ */
00093     kz = k[2]; /* - \f$ k_z \f$ normalized to \f$ 1/\lambda \f$ */
00094     px = p[0]; /* - amplitude (polarization) in x-direction \f$ p_x \f$ */
00095     py = p[1]; /* - amplitude (polarization) in y-direction \f$ p_y \f$ */
00096     pz = p[2]; /* - amplitude (polarization) in z-direction \f$ p_z \f$ */
00097     phix = phi[0]; /* - phase shift in x-direction \f$ \phi_x \f$ */
00098     phiy = phi[1]; /* - phase shift in y-direction \f$ \phi_y \f$ */
00099     phiz = phi[2]; /* - phase shift in z-direction \f$ \phi_z \f$ */
00100 }
00101
00102 /* PlaneWave3D implementation in space */
00103 void PlaneWave3D::addToSpace(sunrealtype x, sunrealtype y, sunrealtype z,
00104     sunrealtype *pTo6Space) const {
00105     const sunrealtype wavelength =
00106         sqrt(kx * kx + ky * ky + kz * kz); /* \f$ 1/\lambda \f$ */

```

```

00107 const sunrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 *
00108     std::numbers::pi; /* $ 2\pi \ \vec{k} \ \cdot \vec{x} */
00109 // Plane wave definition
00110 const std::array<sunrealtype, 3> E{ /* E-field vector */
00111     px * cos(kScalarX - phix), /* $ E_x */
00112     py * cos(kScalarX - phiy), /* $ E_y */
00113     pz * cos(kScalarX - phiz)}; /* $ E_z */
00114 // Put E-field into space
00115 pTo6Space[0] += E[0];
00116 pTo6Space[1] += E[1];
00117 pTo6Space[2] += E[2];
00118 // and B-field
00119 pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
00120 pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00121 pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00122 }
00123
00124 /** Gauss1D construction with */
00125 Gauss1D::Gauss1D(std::array<sunrealtype, 3> k, std::array<sunrealtype, 3> p,
00126                     unrealtype xo, unrealtype phig_, unrealtype phi) {
00127     kx = k[0]; /* - wavevectors $ k_x $ */
00128     ky = k[1]; /* - $ k_y $ */
00129     kz = k[2]; /* - $ k_z $ normalized to $ 1/\lambda $ */
00130     px = p[0]; /* - amplitude (polarization) in x-direction */
00131     py = p[1]; /* - amplitude (polarization) in y-direction */
00132     pz = p[2]; /* - amplitude (polarization) in z-direction */
00133     phix = phi[0]; /* - phase shift in x-direction */
00134     phiy = phi[1]; /* - phase shift in y-direction */
00135     phiz = phi[2]; /* - phase shift in z-direction */
00136     phig = phig_; /* - width */
00137     x0x = xo[0]; /* - shift from origin in x-direction */
00138     x0y = xo[1]; /* - shift from origin in y-direction */
00139     x0z = xo[2]; /* - shift from origin in z-direction */
00140 }
00141 }
00142
00143 /** Gauss1D implementation in space */
00144 void Gauss1D::addToSpace(unrealtype x, unrealtype y, unrealtype z,
00145                           unrealtype *pTo6Space) const {
00146     unrealtype wavelength =
00147         sqrt(kx * kx + ky * ky + kz * kz); /* $ 1/\lambda $ */
00148     x = x - x0x; /* x-coordinate minus shift from origin */
00149     y = y - x0y; /* y-coordinate minus shift from origin */
00150     z = z - x0z; /* z-coordinate minus shift from origin */
00151     unrealtype kScalarX = (kx * x + ky * y + kz * z) * 2 *
00152         std::numbers::pi; /* $ 2\pi \ \vec{k} \ \cdot \vec{x} */
00153     unrealtype envelopeAmp =
00154         exp(-(x * x + y * y + z * z) / phig / phig); /* enveloping Gauss shape */
00155 // Gaussian wave definition
00156     const std::array<sunrealtype, 3> E{
00157         { /* E-field vector */
00158             px * cos(kScalarX - phix) * envelopeAmp, /* $ E_x $ */
00159             py * cos(kScalarX - phiy) * envelopeAmp, /* $ E_y $ */
00160             pz * cos(kScalarX - phiz) * envelopeAmp}}; /* $ E_z $ */
00161 // Put E-field into space
00162 pTo6Space[0] += E[0];
00163 pTo6Space[1] += E[1];
00164 pTo6Space[2] += E[2];
00165 // and B-field
00166 pTo6Space[3] += (ky * E[2] - kz * E[1]) / wavelength;
00167 pTo6Space[4] += (kz * E[0] - kx * E[2]) / wavelength;
00168 pTo6Space[5] += (kx * E[1] - ky * E[0]) / wavelength;
00169 }
00170
00171 /** Gauss2D construction with */
00172 Gauss2D::Gauss2D(std::array<sunrealtype, 3> dis_,
00173                     std::array<sunrealtype, 3> axis_,
00174                     unrealtype Amp_, unrealtype phip_, unrealtype w0_,
00175                     unrealtype zr_, unrealtype Ph0_, unrealtype PhA_) {
00176     dis = dis_; /* - center it approaches */
00177     axis = axis_; /* - direction form where it comes */
00178     Amp = Amp_; /* - amplitude */
00179     phip = phip_; /* - polarization rotation from TE-mode */
00180     w0 = w0_; /* - taille */
00181     zr = zr_; /* - Rayleigh length */
00182     Ph0 = Ph0_; /* - beam center */
00183     PhA = PhA_; /* - beam length */
00184     A1 = Amp * cos(phip); /* - amplitude in z-direction */
00185     A2 = Amp * sin(phip); /* - amplitude on xy-plane */
00186     lambda = std::numbers::pi * w0 * w0 / zr; /* formula for wavelength */
00187 }
00188
00189 void Gauss2D::addToSpace(unrealtype x, unrealtype y, unrealtype z,
00190                           unrealtype *pTo6Space) const {
00191     /* $ \vec{x} = \vec{x}_0 - \vec{dis} $ // coordinates minus distance to
00192     // origin
00193     x -= dis[0];

```

```

00194     y == dis[1];
00195     // z-=dis[2];
00196     z = nan("0x12345"); // unused parameter
00197     // \f$ z_g = \vec{x}\cdot\vec{e}_g \f$ projection on propagation axis
00198     const sunrealtype zg =
00199         x * axis[0] + y * axis[1]; // +z*axis[2]; // =z-z0 -> propagation
00200         // direction, minus origin
00201     // \f$ r = \sqrt{\vec{x}^2 - z_g^2} \f$ -> pythagoras of radius minus
00202     // projection on prop axis
00203     const sunrealtype r = sqrt((x * x + y * y /*+z*z*/) -
00204         zg * zg); // radial distance to propagation axis
00205     // \f$ w(z) = w_0\sqrt{1+(z_g/z_r)^2} \f$
00206     // waist at position z
00207     const sunrealtype wz = w0 * sqrt(1 + (zg * zg / zr / zr));
00208     // \f$ g(z) = \arctan(z_g/z_r) \f$
00209     const sunrealtype gz = atan(zg / zr); // Gouy phase
00210     // \f$ R(z) = z_g*(1+(z_r/z_g)^2) \f$
00211     sunrealtype Rz = nan("0x12345"); // beam curvature
00212     if (abs(zg) > 1e-15)
00213         Rz = zg * (1 + (zr * zr / zg / zg));
00214     else
00215         Rz = 1e308;
00216     // wavenumber \f$ k = 2\pi/\lambda \f$
00217     const sunrealtype k = 2 * std::numbers::pi / lambda;
00218     // \f$ \Phi_F = kr^2/(2*R(z))+g(z)-kz_g \f$
00219     const sunrealtype PhF =
00220         -k * r * r / (2 * Rz) + gz - k * zg; // to be inserted into cosine
00221     // \f$ G = (\sqrt{w_0/w_z})e^{-(r/w(z))^2}e^{-(zg-Ph0)^2/PhA^2}\cos(\Phi_F) \f$
00222     // CVode is a diva, no chance to remove the square in the second exponential
00223     // -> h too small
00224     const sunrealtype G2D = sqrt(w0 / wz) * exp(-r * r / wz / wz) *
00225         exp(-(zg - Ph0) * (zg - Ph0) / PhA / PhA) *
00226         cos(PhF); // gauss shape
00227     // \f$ c_\alpha = \vec{e}_x\cdot\vec{e}_y \f$
00228     // projection components; do like this for CVode convergence -> otherwise
00229     // results in machine error values for non-existant field components if
00230     // axis[0] and axis[1] are given
00231     const sunrealtype ca =
00232         axis[0]; // x-component of propagation axis which is given as parameter
00233     // no z-component for 2D propagation
00234     const sunrealtype sa = sqrt(1 - ca * ca);
00235     // E-field to space: polarization in xy-plane (A2) is projection of
00236     // z-polarization (A1) on x- and y-directions
00237     pTo6Space[0] += sa * (G2D * A2);
00238     pTo6Space[1] += -ca * (G2D * A2);
00239     pTo6Space[2] += G2D * A1;
00240     // B-field -> negative derivative wrt polarization shift of E-field
00241     pTo6Space[3] += -sa * (G2D * A1);
00242     pTo6Space[4] += ca * (G2D * A1);
00243     pTo6Space[5] += G2D * A2;
00244 }
00245
00246 /** Gauss3D construction with */
00247 Gauss3D::Gauss3D(std::array<sunrealtype, 3> dis_,
00248     std::array<sunrealtype, 3> axis_,
00249     unrealtype Amp_,
00250     // std::array<sunrealtype, 3> pol_,
00251     unrealtype phip_, unrealtype w0_, unrealtype zr_,
00252     unrealtype Ph0_, unrealtype PhA_) {
00253     dis = dis_; //** - center it approaches */
00254     axis = axis_; //** - direction from where it comes */
00255     Amp = Amp_; //** - amplitude */
00256     // pol=pol_;
00257     phip = phip_; //** - polarization rotation form TE-mode */
00258     w0 = w0_; //** - taille */
00259     zr = zr_; //** - Rayleigh length */
00260     Ph0 = Ph0_; //** - beam center */
00261     PhA = PhA_; //** - beam length */
00262     lambda = std::numbers::pi * w0 * w0 / zr;
00263     A1 = Amp * cos(phip);
00264     A2 = Amp * sin(phip);
00265 }
00266
00267 /** Gauss3D implementation in space */
00268 void Gauss3D::addToSpace(unrealtype x, unrealtype y, unrealtype z,
00269     unrealtype *pTo6Space) const {
00270     x -= dis[0];
00271     y -= dis[1];
00272     z -= dis[2];
00273     const sunrealtype zg = x * axis[0] + y * axis[1] + z * axis[2];
00274     const sunrealtype r = sqrt((x * x + y * y + z * z) - zg * zg);
00275     const sunrealtype wz = w0 * sqrt(1 + (zg * zg / zr / zr));
00276     const sunrealtype gz = atan(zg / zr);
00277     sunrealtype Rz = nan("0x12345");
00278     if (abs(zg) > 1e-15)
00279         Rz = zg * (1 + (zr * zr / zg / zg));
00280     else

```

```

00281     Rz = 1e308;
00282     const sunrealtype k = 2 * std::numbers::pi / lambda;
00283     const sunrealtype PhF = -k * r * r / (2 * Rz) + gz - k * zg;
00284     const sunrealtype G3D = (w0 / wz) * exp(-r * r / wz / wz) *
00285         exp(-(zg - Ph0) * (zg - Ph0) / PhA / PhA) * cos(PhF);
00286     const sunrealtype ca = axis[0];
00287     const sunrealtype sa = sqrt(1 - ca * ca);
00288     pTo6Space[0] += sa * (G3D * A2);
00289     pTo6Space[1] += -ca * (G3D * A2);
00290     pTo6Space[2] += G3D * A1;
00291     pTo6Space[3] += -sa * (G3D * A1);
00292     pTo6Space[4] += ca * (G3D * A1);
00293     pTo6Space[5] += G3D * A2;
00294 }
00295
00296 /** Evaluate lattice point values to zero and then add initial field values */
00297 void ICSetter::eval(sunrealtype x, unrealtype y, unrealtype z,
00298                      unrealtype *pTo6Space) {
00299     pTo6Space[0] = 0;
00300     pTo6Space[1] = 0;
00301     pTo6Space[2] = 0;
00302     pTo6Space[3] = 0;
00303     pTo6Space[4] = 0;
00304     pTo6Space[5] = 0;
00305     add(x, y, z, pTo6Space);
00306 }
00307
00308 /** Add all initial field values to the lattice space */
00309 void ICSetter::add(unrealtype x, unrealtype y, unrealtype z,
00310                      unrealtype *pTo6Space) {
00311     for (const auto &wave : planeWaves1D)
00312         wave.addToSpace(x, y, z, pTo6Space);
00313     for (const auto &wave : planeWaves2D)
00314         wave.addToSpace(x, y, z, pTo6Space);
00315     for (const auto &wave : planeWaves3D)
00316         wave.addToSpace(x, y, z, pTo6Space);
00317     for (const auto &wave : gauss1Ds)
00318         wave.addToSpace(x, y, z, pTo6Space);
00319     for (const auto &wave : gauss2Ds)
00320         wave.addToSpace(x, y, z, pTo6Space);
00321     for (const auto &wave : gauss3Ds)
00322         wave.addToSpace(x, y, z, pTo6Space);
00323 }
00324
00325 /** Add plane waves in 1D to their container vector */
00326 void ICSetter::addPlaneWave1D(std::array<unrealtype, 3> k,
00327                               std::array<unrealtype, 3> p,
00328                               std::array<unrealtype, 3> phi) {
00329     planeWaves1D.emplace_back(PlaneWave1D(k, p, phi));
00330 }
00331
00332 /** Add plane waves in 2D to their container vector */
00333 void ICSetter::addPlaneWave2D(std::array<unrealtype, 3> k,
00334                               std::array<unrealtype, 3> p,
00335                               std::array<unrealtype, 3> phi) {
00336     planeWaves2D.emplace_back(PlaneWave2D(k, p, phi));
00337 }
00338
00339 /** Add plane waves in 3D to their container vector */
00340 void ICSetter::addPlaneWave3D(std::array<unrealtype, 3> k,
00341                               std::array<unrealtype, 3> p,
00342                               std::array<unrealtype, 3> phi) {
00343     planeWaves3D.emplace_back(PlaneWave3D(k, p, phi));
00344 }
00345
00346 /** Add Gaussian waves in 1D to their container vector */
00347 void ICSetter::addGauss1D(std::array<unrealtype, 3> k,
00348                           std::array<unrealtype, 3> p,
00349                           std::array<unrealtype, 3> xo, unrealtype phig_,
00350                           std::array<unrealtype, 3> phi) {
00351     gauss1Ds.emplace_back(Gauss1D(k, p, xo, phig_, phi));
00352 }
00353
00354 /** Add Gaussian waves in 2D to their container vector */
00355 void ICSetter::addGauss2D(std::array<unrealtype, 3> dis_,
00356                           std::array<unrealtype, 3> axis_,
00357                           unrealtype Amp_, unrealtype phip_, unrealtype w0_,
00358                           unrealtype zr_, unrealtype Ph0_, unrealtype PhA_)
00359 {
00360     gauss2Ds.emplace_back(Gauss2D(dis_, axis_, Amp_, phip_, w0_, zr_, Ph0_, PhA_));
00361 }
00362
00363
00364 /** Add Gaussian waves in 3D to their container vector */
00365 void ICSetter::addGauss3D(std::array<unrealtype, 3> dis_,
00366                           std::array<unrealtype, 3> axis_,
00367                           unrealtype Amp_, unrealtype phip_, unrealtype w0_,

```

```

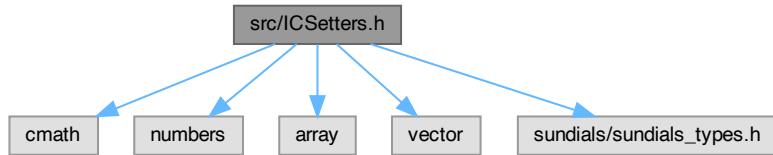
00368     sunrealtype zr_, sunrealtype Ph0_, sunrealtype PhA_)
00369 {   gauss3Ds.emplace_back(
00370     Gauss3D(dis_, axis_, Amp_, phip_, w0_, zr_, Ph0_, PhA_));
00371
00372 }

```

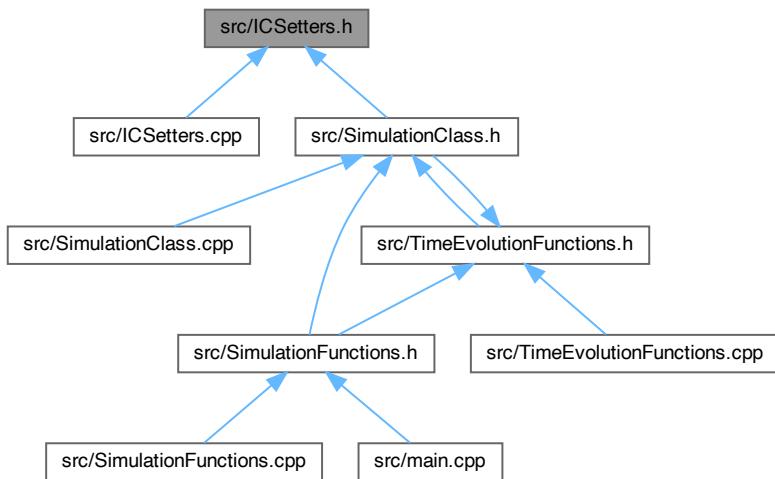
6.38 src/ICSetters.h File Reference

Declaration of the plane wave and Gaussian wave packets.

```
#include <cmath>
#include <numbers>
#include <array>
#include <vector>
#include <sundials/sundials_types.h>
Include dependency graph for ICSetters.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [PlaneWave](#)
super-class for plane waves
- class [PlaneWave1D](#)
class for plane waves in 1D
- class [PlaneWave2D](#)
class for plane waves in 2D
- class [PlaneWave3D](#)
class for plane waves in 3D
- class [Gauss1D](#)
class for Gaussian pulses in 1D
- class [Gauss2D](#)
class for Gaussian pulses in 2D
- class [Gauss3D](#)
class for Gaussian pulses in 3D
- class [ICSetter](#)
ICSetter class to initialize wave types with default parameters.

6.38.1 Detailed Description

Declaration of the plane wave and Gaussian wave packets.

Definition in file [ICSetters.h](#).

6.39 ICSetters.h

[Go to the documentation of this file.](#)

```

00001 ///////////////////////////////////////////////////////////////////
00002 /// @file ICSetters.h
00003 /// @brief Declaration of the plane wave and Gaussian wave packets
00004 ///////////////////////////////////////////////////////////////////
00005
00006 #pragma once
00007
00008 // math, constants, vector, and array
00009 #include <cmath>
00010 #include <numbers>
00011 #include <array>
00012 #include <vector>
00013
00014 #include <sundials/sundials_types.h> /* definition of type sunrealtypes */
00015
00016 /** @brief super-class for plane waves
00017 *
00018 * They are given in the form  $\vec{E} = \vec{E}_0 \cos(\vec{k} \cdot \vec{x} - \phi)$ 
00019 */
00020 class PlaneWave {
00021 protected:
00022     /// wavenumber  $k_x$ 
00023     sunrealtypes kx;
00024     /// wavenumber  $k_y$ 
00025     sunrealtypes ky;
00026     /// wavenumber  $k_z$ 
00027     sunrealtypes kz;
00028     /// polarization & amplitude in x-direction,  $p_x$ 
00029     sunrealtypes px;
00030     /// polarization & amplitude in y-direction,  $p_y$ 
00031     sunrealtypes py;
00032     /// polarization & amplitude in z-direction,  $p_z$ 
00033     sunrealtypes pz;
00034     /// phase shift in x-direction,  $\phi_x$ 
00035     sunrealtypes phix;
```

```

00036  /// phase shift in y-direction, \f$ \phi_y \f$
00037  sunrealtype phiy;
00038  /// phase shift in z-direction, \f$ \phi_z \f$
00039  sunrealtype phiz;
00040 };
00041
00042 /** @brief class for plane waves in 1D */
00043 class PlaneWave1D : public PlaneWave {
00044 public:
00045  /// construction with default parameters
00046  PlaneWave1D(std::array<sunrealtype, 3> k = {1, 0, 0},
00047      std::array<sunrealtype, 3> p = {0, 0, 1},
00048      std::array<sunrealtype, 3> phi = {0, 0, 0});
00049  /// function for the actual implementation in the lattice
00050  void addToSpace(sunrealtype x, unrealtype y, unrealtype z,
00051      unrealtype *pTo6Space) const;
00052 };
00053
00054 /** @brief class for plane waves in 2D */
00055 class PlaneWave2D : public PlaneWave {
00056 public:
00057  /// construction with default parameters
00058  PlaneWave2D(std::array<sunrealtype, 3> k = {1, 0, 0},
00059      std::array<sunrealtype, 3> p = {0, 0, 1},
00060      std::array<sunrealtype, 3> phi = {0, 0, 0});
00061  /// function for the actual implementation in the lattice
00062  void addToSpace(unrealtype x, unrealtype y, unrealtype z,
00063      unrealtype *pTo6Space) const;
00064 };
00065
00066 /** @brief class for plane waves in 3D */
00067 class PlaneWave3D : public PlaneWave {
00068 public:
00069  /// construction with default parameters
00070  PlaneWave3D(std::array<sunrealtype, 3> k = {1, 0, 0},
00071      std::array<sunrealtype, 3> p = {0, 0, 1},
00072      std::array<sunrealtype, 3> phi = {0, 0, 0});
00073  /// function for the actual implementation in space
00074  void addToSpace(unrealtype x, unrealtype y, unrealtype z,
00075      unrealtype *pTo6Space) const;
00076 };
00077
00078 /** @brief class for Gaussian pulses in 1D
00079 *
00080 * They are given in the form \f$ \vec{E}=\vec{p} \cdot \exp \left( -(\vec{x}-\vec{x}_0)^2 / \Phi_g^2 \right) \cos(\vec{k} \cdot \vec{x}) \f$
00081 */
00082
00083 class Gauss1D {
00084 private:
00085  /// wavenumber \f$ k_x \f$
00086  unrealtype kx;
00087  /// wavenumber \f$ k_y \f$
00088  unrealtype ky;
00089  /// wavenumber \f$ k_z \f$
00090  unrealtype kz;
00091  /// polarization & amplitude in x-direction, \f$ p_x \f$
00092  unrealtype px;
00093  /// polarization & amplitude in y-direction, \f$ p_y \f$
00094  unrealtype py;
00095  /// polarization & amplitude in z-direction, \f$ p_z \f$
00096  unrealtype pz;
00097  /// phase shift in x-direction, \f$ \phi_x \f$
00098  unrealtype phix;
00099  /// phase shift in y-direction, \f$ \phi_y \f$
00100  unrealtype phiy;
00101  /// phase shift in z-direction, \f$ \phi_z \f$
00102  unrealtype phiz;
00103  /// center of pulse in x-direction, \f$ x_0 \f$
00104  unrealtype x0x;
00105  /// center of pulse in y-direction, \f$ y_0 \f$
00106  unrealtype x0y;
00107  /// center of pulse in z-direction, \f$ z_0 \f$
00108  unrealtype x0z;
00109  /// pulse width \f$ \Phi_g \f$
00110  unrealtype phig;
00111
00112 public:
00113  /// construction with default parameters
00114  Gauss1D(std::array<sunrealtype, 3> k = {1, 0, 0},
00115      std::array<sunrealtype, 3> p = {0, 0, 1},
00116      std::array<sunrealtype, 3> xo = {0, 0, 0},
00117      unrealtype phig_ = 1.0,
00118      std::array<sunrealtype, 3> phi = {0, 0, 0});
00119  /// function for the actual implementation in space
00120  void addToSpace(unrealtype x, unrealtype y, unrealtype z,
00121      unrealtype *pTo6Space) const;
00122

```

```

00123 public:
00124 };
00125
00126 /** @brief class for Gaussian pulses in 2D
00127 *
00128 * They are given in the form
00129 *  $\vec{E} = A \cdot \vec{\epsilon} \sqrt{\frac{\omega_0}{\omega(z)}} \exp\left(-\frac{(z_g - \Phi_0)/\Phi_A}{2}\right)^2 \cos\left(\frac{k}{r^2}(2R(z)) + g(z) - k\right)$  with
00130 * - propagation direction (subtracted distance to origin)  $z_g$ 
00131 * - radial distance to propagation axis  $r = \sqrt{\vec{x}^2 - z_g^2}$ 
00132 * -  $k = 2\pi / \lambda$ 
00133 * - waist at position  $z$ ,  $\omega(z) = w_0$ ,  $\sqrt{1+(z_g/z_R)^2}$ 
00134 * - Gouy phase  $g(z) = \tan^{-1}(z_g/z_R)$ 
00135 * - beam curvature  $R(z) = z_g / (1+(z_g/z_R)^2)$ 
00136 * obtained via the chosen parameters */
00137
00138 class Gauss2D {
00139 private:
00140     /// distance maximum to origin
00141     std::array<sunrealtyp, 3> dis;
00142     /// normalized propagation axis
00143     std::array<sunrealtyp, 3> axis;
00144     /// amplitude  $A$ 
00145     sunrealtyp Amp;
00146     sunrealtyp phiP;
00147     /// polarization rotation from TE-mode around propagation direction
00148     // that determines  $\vec{\epsilon}$  above
00149     sunrealtyp phiP;
00150     /// taille  $\omega_0$ 
00151     sunrealtyp w0;
00152     /// Rayleigh length  $z_R = \pi \omega_0^2 / \lambda$ 
00153     sunrealtyp zR;
00154     /// center of beam  $\Phi_0$ 
00155     sunrealtyp Phi0;
00156     /// length of beam  $\Phi_A$ 
00157     sunrealtyp PhA;
00158     /// amplitude projection on TE-mode
00159     sunrealtyp A1;
00160     /// amplitude projection on xy-plane
00161     sunrealtyp A2;
00162     /// wavelength  $\lambda$ 
00163     sunrealtyp lambda;
00164
00165 public:
00166     /// construction with default parameters
00167     Gauss2D(std::array<sunrealtyp, 3> dis_ = {0, 0, 0},
00168               std::array<sunrealtyp, 3> axis_ = {1, 0, 0},
00169               sunrealtyp Amp_ = 1.0,
00170               sunrealtyp phiP_ = 0, sunrealtyp w0_ = 1e-5,
00171               sunrealtyp zR_ = 4e-5,
00172               sunrealtyp Phi0_ = 2e-5, sunrealtyp PhA_ = 0.45e-5);
00173     /// function for the actual implementation in space
00174     void addToSpace(sunrealtyp x, sunrealtyp y, sunrealtyp z,
00175                     sunrealtyp *pTo6Space) const;
00176
00177 public:
00178 };
00179
00180 /** @brief class for Gaussian pulses in 3D
00181 *
00182 * They are given in the form
00183 *  $\vec{E} = A \cdot \vec{\epsilon} \sqrt{\frac{\omega_0}{\omega(z)}} \exp\left(-\frac{(z_g - \Phi_0)/\Phi_A}{2}\right)^2 \cos\left(\frac{k}{r^2}(2R(z)) + g(z) - k\right)$  with
00184 * - propagation direction (subtracted distance to origin)  $z_g$ 
00185 * - radial distance to propagation axis  $r = \sqrt{\vec{x}^2 - z_g^2}$ 
00186 * -  $k = 2\pi / \lambda$ 
00187 * - waist at position  $z$ ,  $\omega(z) = w_0$ ,  $\sqrt{1+(z_g/z_R)^2}$ 
00188 * - Gouy phase  $g(z) = \tan^{-1}(z_g/z_R)$ 
00189 * - beam curvature  $R(z) = z_g / (1+(z_g/z_R)^2)$ 
00190 * obtained via the chosen parameters */
00191
00192 class Gauss3D {
00193 private:
00194     /// distance maximum to origin
00195     std::array<sunrealtyp, 3> dis;
00196     /// normalized propagation axis
00197     std::array<sunrealtyp, 3> axis;
00198     /// amplitude  $A$ 
00199     sunrealtyp Amp;
00200     sunrealtyp phiP;
00201     /// polarization rotation from TE-mode around propagation direction
00202     // that determines  $\vec{\epsilon}$  above
00203     sunrealtyp phiP;
00204     /// polarization
00205     std::array<sunrealtyp, 3> pol;
00206     /// taille  $\omega_0$ 
00207     sunrealtyp w0;
00208     /// Rayleigh length  $z_R = \pi \omega_0^2 / \lambda$ 
00209     sunrealtyp zR;

```

```

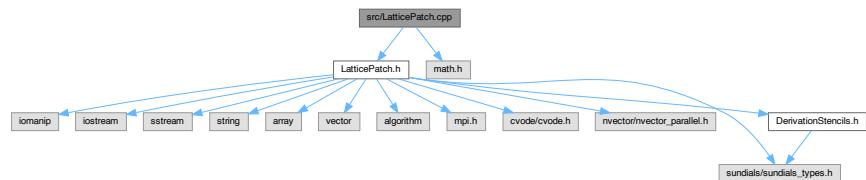
00210  /// center of beam \f$ \Phi_0 \f$
00211  surrealtype Ph0;
00212  /// length of beam \f$ \Phi_A \f$
00213  surrealtype PhA;
00214  /// amplitude projection on TE-mode (z-axis)
00215  surrealtype A1;
00216  /// amplitude projection on xy-plane
00217  surrealtype A2;
00218  /// wavelength \f$ \lambda \f$
00219  surrealtype lambda;
00220
00221 public:
00222  /// construction with default parameters
00223  Gauss3D(std::array<surrealtype, 3> dis_ = {0, 0, 0},
00224      std::array<surrealtype, 3> axis_ = {1, 0, 0},
00225      surrealtype Amp_ = 1.0,
00226      surrealtype phip_ = 0,
00227      // surrealtype pol_={0,0,1},
00228      surrealtype w0_ = 1e-5, surrealtype zr_ = 4e-5,
00229      surrealtype Ph0_ = 2e-5, surrealtype PhA_ = 0.45e-5);
00230  /// function for the actual implementation in space
00231  void addToSpace(surrealtype x, surrealtype y, surrealtype z,
00232      surrealtype *pTo6Space) const;
00233
00234 public:
00235 };
00236
00237 /** @brief ICSetter class to initialize wave types with default parameters */
00238 class ICSetter {
00239 private:
00240  /// container vector for plane waves in 1D
00241  std::vector<PlaneWave1D> planeWaves1D;
00242  /// container vector for plane waves in 2D
00243  std::vector<PlaneWave2D> planeWaves2D;
00244  /// container vector for plane waves in 3D
00245  std::vector<PlaneWave3D> planeWaves3D;
00246  /// container vector for Gaussian wave packets in 1D
00247  std::vector<Gauss1D> gauss1Ds;
00248  /// container vector for Gaussian wave packets in 2D
00249  std::vector<Gauss2D> gauss2Ds;
00250  /// container vector for Gaussian wave packets in 3D
00251  std::vector<Gauss3D> gauss3Ds;
00252
00253 public:
00254  /// function to set all coordinates to zero and then 'add' the field values
00255  void eval(surrealtype x, surrealtype y, surrealtype z,
00256      surrealtype *pTo6Space);
00257  /// function to fill the lattice space with initial field values
00258  // of all field vector containers
00259  void add(surrealtype x, surrealtype y, surrealtype z,
00260      surrealtype *pTo6Space);
00261  /// function to add plane waves in 1D to their container vector
00262  void addPlaneWave1D(std::array<surrealtype, 3> k = {1, 0, 0},
00263      std::array<surrealtype, 3> p = {0, 0, 1},
00264      std::array<surrealtype, 3> phi = {0, 0, 0});
00265  /// function to add plane waves in 2D to their container vector
00266  void addPlaneWave2D(std::array<surrealtype, 3> k = {1, 0, 0},
00267      std::array<surrealtype, 3> p = {0, 0, 1},
00268      std::array<surrealtype, 3> phi = {0, 0, 0});
00269  /// function to add plane waves in 3D to their container vector
00270  void addPlaneWave3D(std::array<surrealtype, 3> k = {1, 0, 0},
00271      std::array<surrealtype, 3> p = {0, 0, 1},
00272      std::array<surrealtype, 3> phi = {0, 0, 0});
00273  /// function to add Gaussian wave packets in 1D to their container vector
00274  void addGauss1D(std::array<surrealtype, 3> k = {1, 0, 0},
00275      std::array<surrealtype, 3> p = {0, 0, 1},
00276      std::array<surrealtype, 3> xo = {0, 0, 0},
00277      surrealtype phig_ = 1.0,
00278      std::array<surrealtype, 3> phi = {0, 0, 0});
00279  /// function to add Gaussian wave packets in 2D to their container vector
00280  void addGauss2D(std::array<surrealtype, 3> dis_ = {0, 0, 0},
00281      std::array<surrealtype, 3> axis_ = {1, 0, 0},
00282      surrealtype Amp_ = 1.0, surrealtype phip_ = 0,
00283      surrealtype w0_ = 1e-5, surrealtype zr_ = 4e-5,
00284      surrealtype Ph0_ = 2e-5, surrealtype PhA_ = 0.45e-5);
00285  /// function to add Gaussian wave packets in 3D to their container vector
00286  void addGauss3D(std::array<surrealtype, 3> dis_ = {0, 0, 0},
00287      std::array<surrealtype, 3> axis_ = {1, 0, 0},
00288      surrealtype Amp_ = 1.0, surrealtype phip_ = 0,
00289      surrealtype w0_ = 1e-5, surrealtype zr_ = 4e-5,
00290      surrealtype Ph0_ = 2e-5, surrealtype PhA_ = 0.45e-5);
00291 };
00292

```

6.40 src/LatticePatch.cpp File Reference

Construction of the overall envelope lattice and the lattice patches.

```
#include "LatticePatch.h"
#include <math.h>
Include dependency graph for LatticePatch.cpp:
```



Functions

- int [generatePatchwork](#) (const [Lattice](#) &envelopeLattice, [LatticePatch](#) &patchToMold, const int DLx, const int DLy, const int DLz)

Set up the patchwork.
- void [errorKill](#) (const std::string &errorMessage)

Print a specific error message to stderr.
- int [check_error](#) (int error, const char *funcname, int id)

helper function to check MPI errors
- int [check_retval](#) (void *returnvalue, const char *funcname, int opt, int id)

helper function to check CVode errors

6.40.1 Detailed Description

Construction of the overall envelope lattice and the lattice patches.

Definition in file [LatticePatch.cpp](#).

6.40.2 Function Documentation

6.40.2.1 check_error()

```
int check_error (
    int error,
    const char * funcname,
    int id )
```

helper function to check MPI errors

Check MPI errors. Error handler must be set.

Definition at line 928 of file [LatticePatch.cpp](#).

```
00928     {
00929         int eclass, len;
00930         char errorstring[MPI_MAX_ERROR_STRING];
00931         if( error != MPI_SUCCESS ) {
00932             MPI_Error_class(error,&eclass);
00933             MPI_Error_string(error,errorstring,&len);
00934             std::cerr << "MPI Error(process " << id << ") in " << funcname << " : "
00935             << errorstring << ", from class " << eclass << std::endl;
00936         return 1;
00937     }
00938     return 0;
00939 }
```

6.40.2.2 check_retval()

```
int check_retval (
    void * returnvalue,
    const char * funcname,
    int opt,
    int id )
```

helper function to check CVode errors

Check function return value. Adapted from CVode examples. opt == 0 means SUNDIALS function allocates memory so check if returned NULL pointer opt == 1 means SUNDIALS function returns an integer value so check if retval < 0 opt == 2 means function allocates memory so check if returned NULL pointer

Definition at line 948 of file [LatticePatch.cpp](#).

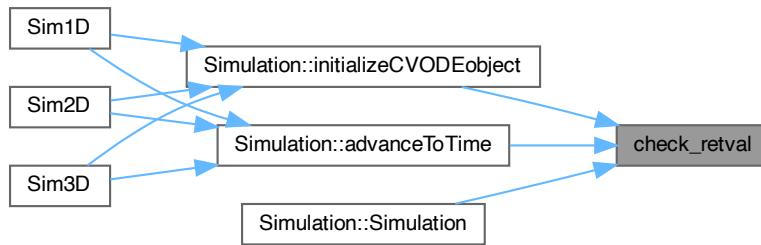
```
00948     {
00949         int *retval = nullptr;
00950
00951         /* Check if SUNDIALS function returned NULL pointer - no memory allocated */
00952         if (opt == 0 && returnvalue == nullptr) {
00953             fprintf(stderr,
00954                 "\nSUNDIALS_ERROR(%d): %s() failed - returned NULL pointer\n\n", id,
00955                 funcname);
00956             return (1);
00957         }
00958
00959         /* Check if retval < 0 */
00960         else if (opt == 1) {
00961             retval = (int *)returnvalue;
00962             char *flagname = CVodeGetReturnFlagName(*retval);
00963             if (*retval < 0) {
00964                 fprintf(stderr, "\nSUNDIALS_ERROR(%d): %s() failed with retval = %d: "
00965                     "%s\n\n",
00966                     id, funcname, *retval, flagname);
00967             return (1);
00968         }
00969     }
00970
00971         /* Check if function returned NULL pointer - no memory allocated */
00972         else if (opt == 2 && returnvalue == nullptr) {
00973             fprintf(stderr,
00974                 "\nMEMORY_ERROR(%d): %s() failed - returned NULL pointer\n\n", id,
00975                 funcname);
```

```

00976     return (1);
00977 }
00978
00979 return (0);
00980 }
```

Referenced by [Simulation::advanceToTime\(\)](#), [Simulation::initializeCVODEobject\(\)](#), and [Simulation::Simulation\(\)](#).

Here is the caller graph for this function:



6.40.2.3 errorKill()

```

void errorKill (
    const std::string & errorMessage )
```

Print a specific error message to stderr.

helper function for error messages

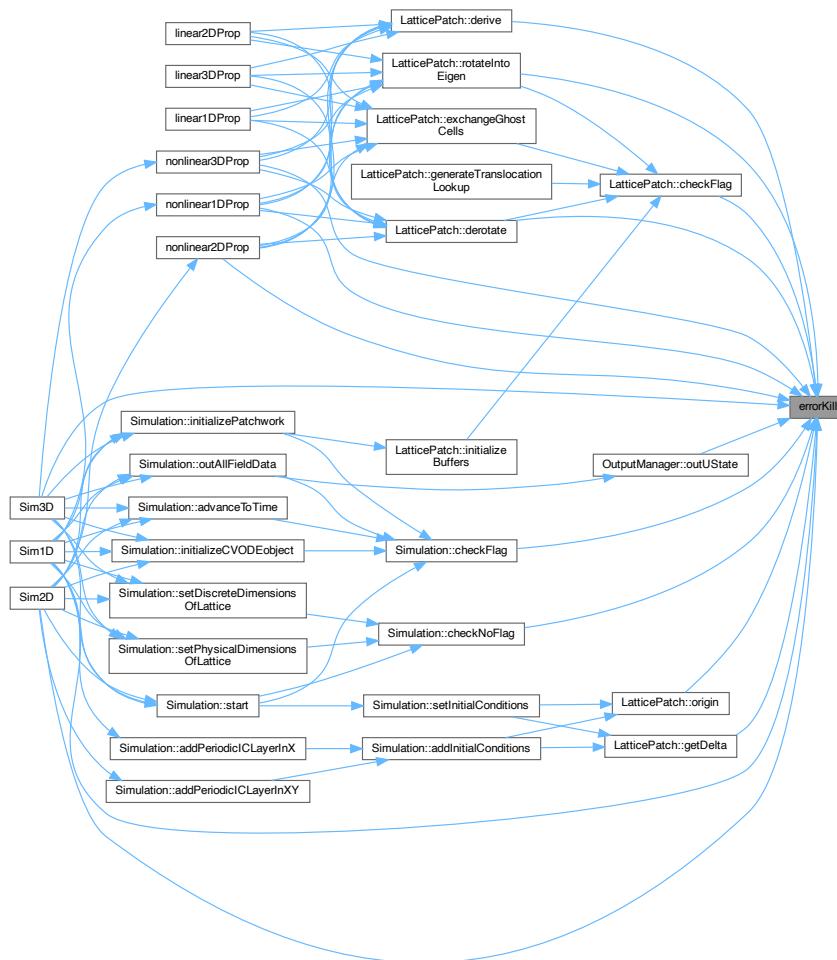
Definition at line 916 of file [LatticePatch.cpp](#).

```

00916     int my_prc=0;
00917     MPI_Comm_rank (MPI_COMM_WORLD, &my_prc);
00918     if (my_prc==0) {
00919         std::cerr << std::endl << "Error: " << errorMessage
00920         << "\nAborting..." << std::endl;
00921         MPI_Abort (MPI_COMM_WORLD, 1);
00922         return;
00923     }
00924 }
```

Referenced by [LatticePatch::checkFlag\(\)](#), [Simulation::checkFlag\(\)](#), [Simulation::checkNoFlag\(\)](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::getDelta\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), [nonlinear3DProp\(\)](#), [LatticePatch::origin\(\)](#), [OutputManager::outUState\(\)](#), [LatticePatch::rotateIntoEigen\(\)](#), [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the caller graph for this function:



6.40.2.4 generatePatchwork()

```

int generatePatchwork (
    const Lattice & envelopeLattice,
    LatticePatch & patchToMold,
    const int DLx,
    const int DLy,
    const int DLz )
    
```

Set up the patchwork.

friend function for creating the patchwork slicing of the overall lattice

Definition at line 109 of file [LatticePatch.cpp](#).

```

00111 // Retrieve the ghost layer depth
00112 const int gLW = envelopeLattice.get_ghostLayerWidth();
00113 // Retrieve the data point dimension
00114 const int dPD = envelopeLattice.get_dataPointDimension();
    
```

```

00116 // MPI process/patch
00117 const int my_prc = envelopeLattice.my_prc;
00118 // Determine thickness of the slice
00119 const sunindextype tot_NOXP = envelopeLattice.get_tot_nx();
00120 const sunindextype tot_NOYP = envelopeLattice.get_tot_ny();
00121 const sunindextype tot_NOZP = envelopeLattice.get_tot_nz();
00122 // position of the patch in the lattice of patches -> process associated to
00123 // position
00124 const sunindextype LIx = my_prc % DLx;
00125 const sunindextype LIy = (my_prc / DLx) % DLy;
00126 const sunindextype LIz = (my_prc / DLx) / DLz;
00127 // Determine the number of points in the patch and first absolute points in
00128 // each dimension
00129 const sunindextype local_NOXP = tot_NOXP / DLx;
00130 const sunindextype local_NOYP = tot_NOYP / DLy;
00131 const sunindextype local_NOZP = tot_NOZP / DLz;
00132 // absolute positions of the first point in each dimension
00133 const sunindextype firstXPoint = local_NOXP * LIx;
00134 const sunindextype firstYPoint = local_NOYP * LIy;
00135 const sunindextype firstZPoint = local_NOZP * LIz;
00136 // total number of points in the patch
00137 const sunindextype local_NODP = dPD * local_NOXP * local_NOYP * local_NOZP;
00138
00139 // Set patch up with above derived quantities
00140 patchToMold.dx = envelopeLattice.get_dx();
00141 patchToMold.dy = envelopeLattice.get_dy();
00142 patchToMold.dz = envelopeLattice.get_dz();
00143 patchToMold.x0 = firstXPoint * patchToMold.dx;
00144 patchToMold.y0 = firstYPoint * patchToMold.dy;
00145 patchToMold.z0 = firstZPoint * patchToMold.dz;
00146 patchToMold.LIx = LIx;
00147 patchToMold.LIy = LIy;
00148 patchToMold.LIz = LIz;
00149 patchToMold.nx = local_NOXP;
00150 patchToMold.ny = local_NOYP;
00151 patchToMold.nz = local_NOZP;
00152 patchToMold.lx = patchToMold.nx * patchToMold.dx;
00153 patchToMold.ly = patchToMold.ny * patchToMold.dy;
00154 patchToMold.lz = patchToMold.nz * patchToMold.dz;
00155
00156 #ifdef _OPENMP
00157 // OpenMP and MPI+X NVector interoperability
00158 // OpenMP NVectors with local patch size
00159 int num_threads = 1;
00160 num_threads = omp_get_max_threads();
00161 patchToMold.uLocal = N_VNew_OpenMP(local_NODP, num_threads,
00162     envelopeLattice.sunctx);
00163 patchToMold.duLocal = N_VNew_OpenMP(local_NODP, num_threads,
00164     envelopeLattice.sunctx);
00165 // MPI+X NVectors containing local OpenMP NVectors
00166 patchToMold.u = N_VMake_MPIPlusX(envelopeLattice.comm, patchToMold.uLocal,
00167     envelopeLattice.sunctx);
00168 patchToMold.du = N_VMake_MPIPlusX(envelopeLattice.comm, patchToMold.duLocal,
00169     envelopeLattice.sunctx);
00170 // Pointers to local vectors
00171 patchToMold.uData = N_VGetArrayPointer_MPIPlusX(patchToMold.u);
00172 patchToMold.duData = N_VGetArrayPointer_MPIPlusX(patchToMold.du);
00173 #else
00174 // MPI NVectors with local patch and global lattice size
00175 patchToMold.u =
00176     N_VNew_Parallel(envelopeLattice.comm, local_NODP,
00177         envelopeLattice.get_tot_noDP(), envelopeLattice.sunctx);
00178 patchToMold.du =
00179     N_VNew_Parallel(envelopeLattice.comm, local_NODP,
00180         envelopeLattice.get_tot_noDP(), envelopeLattice.sunctx);
00181 patchToMold.uData = NV_DATA_P(patchToMold.u);
00182 patchToMold.duData = NV_DATA_P(patchToMold.du);
00183 #endif
00184
00185 // Allocate space for auxiliary uAux so that the lattice and all possible
00186 // directions of ghost Layers fit
00187 const sunindextype s1 = patchToMold.nx, s2 = patchToMold.ny,
00188 s3 = patchToMold.nz;
00189 const sunindextype s_min = std::min(s1, std::min(s2, s3));
00190 patchToMold.uAux.resize(s1 * s2 * s3 / s_min * (s_min + 2 * gLW) * dPD);
00191 patchToMold.uAuxData = &patchToMold.uAux[0];
00192 patchToMold.envelopeLattice = &envelopeLattice;
00193 // Set patch "name" to process number -> only for debugging
00194 // patchToMold.ID=my_prc;
00195 // set flag
00196 patchToMold.statusFlags = FlatticePatchSetUp;
00197 patchToMold.generateTranslocationLookup();
00198 return 0;
00199 }

```

Referenced by [Simulation::initializePatchwork\(\)](#).

Here is the caller graph for this function:



6.41 LatticePatch.cpp

[Go to the documentation of this file.](#)

```

00001 //////////////////////////////////////////////////////////////////
00002 /// @file LatticePatch.cpp
00003 /// @brief Construction of the overall envelope lattice and the lattice patches
00004 //////////////////////////////////////////////////////////////////
00005
00006 #include "LatticePatch.h"
00007
00008 #include <math.h>
00009
00010 //////////////////////////////////////////////////////////////////
00011 /// Implementation of Lattice component functions ///
00012 //////////////////////////////////////////////////////////////////
00013
00014 /// Initialize the cartesian communicator
00015 void Lattice::initializeCommunicator(const int Nx, const int Ny,
00016                                         const int Nz, const bool per) {
00017     const int dims[3] = {Nz, Ny, Nx};
00018     const int periods[3] = {static_cast<int>(per), static_cast<int>(per),
00019                           static_cast<int>(per)};
00020     // Create the cartesian communicator for MPI_COMM_WORLD
00021     MPI_Cart_create(MPI_COMM_WORLD, 3, dims, periods, 1, &comm);
00022     // Set MPI variables of the lattice
00023     MPI_Comm_size(comm, &(n_prc));
00024     MPI_Comm_rank(comm, &(my_prc));
00025     // Associate name to the communicator to identify it -> for debugging and
00026     // nicer error messages
00027     constexpr char lattice_comm_name[] = "Lattice";
00028     MPI_Comm_set_name(comm, lattice_comm_name);
00029 }
00030
00031 /// Construct the lattice and set the stencil order
00032 Lattice::Lattice(const int Sto) : stencilOrder(Sto),
00033     ghostLayerWidth(Sto/2+1) {
00034     statusFlags = 0;
00035 }
00036
00037 /// Set the number of points in each dimension of the lattice
00038 void Lattice::setDiscreteDimensions(const sunindextype _nx,
00039                                       const sunindextype _ny, const sunindextype _nz) {
00040     // copy the given data for number of points
00041     tot_nx = _nx;
00042     tot_ny = _ny;
00043     tot_nz = _nz;
00044     // compute the resulting number of points and datapoints
00045     tot_noP = tot_nx * tot_ny * tot_nz;
00046     tot_noDP = dataPointDimension * tot_noP;
00047     // compute the new Delta, the physical resolution
00048     dx = tot_lx / tot_nx;
00049     dy = tot_ly / tot_ny;
00050     dz = tot_lz / tot_nz;
00051 }
00052
00053 /// Set the physical size of the lattice
00054 void Lattice::setPhysicalDimensions(const sunrealtype _lx,
00055                                       const sunrealtype _ly, const sunrealtype _lz) {
00056     tot_lx = _lx;
  
```

```

00057     tot_ly = _ly;
00058     tot_lz = _lz;
00059     // calculate physical distance between points
00060     dx = tot_lx / tot_nx;
00061     dy = tot_ly / tot_ny;
00062     dz = tot_lz / tot_nz;
00063     statusFlags |= FLatticeDimensionSet;
00064 }
00065
00066 ///////////////////////////////////////////////////////////////////
00067 // Implementation of LatticePatch component functions ////
00068 ///////////////////////////////////////////////////////////////////
00069
00070 // Construct the lattice patch
00071 LatticePatch::LatticePatch() {
00072     // set default origin coordinates to (0,0,0)
00073     x0 = y0 = z0 = 0;
00074     // set default position in Lattice-Patchwork to (0,0,0)
00075     LIx = LIy = LIz = 0;
00076     // set default physical length for lattice patch to (0,0,0)
00077     lx = ly = lz = 0;
00078     // set default discrete length for lattice patch to (0,1,1)
00079     /* This is done in this manner as even in 1D simulations require a 1 point
00080      * width */
00081     nx = 0;
00082     ny = nz = 1;
00083
00084     // u is not initialized as it wouldn't make any sense before the dimensions
00085     // are set idem for the enveloping lattice
00086
00087     // set default statusFlags to non set
00088     statusFlags = 0;
00089 }
00090
00091 // Destruct the patch and thereby destroy the NVectors
00092 LatticePatch::~LatticePatch() {
00093     // Deallocate memory for solution vector
00094     if (statusFlags & FLatticePatchSetUp) {
00095         // Destroy data vectors
00096 #if defined(_OPENMP)
00097         N_VDestroy(u);
00098         N_VDestroy(du);
00099         N_VDestroy_OpenMP(uLocal);
00100         N_VDestroy_OpenMP(duLocal);
00101 #else
00102         N_VDestroy_Parallel(u);
00103         N_VDestroy_Parallel(du);
00104 #endif
00105     }
00106 }
00107
00108 // Set up the patchwork
00109 int generatePatchwork(const Lattice &envelopeLattice,
00110     LatticePatch &patchToMold,
00111     const int DLx, const int DLy, const int DLz) {
00112     // Retrieve the ghost layer depth
00113     const int gLW = envelopeLattice.get_ghostLayerWidth();
00114     // Retrieve the data point dimension
00115     const int dPD = envelopeLattice.get_dataPointDimension();
00116     // MPI process/patch
00117     const int my_prc = envelopeLattice.my_prc;
00118     // Determine thickness of the slice
00119     const sunindextype tot_NOXP = envelopeLattice.get_tot_nx();
00120     const sunindextype tot_NOYP = envelopeLattice.get_tot_ny();
00121     const sunindextype tot_NOZP = envelopeLattice.get_tot_nz();
00122     // position of the patch in the lattice of patches -> process associated to
00123     // position
00124     const sunindextype LIx = my_prc % DLx;
00125     const sunindextype LIy = (my_prc / DLx) % DLy;
00126     const sunindextype LIz = (my_prc / DLx) / DLy;
00127     // Determine the number of points in the patch and first absolute points in
00128     // each dimension
00129     const sunindextype local_NOXP = tot_NOXP / DLx;
00130     const sunindextype local_NOYP = tot_NOYP / DLy;
00131     const sunindextype local_NOZP = tot_NOZP / DLz;
00132     // absolute positions of the first point in each dimension
00133     const sunindextype firstXPoint = local_NOXP * LIx;
00134     const sunindextype firstYPoint = local_NOYP * LIy;
00135     const sunindextype firstZPoint = local_NOZP * LIz;
00136     // total number of points in the patch
00137     const sunindextype local_NODP = dPD * local_NOXP * local_NOYP * local_NOZP;
00138
00139     // Set patch up with above derived quantities
00140     patchToMold.dx = envelopeLattice.get_dx();
00141     patchToMold.dy = envelopeLattice.get_dy();
00142     patchToMold.dz = envelopeLattice.get_dz();
00143     patchToMold.x0 = firstXPoint * patchToMold.dx;

```

```

00144 patchToMold.y0 = firstYPoint * patchToMold.dy;
00145 patchToMold.z0 = firstZPoint * patchToMold.dz;
00146 patchToMold.LIx = LIx;
00147 patchToMold.LIy = LIy;
00148 patchToMold.LIz = LIz;
00149 patchToMold.nx = local_NOXP;
00150 patchToMold.ny = local_NOYP;
00151 patchToMold.nz = local_NOZP;
00152 patchToMold.lx = patchToMold.nx * patchToMold.dx;
00153 patchToMold.ly = patchToMold.ny * patchToMold.dy;
00154 patchToMold.lz = patchToMold.nz * patchToMold.dz;
00155
00156 #ifdef _OPENMP
00157 // OpenMP and MPI+X NVector interoperability
00158 // OpenMP NVectors with local patch size
00159 int num_threads = 1;
00160 num_threads = omp_get_max_threads();
00161 patchToMold.uLocal = N_VNew_OpenMP(local_NODP, num_threads,
00162 envelopeLattice.sunctx);
00163 patchToMold.duLocal = N_VNew_OpenMP(local_NODP, num_threads,
00164 envelopeLattice.sunctx);
00165 // MPI+X NVectors containing local OpenMP NVectors
00166 patchToMold.u = N_VMake_MPIPlusX(envelopeLattice.comm, patchToMold.uLocal,
00167 envelopeLattice.sunctx);
00168 patchToMold.du = N_VMake_MPIPlusX(envelopeLattice.comm, patchToMold.duLocal,
00169 envelopeLattice.sunctx);
00170 // Pointers to local vectors
00171 patchToMold.uData = N_VGetArrayPointer_MPIPlusX(patchToMold.u);
00172 patchToMold.duData = N_VGetArrayPointer_MPIPlusX(patchToMold.du);
00173 #else
00174 // MPI NVectors with local patch and global lattice size
00175 patchToMold.u =
00176     N_VNew_Parallel(envelopeLattice.comm, local_NODP,
00177 envelopeLattice.get_tot_noDP(), envelopeLattice.sunctx);
00178 patchToMold.du =
00179     N_VNew_Parallel(envelopeLattice.comm, local_NODP,
00180 envelopeLattice.get_tot_noDP(), envelopeLattice.sunctx);
00181 patchToMold.uData = NV_DATA_P(patchToMold.u);
00182 patchToMold.duData = NV_DATA_P(patchToMold.du);
00183 #endif
00184
00185 // Allocate space for auxiliary uAux so that the lattice and all possible
00186 // directions of ghost Layers fit
00187 const sunindextype s1 = patchToMold.nx, s2 = patchToMold.ny,
00188     s3 = patchToMold.nz;
00189 const sunindextype s_min = std::min(s1, std::min(s2, s3));
00190 patchToMold.uAux.resize(s1 * s2 * s3 / s_min * (s_min + 2 * gLW) * dPD);
00191 patchToMold.uAuxData = &patchToMold.uAux[0];
00192 patchToMold.envelopeLattice = &envelopeLattice;
00193 // Set patch "name" to process number -> only for debugging
00194 // patchToMold.ID=my_prc;
00195 // set flag
00196 patchToMold.statusFlags = FLatticePatchSetUp;
00197 patchToMold.generateTranslocationLookup();
00198 return 0;
00199 }
00200
00201 /// Return the discrete size of the patch: number of lattice patch points in
00202 /// specified dimension
00203 sunindextype LatticePatch::discreteSize(int dir) const {
00204     switch (dir) {
00205     case 0:
00206         return nx * ny * nz;
00207     case 1:
00208         return nx;
00209     case 2:
00210         return ny;
00211     case 3:
00212         return nz;
00213     // case 4: return uAux.size(); // for debugging
00214     default:
00215         return -1;
00216     }
00217 }
00218
00219 /// Return the physical origin of the patch in a dimension
00220 sunrealtype LatticePatch::origin(const int dir) const {
00221     switch (dir) {
00222     case 1:
00223         return x0;
00224     case 2:
00225         return y0;
00226     case 3:
00227         return z0;
00228     default:
00229         errorKill("LatticePatch::origin function called with wrong dir parameter");
00230         return -1;

```

```

00231     }
00232 }
00233
00234 // Return the distance between points in the patch in a dimension
00235 sunrealtype LatticePatch::getDelta(const int dir) const {
00236     switch (dir) {
00237     case 1:
00238         return dx;
00239     case 2:
00240         return dy;
00241     case 3:
00242         return dz;
00243     default:
00244         errorKill(
00245             "LatticePatch::getDelta function called with wrong dir parameter");
00246         return -1;
00247     }
00248 }
00249
00250 /** In order to avoid cache misses:
00251 * create vectors to translate u vector into space coordinates and vice versa
00252 * and same for left and right ghost layers to space */
00253 void LatticePatch::generateTranslocationLookup() {
00254     // Check that the lattice has been set up
00255     checkFlag(FLatticeDimensionSet);
00256     // lengths for auxilliary layers, including ghost layers
00257     const int gLW = envelopeLattice->get_ghostLayerWidth();
00258     const sunindextype mx = nx + 2 * gLW;
00259     const sunindextype my = ny + 2 * gLW;
00260     const sunindextype mz = nz + 2 * gLW;
00261     // sizes for lookup vectors
00262     const sunindextype totalNP = nx * ny * nz;
00263     const sunindextype haloXSize = mx * ny * nz;
00264     const sunindextype haloYSize = nx * my * nz;
00265     const sunindextype haloZSize = nx * ny * mz;
00266     // generate u->uAux
00267     uTox.resize(totalNP);
00268     uToy.resize(totalNP);
00269     uToz.resize(totalNP);
00270     // generate uAux->u with length including halo
00271     xTou.resize(haloXSize);
00272     yTou.resize(haloYSize);
00273     zTou.resize(haloZSize);
00274     // same for ghost layer lookup tables
00275     const sunindextype ghostXSize = gLW * ny * nz;
00276     const sunindextype ghostYSize = gLW * nx * nz;
00277     const sunindextype ghostZSize = gLW * nx * ny;
00278     lgcTox.resize(ghostXSize);
00279     rgcTox.resize(ghostXSize);
00280     lgcToy.resize(ghostYSize);
00281     rgcToy.resize(ghostYSize);
00282     lgcToz.resize(ghostZSize);
00283     rgcToz.resize(ghostZSize);
00284     // variables for cartesian position in the 3D discrete lattice
00285     sunindextype px = 0, py = 0, pz = 0;
00286     // Fill the lookup tables
00287     #pragma omp parallel default(none) \
00288     private(px, py, pz) \
00289     shared(uTox, uToy, uToz, xTou, yTou, zTou, \
00290            nx, ny, mx, my, mz, gLW, totalNP, \
00291            lgcTox, rgcTox, lgcToy, lgcToz, rgcToz, \
00292            ghostXSize, ghostYSize, ghostZSize)
00293     {
00294     #pragma omp for simd schedule(static)
00295     for (sunindextype i = 0; i < totalNP; i++) { // loop over the patch
00296         // calculate cartesian coordinates
00297         px = i % nx;
00298         py = (i / nx) % ny;
00299         pz = (i / nx) / ny;
00300         // fill lookups extended by halos (useful for y and z direction)
00301         uTox[i] = (px + gLW) + py * mx +
00302                     pz * mx * ny; // unroll (de-flatten) cartesian dimension
00303         xTou[px + py * mx + pz * mx * ny] =
00304             i; // match cartesian point to u location
00305         uToy[i] = (py + gLW) + pz * my + px * my * nz;
00306         yTou[py + pz * my + px * my * nz] = i;
00307         uToz[i] = (pz + gLW) + px * mz + py * mz * nx;
00308         zTou[pz + px * mz + py * mz * nx] = i;
00309     }
00310     #pragma omp for simd schedule(static)
00311     for (sunindextype i = 0; i < ghostXSize; i++) {
00312         px = i % gLW;
00313         py = (i / gLW) % ny;
00314         pz = (i / gLW) / ny;
00315         lgcTox[i] = px + py * mx + pz * mx * ny;
00316         rgcTox[i] = px + nx + gLW + py * mx + pz * mx * ny;
00317     }

```

```

00318 #pragma omp for simd schedule(static)
00319 for (sunindextype i = 0; i < ghostYSize; i++) {
00320     px = i % nx;
00321     py = (i / nx) % gLW;
00322     pz = (i / nx) / gLW;
00323     lgcToy[i] = py + pz * my + px * my * nz;
00324     rgcToy[i] = py + ny + gLW + pz * my + px * my * nz;
00325 }
00326 #pragma omp for simd schedule(static)
00327 for (sunindextype i = 0; i < ghostZSize; i++) {
00328     px = i % nx;
00329     py = (i / nx) % ny;
00330     pz = (i / nx) / ny;
00331     lgcToz[i] = pz + px * mz + py * mz * nx;
00332     rgcToz[i] = pz + nz + gLW + px * mz + py * mz * nx;
00333 }
00334 }
00335 statusFlags |= TranslocationLookupSetUp;
00336 }
00337
00338 /** Rotate into eigenraum along R matrices of paper using the rotation
00339 * methods;
00340 * uAuxData gets the rotated left-halo-, inner-patch-, right-halo-data */
00341 void LatticePatch::rotateIntoEigen(const int dir) {
00342 // Check that the lattice, ghost layers as well as the translocation lookups
00343 // have been set up;
00344 checkFlag(FLatticePatchSetUp);
00345 checkFlag(TranslocationLookupSetUp);
00346 checkFlag(GhostLayersInitialized); // this check is only after call to
00347 // exchange ghost cells
00348 switch (dir) {
00349 case 1:
00350     rotateToX(uAuxData, gCLData, lgcTox);
00351     rotateToX(uAuxData, uData, uTox);
00352     rotateToX(uAuxData, gCRData, rgcTox);
00353     break;
00354 case 2:
00355     rotateToY(uAuxData, gCLData, lgcToy);
00356     rotateToY(uAuxData, uData, uToy);
00357     rotateToY(uAuxData, gCRData, rgcToy);
00358     break;
00359 case 3:
00360     rotateToZ(uAuxData, gCLData, lgcToz);
00361     rotateToZ(uAuxData, uData, uToz);
00362     rotateToZ(uAuxData, gCRData, rgcToz);
00363     break;
00364 default:
00365     errorKill("Tried to rotate into the wrong direction");
00366     break;
00367 }
00368 }
00369
00370 /// Rotate halo and inner-patch data vectors with rotation matrix Rx into
00371 /// eigenspace of Z matrix and write to auxiliary vector
00372 inline void LatticePatch::rotateToX(sunrealtype *outArray,
00373                                     const unrealtype *inArray,
00374                                     const std::vector<sunindextype> &lookup) {
00375     sunindextype ii = 0, target = 0;
00376     const sunindextype size = lookup.size();
00377     const int dPD = envelopeLattice->get_dataPointDimension();
00378     #pragma omp parallel for simd \
00379     private(target, ii) \
00380     shared(lookup, outArray, inArray, size, dPD) \
00381     schedule(static)
00382     for (sunindextype i = 0; i < size; i++) {
00383         // get correct u-vector and spatial indices along previously defined lookup
00384         // tables
00385         target = dPD * lookup[i];
00386         ii = dPD * i;
00387         outArray[target + 0] = -inArray[1 + ii] + inArray[5 + ii];
00388         outArray[target + 1] = inArray[2 + ii] + inArray[4 + ii];
00389         outArray[target + 2] = inArray[1 + ii] + inArray[5 + ii];
00390         outArray[target + 3] = -inArray[2 + ii] + inArray[4 + ii];
00391         outArray[target + 4] = inArray[3 + ii];
00392         outArray[target + 5] = inArray[ii];
00393     }
00394 }
00395
00396 /// Rotate halo and inner-patch data vectors with rotation matrix Ry into
00397 /// eigenspace of Z matrix and write to auxiliary vector
00398 inline void LatticePatch::rotateToY(sunrealtype *outArray,
00399                                     const unrealtype *inArray,
00400                                     const std::vector<sunindextype> &lookup) {
00401     sunindextype ii = 0, target = 0;
00402     const int dPD = envelopeLattice->get_dataPointDimension();
00403     const sunindextype size = lookup.size();
00404     #pragma omp parallel for simd \

```

```

00405  private(target, ii) \
00406  shared(lookup, outArray, inArray, size, dPD) \
00407  schedule(static)
00408  for (sunindextype i = 0; i < size; i++) {
00409    target = dPD * lookup[i];
00410    ii = dPD * i;
00411    outArray[target + 0] = inArray[ii] + inArray[5 + ii];
00412    outArray[target + 1] = -inArray[2 + ii] + inArray[3 + ii];
00413    outArray[target + 2] = -inArray[ii] + inArray[5 + ii];
00414    outArray[target + 3] = inArray[2 + ii] + inArray[3 + ii];
00415    outArray[target + 4] = inArray[4 + ii];
00416    outArray[target + 5] = inArray[1 + ii];
00417  }
00418 }
00419
00420 /// Rotate halo and inner-patch data vectors with rotation matrix Rz into
00421 /// eigenspace of Z matrix and write to auxiliary vector
00422 inline void LatticePatch::rotateToZ(sunrealtype *outArray,
00423                                     const unrealtype *inArray,
00424                                     const std::vector<sunindextype> &lookup) {
00425   sunindextype ii = 0, target = 0;
00426   const sunindextype size = lookup.size();
00427   const int dPD = envelopeLattice->get_dataPointDimension();
00428   #pragma omp parallel for simd \
00429   private(target, ii) \
00430   shared(lookup, outArray, inArray, size, dPD) \
00431   schedule(static)
00432   for (sunindextype i = 0; i < size; i++) {
00433     target = dPD * lookup[i];
00434     ii = dPD * i;
00435     outArray[target + 0] = -inArray[ii] + inArray[4 + ii];
00436     outArray[target + 1] = inArray[1 + ii] + inArray[3 + ii];
00437     outArray[target + 2] = inArray[ii] + inArray[4 + ii];
00438     outArray[target + 3] = -inArray[1 + ii] + inArray[3 + ii];
00439     outArray[target + 4] = inArray[5 + ii];
00440     outArray[target + 5] = inArray[2 + ii];
00441   }
00442 }
00443
00444 /// Derotate uAux with transposed rotation matrices and write to derivative
00445 /// buffer -- normalization is done here by the factor 1/2
00446 void LatticePatch::derotate(int dir, unrealtype *buffOut) {
00447   // Check that the lattice as well as the translocation lookups have been set
00448   // up;
00449   checkFlag(FLatticePatchSetUp);
00450   checkFlag(TranslocationLookupSetUp);
00451   const int dPD = envelopeLattice->get_dataPointDimension();
00452   const int gLW = envelopeLattice->get_ghostLayerWidth();
00453   const sunindextype totalNP = discreteSize();
00454   sunindextype ii = 0, target = 0;
00455   switch (dir) {
00456     case 1:
00457       #pragma omp parallel for simd \
00458       private(ii, target) \
00459       shared(dPD, gLW, totalNP, uTox, uAux, buffOut) \
00460       schedule(static)
00461       for (sunindextype i = 0; i < totalNP; i++) {
00462         // get correct indices in u and rotation space
00463         target = dPD * i;
00464         ii = dPD * (uTox[i] - gLW);
00465         buffOut[target + 0] = uAux[5 + ii];
00466         buffOut[target + 1] = (-uAux[ii] + uAux[2 + ii]) / 2.;
00467         buffOut[target + 2] = (uAux[1 + ii] - uAux[3 + ii]) / 2.;
00468         buffOut[target + 3] = uAux[4 + ii];
00469         buffOut[target + 4] = (uAux[1 + ii] + uAux[3 + ii]) / 2.;
00470         buffOut[target + 5] = (uAux[ii] + uAux[2 + ii]) / 2.;
00471       }
00472       break;
00473     case 2:
00474       #pragma omp parallel for simd \
00475       private(ii, target) \
00476       shared(dPD, gLW, totalNP, uTox, uAux, buffOut) \
00477       schedule(static)
00478       for (sunindextype i = 0; i < totalNP; i++) {
00479         target = dPD * i;
00480         ii = dPD * (uTox[i] - gLW);
00481         buffOut[target + 0] = (uAux[ii] - uAux[2 + ii]) / 2.;
00482         buffOut[target + 1] = uAux[5 + ii];
00483         buffOut[target + 2] = (-uAux[1 + ii] + uAux[3 + ii]) / 2.;
00484         buffOut[target + 3] = (uAux[1 + ii] + uAux[3 + ii]) / 2.;
00485         buffOut[target + 4] = uAux[4 + ii];
00486         buffOut[target + 5] = (uAux[ii] + uAux[2 + ii]) / 2.;
00487       }
00488       break;
00489     case 3:
00490       #pragma omp parallel for simd \
00491       private(ii, target) \

```

```

00492     shared(dPD, gLW, totalNP, uTox, uAux, buffOut) \
00493     schedule(static)
00494     for (sunindextype i = 0; i < totalNP; i++) {
00495         target = dPD * i;
00496         ii = dPD * (uToz[i] - gLW);
00497         buffOut[target + 0] = (-uAux[ii] + uAux[2 + ii]) / 2.;
00498         buffOut[target + 1] = (uAux[1 + ii] - uAux[3 + ii]) / 2.;
00499         buffOut[target + 2] = uAux[5 + ii];
00500         buffOut[target + 3] = (uAux[1 + ii] + uAux[3 + ii]) / 2.;
00501         buffOut[target + 4] = (uAux[ii] + uAux[2 + ii]) / 2.;
00502         buffOut[target + 5] = uAux[4 + ii];
00503     }
00504     break;
00505     default:
00506         errorKill("Tried to derotate from the wrong direction");
00507         break;
00508     }
00509 }
00510
00511 /// Create buffers to save derivative values, optimizing computational load
00512 void LatticePatch::initializeBuffers() {
00513     // Check that the lattice has been set up
00514     checkFlag(FLatticeDimensionSet);
00515     const int dPD = envelopeLattice->get_dataPointDimension();
00516     buffX.resize(nx * ny * nz * dPD);
00517     buffY.resize(nx * ny * nz * dPD);
00518     buffZ.resize(nx * ny * nz * dPD);
00519     // Set pointers used for propagation functions
00520     buffData[0] = &buffX[0];
00521     buffData[1] = &buffY[0];
00522     buffData[2] = &buffZ[0];
00523     statusFlags |= BuffersInitialized;
00524 }
00525
00526 /// Perform the ghost cell exchange in a specified direction
00527 void LatticePatch::exchangeGhostCells(const int dir) {
00528     // Check that the lattice has been set up
00529     checkFlag(FLatticeDimensionSet);
00530     checkFlag(FLatticePatchSetUp);
00531     // Variables to per dimension calculate the halo indices, and distance to
00532     // other side halo boundary
00533     int mx = 1, my = 1, mz = 1, distToLeft = 1;
00534     const int gLW = envelopeLattice->get_ghostLayerWidth();
00535     // In the chosen direction m is set to ghost layer width while the others
00536     // remain to form the plane
00537     switch (dir) {
00538     case 1:
00539         mx = gLW;
00540         my = ny;
00541         mz = nz;
00542         distToLeft = (nx - gLW);
00543         break;
00544     case 2:
00545         mx = nx;
00546         my = gLW;
00547         mz = nz;
00548         distToLeft = nx * (ny - gLW);
00549         break;
00550     case 3:
00551         mx = nx;
00552         my = ny;
00553         mz = gLW;
00554         distToLeft = nx * ny * (nz - gLW);
00555         break;
00556     }
00557     // total number of exchanged points
00558     const int dPD = envelopeLattice->get_dataPointDimension();
00559     const sunindextype exchangeSize = mx * my * mz * dPD;
00560     // provide size of the halos for ghost cells
00561     ghostCellLeft.resize(exchangeSize);
00562     ghostCellRight.resize(ghostCellLeft.size());
00563     ghostCellLeftToSend.resize(ghostCellLeft.size());
00564     ghostCellRightToSend.resize(ghostCellLeft.size());
00565     gCLData = &ghostCellLeft[0];
00566     gCRData = &ghostCellRight[0];
00567     statusFlags |= GhostLayersInitialized;
00568
00569     // Initialize running index li for the halo buffers, and index ui of uData for
00570     // data transfer
00571     sunindextype li = 0, ui = 0;
00572     // Fill the halo buffers
00573     #pragma omp parallel for default(None) \
00574     private(ui, li) \
00575     shared(nx, ny, mx, my, mz, dPD, distToLeft, uData, \
00576             ghostCellLeftToSend, ghostCellRightToSend)
00577     for (sunindextype iz = 0; iz < mz; iz++) {
00578         for (sunindextype iy = 0; iy < my; iy++) {

```

```

00579     // uData vector start index of halo data to be transferred
00580     // with each z-step add the whole xy-plane and with y-step the x-range ->
00581     // iterate all x-ranges
00582     ui = (iz * nx * ny + iy * nx) * dPD;
00583     // increase halo index by transferred items of previous iteration steps
00584     li = (iz * my * mx + iy * mx) * dPD;
00585     // copy left halo data from uData to buffer, transfer size is given by
00586     // x-length (not x-range)
00587     std::copy(&uData[ui], &uData[ui + mx * dPD], &ghostCellLeftToSend[li]);
00588     ui += distToRight * dPD;
00589     std::copy(&uData[ui], &uData[ui + mx * dPD], &ghostCellRightToSend[li]);
00590 }
00591 }
00592
00593 /* Send and receive the data to and from neighboring latticePatches */
00594 // Adjust direction to cartesian communicator
00595 int dim = 2; // default for dir==1
00596 if (dir == 2) {
00597     dim = 1;
00598 } else if (dir == 3) {
00599     dim = 0;
00600 }
00601 int rank_source = 0, rank_dest = 0;
00602 MPI_Cart_shift(envelopeLattice->comm, dim, -1, &rank_source,
00603                 &rank_dest); // s.t. rank_dest is left & v.v.
00604
00605 // nonblocking Irecv/Isend
00606
00607 MPI_Request requests[4];
00608 MPI_Irecv(&ghostCellRight[0], exchangeSize, MPI_SUNREALTYPE, rank_source, 1,
00609 envelopeLattice->comm, &requests[0]);
00610 MPI_Isend(&ghostCellLeftToSend[0], exchangeSize, MPI_SUNREALTYPE, rank_dest,
00611 1, envelopeLattice->comm, &requests[1]);
00612 MPI_Irecv(&ghostCellLeft[0], exchangeSize, MPI_SUNREALTYPE, rank_dest, 2,
00613 envelopeLattice->comm, &requests[2]);
00614 MPI_Isend(&ghostCellRightToSend[0], exchangeSize, MPI_SUNREALTYPE,
00615 rank_source, 2, envelopeLattice->comm, &requests[3]);
00616 MPI_Waitall(4, requests, MPI_STATUS_IGNORE);
00617
00618
00619 // blocking Sendrecv:
00620 /*
00621 MPI_Sendrecv(&ghostCellLeftToSend[0], exchangeSize, MPI_SUNREALTYPE,
00622             rank_dest, 1, &ghostCellRight[0], exchangeSize, MPI_SUNREALTYPE,
00623             rank_source, 1, envelopeLattice->comm, MPI_STATUS_IGNORE);
00624 MPI_Sendrecv(&ghostCellRightToSend[0], exchangeSize, MPI_SUNREALTYPE,
00625             rank_source, 2, &ghostCellLeft[0], exchangeSize, MPI_SUNREALTYPE,
00626             rank_dest, 2, envelopeLattice->comm, MPI_STATUS_IGNORE);
00627 */
00628 }
00629
00630 /// Check if all flags are set
00631 void LatticePatch::checkFlag(unsigned int flag) const {
00632     if (!(statusFlags & flag)) {
00633         std::string errorMessage;
00634         switch (flag) {
00635             case FLatticePatchSetUp:
00636                 errorMessage = "The Lattice patch was not set up please make sure to "
00637                             "initialize a Lattice topology";
00638                 break;
00639             case TranslocationLookupSetUp:
00640                 errorMessage = "The translocation lookup tables have not been generated, "
00641                             "please be sure to run generateTranslocationLookup()";
00642                 break;
00643             case GhostLayersInitialized:
00644                 errorMessage = "The space for the ghost layers has not been allocated, "
00645                             "please be sure that the ghost cells are initialized ";
00646                 break;
00647             case BuffersInitialized:
00648                 errorMessage = "The space for the buffers has not been allocated, please "
00649                             "be sure to run initializeBuffers()";
00650                 break;
00651             default:
00652                 errorMessage = "Uppss, you've made a non-standard error, sadly I can't "
00653                             "help you there";
00654                 break;
00655         }
00656         errorKill(errorMessage);
00657     }
00658     return;
00659 }
00660
00661 /// Calculate derivatives in the patch (uAux) in the specified direction
00662 void LatticePatch::derive(const int dir) {
00663     // ghost layer width adjusted to the chosen stencil order
00664     const int gLW = envelopeLattice->get_ghostLayerWidth();
00665     // dimensionality of data points -> 6

```

```

00666 const int dPD = envelopeLattice->get_dataPointDimension();
00667 // total width of patch in given direction including ghost layers at ends
00668 const sunindextype dirWidth = discreteSize(dir) + 2 * gLW;
00669 // width of patch only in given direction
00670 const sunindextype dirWidthO = discreteSize(dir);
00671 // size of plane perpendicular to given dimension
00672 const sunindextype perpPlainSize = discreteSize() / discreteSize(dir);
00673 // physical distance between points in that direction
00674 sunrealtype dxi = nan("0x12345");
00675 switch (dir) {
00676 case 1:
00677     dxi = dx;
00678     break;
00679 case 2:
00680     dxi = dy;
00681     break;
00682 case 3:
00683     dxi = dz;
00684     break;
00685 default:
00686     dxi = 1;
00687     errorKill("Tried to derive in the wrong direction");
00688     break;
00689 }
00690 // Derive according to chosen stencil accuracy order
00691 const int order = envelopeLattice->get_stencilOrder();
00692 switch (order) {
00693 case 1: // gLW=1
00694     #pragma omp parallel for default(none) \
00695     shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00696     for (sunindextype i = 0; i < perpPlainSize; i++) {
00697         #pragma omp simd
00698         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00699             j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00700             uAux[j + 0 - gLW * dPD] = slb(&uAux[j + 0]) / dxi;
00701             uAux[j + 1 - gLW * dPD] = slb(&uAux[j + 1]) / dxi;
00702             uAux[j + 2 - gLW * dPD] = slf(&uAux[j + 2]) / dxi;
00703             uAux[j + 3 - gLW * dPD] = slf(&uAux[j + 3]) / dxi;
00704             uAux[j + 4 - gLW * dPD] = slf(&uAux[j + 4]) / dxi;
00705             uAux[j + 5 - gLW * dPD] = slf(&uAux[j + 5]) / dxi;
00706         }
00707     }
00708     break;
00709 case 2: // gLW=2
00710     #pragma omp parallel for default(none) \
00711     shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00712     for (sunindextype i = 0; i < perpPlainSize; i++) {
00713         #pragma omp simd
00714         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00715             j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00716             uAux[j + 0 - gLW * dPD] = s2b(&uAux[j + 0]) / dxi;
00717             uAux[j + 1 - gLW * dPD] = s2b(&uAux[j + 1]) / dxi;
00718             uAux[j + 2 - gLW * dPD] = s2f(&uAux[j + 2]) / dxi;
00719             uAux[j + 3 - gLW * dPD] = s2f(&uAux[j + 3]) / dxi;
00720             uAux[j + 4 - gLW * dPD] = s2c(&uAux[j + 4]) / dxi;
00721             uAux[j + 5 - gLW * dPD] = s2c(&uAux[j + 5]) / dxi;
00722         }
00723     }
00724     break;
00725 case 3: // gLW=2
00726     #pragma omp parallel for default(none) \
00727     shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00728     for (sunindextype i = 0; i < perpPlainSize; i++) {
00729         #pragma omp simd
00730         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00731             j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00732             uAux[j + 0 - gLW * dPD] = s3b(&uAux[j + 0]) / dxi;
00733             uAux[j + 1 - gLW * dPD] = s3b(&uAux[j + 1]) / dxi;
00734             uAux[j + 2 - gLW * dPD] = s3f(&uAux[j + 2]) / dxi;
00735             uAux[j + 3 - gLW * dPD] = s3f(&uAux[j + 3]) / dxi;
00736             uAux[j + 4 - gLW * dPD] = s3f(&uAux[j + 4]) / dxi;
00737             uAux[j + 5 - gLW * dPD] = s3f(&uAux[j + 5]) / dxi;
00738         }
00739     }
00740     break;
00741 case 4: // gLW=3
00742     #pragma omp parallel for default(none) \
00743     shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00744     for (sunindextype i = 0; i < perpPlainSize; i++) {
00745         #pragma omp simd
00746         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00747             j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00748             uAux[j + 0 - gLW * dPD] = s4b(&uAux[j + 0]) / dxi;
00749             uAux[j + 1 - gLW * dPD] = s4b(&uAux[j + 1]) / dxi;
00750             uAux[j + 2 - gLW * dPD] = s4f(&uAux[j + 2]) / dxi;
00751             uAux[j + 3 - gLW * dPD] = s4f(&uAux[j + 3]) / dxi;
00752             uAux[j + 4 - gLW * dPD] = s4c(&uAux[j + 4]) / dxi;

```

```

00753         uAux[j + 5 - gLW * dPD] = s4c(&uAux[j + 5]) / dx;
00754     }
00755 }
00756 break;
00757 case 5: // gLW=3
00758 #pragma omp parallel for default(none) \
00759 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00760 for (sunindextype i = 0; i < perpPlainSize; i++) {
00761     #pragma omp simd
00762     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00763         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00764         uAux[j + 0 - gLW * dPD] = s5b(&uAux[j + 0]) / dx;
00765         uAux[j + 1 - gLW * dPD] = s5b(&uAux[j + 1]) / dx;
00766         uAux[j + 2 - gLW * dPD] = s5f(&uAux[j + 2]) / dx;
00767         uAux[j + 3 - gLW * dPD] = s5f(&uAux[j + 3]) / dx;
00768         uAux[j + 4 - gLW * dPD] = s5f(&uAux[j + 4]) / dx;
00769         uAux[j + 5 - gLW * dPD] = s5f(&uAux[j + 5]) / dx;
00770     }
00771 }
00772 break;
00773 case 6: // gLW=4
00774 #pragma omp parallel for default(none) \
00775 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00776 for (sunindextype i = 0; i < perpPlainSize; i++) {
00777     #pragma omp simd
00778     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00779         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00780         uAux[j + 0 - gLW * dPD] = s6b(&uAux[j + 0]) / dx;
00781         uAux[j + 1 - gLW * dPD] = s6b(&uAux[j + 1]) / dx;
00782         uAux[j + 2 - gLW * dPD] = s6f(&uAux[j + 2]) / dx;
00783         uAux[j + 3 - gLW * dPD] = s6f(&uAux[j + 3]) / dx;
00784         uAux[j + 4 - gLW * dPD] = s6c(&uAux[j + 4]) / dx;
00785         uAux[j + 5 - gLW * dPD] = s6c(&uAux[j + 5]) / dx;
00786     }
00787 }
00788 break;
00789 case 7: // gLW=4
00790 #pragma omp parallel for default(none) \
00791 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00792 for (sunindextype i = 0; i < perpPlainSize; i++) {
00793     #pragma omp simd
00794     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00795         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00796         uAux[j + 0 - gLW * dPD] = s7b(&uAux[j + 0]) / dx;
00797         uAux[j + 1 - gLW * dPD] = s7b(&uAux[j + 1]) / dx;
00798         uAux[j + 2 - gLW * dPD] = s7f(&uAux[j + 2]) / dx;
00799         uAux[j + 3 - gLW * dPD] = s7f(&uAux[j + 3]) / dx;
00800         uAux[j + 4 - gLW * dPD] = s7f(&uAux[j + 4]) / dx;
00801         uAux[j + 5 - gLW * dPD] = s7f(&uAux[j + 5]) / dx;
00802     }
00803 }
00804 break;
00805 case 8: // gLW=5
00806 #pragma omp parallel for default(none) \
00807 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00808 for (sunindextype i = 0; i < perpPlainSize; i++) {
00809     #pragma omp simd
00810     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00811         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00812         uAux[j + 0 - gLW * dPD] = s8b(&uAux[j + 0]) / dx;
00813         uAux[j + 1 - gLW * dPD] = s8b(&uAux[j + 1]) / dx;
00814         uAux[j + 2 - gLW * dPD] = s8f(&uAux[j + 2]) / dx;
00815         uAux[j + 3 - gLW * dPD] = s8f(&uAux[j + 3]) / dx;
00816         uAux[j + 4 - gLW * dPD] = s8c(&uAux[j + 4]) / dx;
00817         uAux[j + 5 - gLW * dPD] = s8c(&uAux[j + 5]) / dx;
00818     }
00819 }
00820 break;
00821 case 9: // gLW=5
00822 #pragma omp parallel for default(none) \
00823 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)
00824 for (sunindextype i = 0; i < perpPlainSize; i++) {
00825     #pragma omp simd
00826     for (sunindextype j = (i * dirWidth + gLW) * dPD;
00827         j < (i * dirWidth + gLW + dirWidthO) * dPD; j += dPD) {
00828         uAux[j + 0 - gLW * dPD] = s9b(&uAux[j + 0]) / dx;
00829         uAux[j + 1 - gLW * dPD] = s9b(&uAux[j + 1]) / dx;
00830         uAux[j + 2 - gLW * dPD] = s9f(&uAux[j + 2]) / dx;
00831         uAux[j + 3 - gLW * dPD] = s9f(&uAux[j + 3]) / dx;
00832         uAux[j + 4 - gLW * dPD] = s9f(&uAux[j + 4]) / dx;
00833         uAux[j + 5 - gLW * dPD] = s9f(&uAux[j + 5]) / dx;
00834     }
00835 }
00836 break;
00837 case 10: // gLW=6
00838 #pragma omp parallel for default(none) \
00839 shared(perpPlainSize, dxi, dirWidth, dirWidthO, gLW, dPD, uAux)

```

```

00840     for (sunindextype i = 0; i < perpPlainSize; i++) {
00841         #pragma omp simd
00842         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00843             j < (i * dirWidth + gLW + dirWidth0) * dPD; j += dPD) {
00844             uAux[j + 0 - gLW * dPD] = s10b(&uAux[j + 0]) / dxi;
00845             uAux[j + 1 - gLW * dPD] = s10b(&uAux[j + 1]) / dxi;
00846             uAux[j + 2 - gLW * dPD] = s10f(&uAux[j + 2]) / dxi;
00847             uAux[j + 3 - gLW * dPD] = s10f(&uAux[j + 3]) / dxi;
00848             uAux[j + 4 - gLW * dPD] = s10c(&uAux[j + 4]) / dxi;
00849             uAux[j + 5 - gLW * dPD] = s10c(&uAux[j + 5]) / dxi;
00850         }
00851     }
00852     break;
00853 case 11: // gLW=6
00854     #pragma omp parallel for default(none) \
00855     shared(perpPlainSize, dxi, dirWidth, dirWidth0, gLW, dPD, uAux)
00856     for (sunindextype i = 0; i < perpPlainSize; i++) {
00857         #pragma omp simd
00858         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00859             j < (i * dirWidth + gLW + dirWidth0) * dPD; j += dPD) {
00860             uAux[j + 0 - gLW * dPD] = s11b(&uAux[j + 0]) / dxi;
00861             uAux[j + 1 - gLW * dPD] = s11b(&uAux[j + 1]) / dxi;
00862             uAux[j + 2 - gLW * dPD] = s11f(&uAux[j + 2]) / dxi;
00863             uAux[j + 3 - gLW * dPD] = s11f(&uAux[j + 3]) / dxi;
00864             uAux[j + 4 - gLW * dPD] = s11f(&uAux[j + 4]) / dxi;
00865             uAux[j + 5 - gLW * dPD] = s11f(&uAux[j + 5]) / dxi;
00866         }
00867     }
00868     break;
00869 case 12: // gLW=7
00870     #pragma omp parallel for default(none) \
00871     shared(perpPlainSize, dxi, dirWidth, dirWidth0, gLW, dPD, uAux)
00872     for (sunindextype i = 0; i < perpPlainSize; i++) {
00873         #pragma omp simd
00874         for (sunindextype j = (i * dirWidth + gLW) * dPD;
00875             j < (i * dirWidth + gLW + dirWidth0) * dPD; j += dPD) {
00876             uAux[j + 0 - gLW * dPD] = s12b(&uAux[j + 0]) / dxi;
00877             uAux[j + 1 - gLW * dPD] = s12b(&uAux[j + 1]) / dxi;
00878             uAux[j + 2 - gLW * dPD] = s12f(&uAux[j + 2]) / dxi;
00879             uAux[j + 3 - gLW * dPD] = s12f(&uAux[j + 3]) / dxi;
00880             uAux[j + 4 - gLW * dPD] = s12c(&uAux[j + 4]) / dxi;
00881             uAux[j + 5 - gLW * dPD] = s12c(&uAux[j + 5]) / dxi;
00882         }
00883     }
00884     break;
00885 case 13: // gLW=7
00886     // For all points in the plane perpendicular to the given direction
00887     #pragma omp parallel for default(none) \
00888     shared(perpPlainSize, dxi, dirWidth, dirWidth0, gLW, dPD, uAux)
00889     for (sunindextype i = 0; i < perpPlainSize; i++) {
00890         // iterate through the derivation direction
00891         #pragma omp simd
00892         for (sunindextype j = (i * dirWidth
00893             + gLW /*to shift left by gLW below*/) * dPD;
00894             j < (i * dirWidth + gLW + dirWidth0) * dPD; j += dPD) {
00895             // Compute the stencil derivative for any of the six field components
00896             // and update position by ghost width shift
00897             uAux[j + 0 - gLW * dPD] = s13b(&uAux[j + 0]) / dxi;
00898             uAux[j + 1 - gLW * dPD] = s13b(&uAux[j + 1]) / dxi;
00899             uAux[j + 2 - gLW * dPD] = s13f(&uAux[j + 2]) / dxi;
00900             uAux[j + 3 - gLW * dPD] = s13f(&uAux[j + 3]) / dxi;
00901             uAux[j + 4 - gLW * dPD] = s13f(&uAux[j + 4]) / dxi;
00902             uAux[j + 5 - gLW * dPD] = s13f(&uAux[j + 5]) / dxi;
00903         }
00904     }
00905     break;
00906
00907 default:
00908     errorKill("Please set an existing stencil order");
00909     break;
00910 }
00911 }
00912
00913 ////////////// Helper functions ///////////
00914
00915 /// Print a specific error message to stderr
00916 void errorKill(const std::string & errorMessage) {
00917     int my_prc=0;
00918     MPI_Comm_rank(MPI_COMM_WORLD,&my_prc);
00919     if (my_prc==0) {
00920         std::cerr << std::endl << "Error: " << errorMessage
00921         << "\nAborting..." << std::endl;
00922         MPI_Abort(MPI_COMM_WORLD, 1);
00923         return;
00924     }
00925 }
00926

```

```

00927 /** Check MPI errors. Error handler must be set. */
00928 int check_error(int error, const char *funcname, int id) {
00929     int eclass, len;
00930     char errorstring[MPI_MAX_ERROR_STRING];
00931     if( error != MPI_SUCCESS ) {
00932         MPI_Error_class(error,&eclass);
00933         MPI_Error_string(error,errorstring,&len);
00934         std::cerr << "MPI Error(process " << id << ") in " << funcname << " : "
00935             << errorstring << ", from class " << eclass << std::endl;
00936         return 1;
00937     }
00938     return 0;
00939 }
00940
00941 /** Check function return value. Adapted from CVode examples.
00942     opt == 0 means SUNDIALS function allocates memory so check if
00943         returned NULL pointer
00944     opt == 1 means SUNDIALS function returns an integer value so check if
00945         retval < 0
00946     opt == 2 means function allocates memory so check if returned
00947         NULL pointer */
00948 int check_retval(void *returnvalue, const char *funcname, int opt, int id) {
00949     int *retval = nullptr;
00950
00951     /* Check if SUNDIALS function returned NULL pointer - no memory allocated */
00952     if (opt == 0 && returnvalue == nullptr) {
00953         fprintf(stderr,
00954             "\nSUNDIALS_ERROR(%d): %s() failed - returned NULL pointer\n\n", id,
00955             funcname);
00956         return (1);
00957     }
00958
00959     /* Check if retval < 0 */
00960     else if (opt == 1) {
00961         retval = (int *)returnvalue;
00962         char *flagname = CVodeGetReturnFlagName(*retval);
00963         if (*retval < 0) {
00964             fprintf(stderr, "\nSUNDIALS_ERROR(%d): %s() failed with retval = %d: "
00965                 "%s\n\n",
00966                 id, funcname, *retval, flagname);
00967             return (1);
00968         }
00969     }
00970
00971     /* Check if function returned NULL pointer - no memory allocated */
00972     else if (opt == 2 && returnvalue == nullptr) {
00973         fprintf(stderr,
00974             "\nMEMORY_ERROR(%d): %s() failed - returned NULL pointer\n\n", id,
00975             funcname);
00976         return (1);
00977     }
00978
00979     return (0);
00980 }

```

6.42 src/LatticePatch.h File Reference

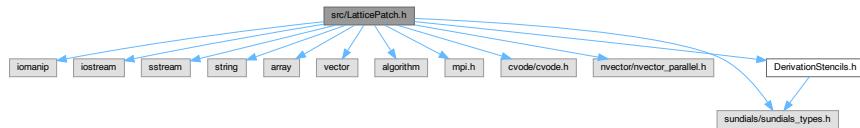
Declaration of the lattice and lattice patches.

```

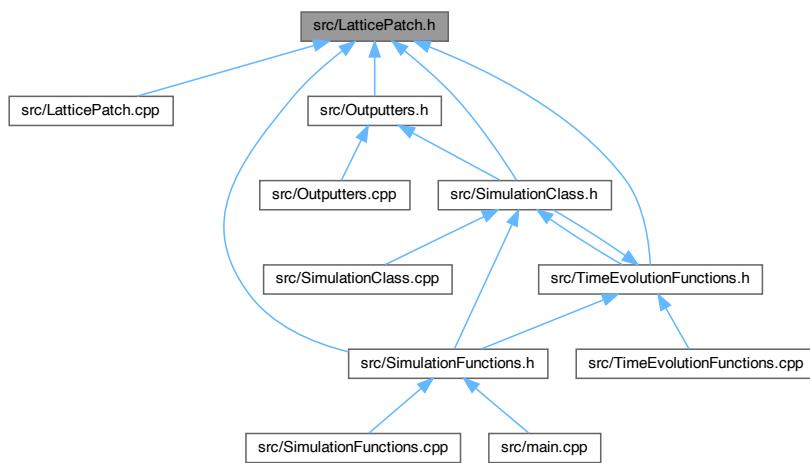
#include <iomanip>
#include <iostream>
#include <sstream>
#include <string>
#include <array>
#include <vector>
#include <algorithm>
#include <mpi.h>
#include <cvode/cvode.h>
#include <nvector/nvector_parallel.h>
#include <sundials/sundials_types.h>

```

```
#include "DerivationStencils.h"
Include dependency graph for LatticePatch.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Lattice](#)
Lattice class for the construction of the enveloping discrete simulation space.
- class [LatticePatch](#)
LatticePatch class for the construction of the patches in the enveloping lattice.

Functions

- void [errorKill](#) (const std::string &errorMessage)
helper function for error messages
- int [check_error](#) (int error, const char *funcname, int id)
helper function to check MPI errors
- int [check_retval](#) (void *returnvalue, const char *funcname, int opt, int id)
helper function to check CVode errors

Variables

- `constexpr unsigned int FLatticeDimensionSet = 0x01`
lattice construction checking flag
- `constexpr unsigned int FLatticePatchSetUp = 0x01`
- `constexpr unsigned int TranslocationLookupSetUp = 0x02`
- `constexpr unsigned int GhostLayersInitialized = 0x04`
- `constexpr unsigned int BuffersInitialized = 0x08`

6.42.1 Detailed Description

Declaration of the lattice and lattice patches.

Definition in file [LatticePatch.h](#).

6.42.2 Function Documentation

6.42.2.1 check_error()

```
int check_error (
    int error,
    const char * funcname,
    int id )
```

helper function to check MPI errors

Check MPI errors. Error handler must be set.

Definition at line 928 of file [LatticePatch.cpp](#).

```
00928
00929     int eclass, len;
00930     char errorstring[MPI_MAX_ERROR_STRING];
00931     if( error != MPI_SUCCESS ) {
00932         MPI_Error_class(error,&eclass);
00933         MPI_Error_string(error,errorstring,&len);
00934         std::cerr << "MPI Error(process " << id << ") in " << funcname << " : "
00935             << errorstring << ", from class " << eclass << std::endl;
00936         return 1;
00937     }
00938     return 0;
00939 }
```

6.42.2.2 check_retval()

```
int check_retval (
    void * returnvalue,
    const char * funcname,
    int opt,
    int id )
```

helper function to check CVode errors

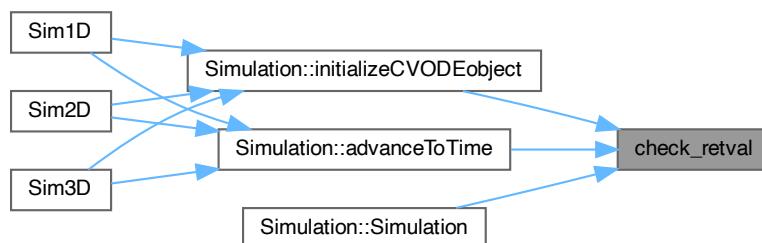
Check function return value. Adapted from CVode examples. opt == 0 means SUNDIALS function allocates memory so check if returned NULL pointer opt == 1 means SUNDIALS function returns an integer value so check if retval < 0 opt == 2 means function allocates memory so check if returned NULL pointer

Definition at line 948 of file [LatticePatch.cpp](#).

```
00948     int *retval = nullptr;
00949
00950
00951     /* Check if SUNDIALS function returned NULL pointer - no memory allocated */
00952     if (opt == 0 && returnvalue == nullptr) {
00953         fprintf(stderr,
00954             "\nSUNDIALS_ERROR(%d): %s() failed - returned NULL pointer\n\n", id,
00955             funcname);
00956         return (1);
00957     }
00958
00959     /* Check if retval < 0 */
00960     else if (opt == 1) {
00961         retval = (int *)returnvalue;
00962         char *flagname = CVodeGetReturnFlagName(*retval);
00963         if (*retval < 0) {
00964             fprintf(stderr, "\nSUNDIALS_ERROR(%d): %s() failed with retval = %d: "
00965                 "%s\n\n",
00966                 id, funcname, *retval, flagname);
00967             return (1);
00968         }
00969     }
00970
00971     /* Check if function returned NULL pointer - no memory allocated */
00972     else if (opt == 2 && returnvalue == nullptr) {
00973         fprintf(stderr,
00974             "\nMEMORY_ERROR(%d): %s() failed - returned NULL pointer\n\n", id,
00975             funcname);
00976         return (1);
00977     }
00978
00979     return (0);
00980 }
```

Referenced by [Simulation::advanceToTime\(\)](#), [Simulation::initializeCVODEobject\(\)](#), and [Simulation::Simulation\(\)](#).

Here is the caller graph for this function:



6.42.2.3 errorKill()

```
void errorKill (
    const std::string & errorMessage )
```

helper function for error messages

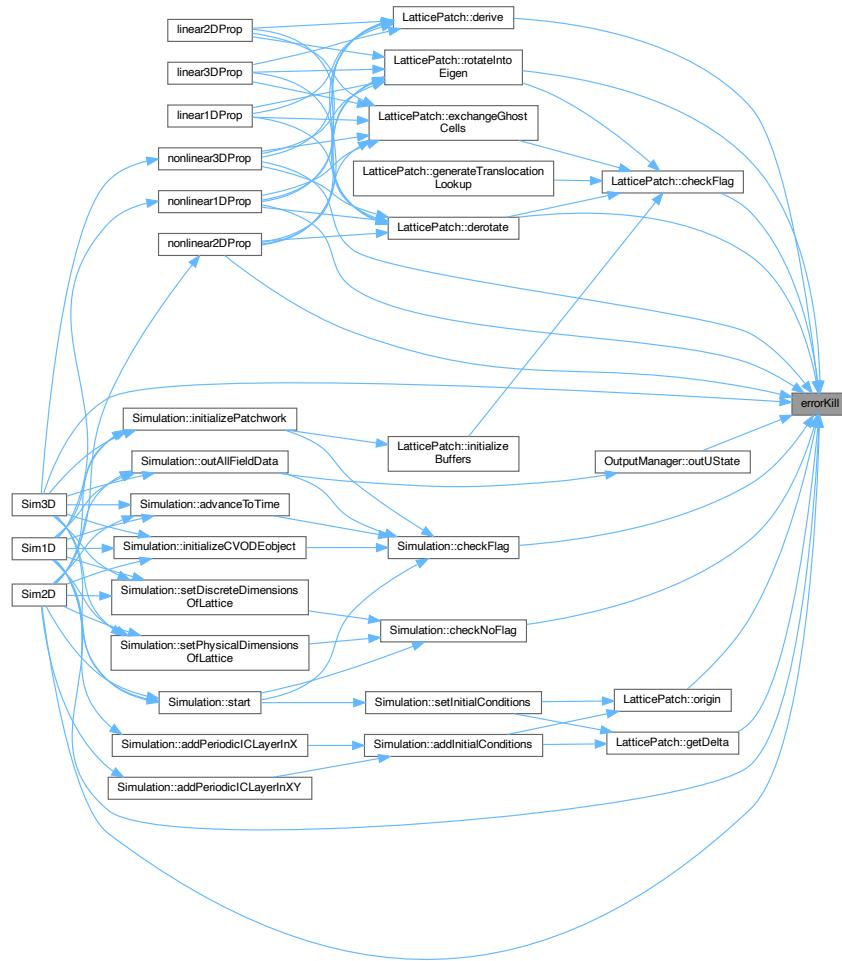
helper function for error messages

Definition at line 916 of file [LatticePatch.cpp](#).

```
00916
00917     int my_prc=0;
00918     MPI_Comm_rank(MPI_COMM_WORLD,&my_prc);
00919     if (my_prc==0) {
00920         std::cerr << std::endl << "Error: " << errorMessage
00921         << "\nAborting..." << std::endl;
00922         MPI_Abort(MPI_COMM_WORLD, 1);
00923         return;
00924     }
00925 }
```

Referenced by [LatticePatch::checkFlag\(\)](#), [Simulation::checkFlag\(\)](#), [Simulation::checkNoFlag\(\)](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::getDelta\(\)](#), [nonlinear1DProp\(\)](#), [nonlinear2DProp\(\)](#), [nonlinear3DProp\(\)](#), [LatticePatch::origin\(\)](#), [OutputManager::outUState\(\)](#), [LatticePatch::rotateIntoEigen\(\)](#), [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the caller graph for this function:



6.42.3 Variable Documentation

6.42.3.1 BuffersInitialized

```
constexpr unsigned int BuffersInitialized = 0x08 [constexpr]
```

lattice patch construction checking flag

Definition at line 47 of file [LatticePatch.h](#).

Referenced by [LatticePatch::checkFlag\(\)](#), and [LatticePatch::initializeBuffers\(\)](#).

6.42.3.2 FLatticeDimensionSet

```
constexpr unsigned int FLatticeDimensionSet = 0x01 [constexpr]
```

lattice construction checking flag

Definition at line 40 of file [LatticePatch.h](#).

Referenced by [LatticePatch::exchangeGhostCells\(\)](#), [LatticePatch::generateTranslocationLookup\(\)](#), [LatticePatch::initializeBuffers\(\)](#), and [Lattice::setPhysicalDimensions\(\)](#).

6.42.3.3 FLatticePatchSetUp

```
constexpr unsigned int FLatticePatchSetUp = 0x01 [constexpr]
```

lattice patch construction checking flag

Definition at line 44 of file [LatticePatch.h](#).

Referenced by [LatticePatch::checkFlag\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), [LatticePatch::rotateIntoEigen\(\)](#), and [LatticePatch::~LatticePatch\(\)](#).

6.42.3.4 GhostLayersInitialized

```
constexpr unsigned int GhostLayersInitialized = 0x04 [constexpr]
```

lattice patch construction checking flag

Definition at line 46 of file [LatticePatch.h](#).

Referenced by [LatticePatch::checkFlag\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

6.42.3.5 TranslocationLookupSetUp

```
constexpr unsigned int TranslocationLookupSetUp = 0x02 [constexpr]
```

lattice patch construction checking flag

Definition at line 45 of file [LatticePatch.h](#).

Referenced by [LatticePatch::checkFlag\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::generateTranslocationLookup\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

6.43 LatticePatch.h

[Go to the documentation of this file.](#)

```
00001 ///////////////////////////////////////////////////////////////////
00002 /// @file LatticePatch.h
00003 /// @brief Declaration of the lattice and lattice patches
00004 ///////////////////////////////////////////////////////////////////
00005
00006 #pragma once
00007
00008 // IO
00009 #include <iomanip>
00010 #include <iostream>
00011 #include <sstream>
00012
00013 // string, container, algorithm
00014 #include <string>
00015 // #include <string_view>
00016 #include <array>
00017 #include <vector>
00018 #include <algorithm>
00019
00020 // MPI & OpenMP
00021 #include <mpi.h>
00022 #if defined(_OPENMP)
00023 #include <omp.h>
00024 #endif
00025
00026 // Sundials
00027 #include <cvode/cvode.h>           /* prototypes for CVODE fcts. */
00028 #if defined(_OPENMP)
00029 #include <nvector/nvector_openmp.h> /* definition of OpenMP N_Vector */
00030 #include <nvector/nvector_mpiplusx.h> /* definition of MPI+X N_Vector */
00031 #else
00032 #include <nvector/nvector_parallel.h> /* definition of MPI N_Vector */
00033 #endif
00034 #include <sundials/sundials_types.h> /* definition of type sunrealtype */
00035
00036 // stencils
00037 #include "DerivationStencils.h"
00038
00039 /// lattice construction checking flag
00040 constexpr unsigned int FLatticeDimensionSet = 0x01;
00041
00042 /**
00043 ** lattice patch construction checking flag */
00044 constexpr unsigned int FLatticePatchSetUp = 0x01;
00045 constexpr unsigned int TranslocationLookupSetUp = 0x02;
00046 constexpr unsigned int GhostLayersInitialized = 0x04;
00047 constexpr unsigned int BuffersInitialized = 0x08;
00048 /**
00049
00050 /** @brief Lattice class for the construction of the enveloping discrete
00051 * simulation space */
00052 class Lattice {
00053 private:
00054     /// physical size of the lattice in x-direction
00055     sunrealtype tot_lx;
00056     /// physical size of the lattice in y-direction
00057     sunrealtype tot_ly;
00058     /// physical size of the lattice in z-direction
00059     sunrealtype tot_lz;
00060     /// number of points in x-direction
00061     sunindextype tot_nx;
00062     /// number of points in y-direction
```

```

00063     sunindextype tot_ny;
00064     /// number of points in z-direction
00065     sunindextype tot_nz;
00066     /// total number of lattice points
00067     sunindextype tot_noP;
00068     /// dimension of each data point set once and for all
00069     static constexpr int dataPointDimension = 6;
00070     /// number of lattice points times data dimension of each point
00071     sunindextype tot_noDP;
00072     /// physical distance between lattice points in x-direction
00073     sunrealtype dx;
00074     /// physical distance between lattice points in y-direction
00075     sunrealtype dy;
00076     /// physical distance between lattice points in z-direction
00077     sunrealtype dz;
00078     /// stencil order
00079     const int stencilOrder;
00080     /// required width of ghost layers (depends on the stencil order)
00081     const int ghostLayerWidth;
00082     /// lattice status flags
00083     unsigned int statusFlags;
00084
00085 public:
00086     /// number of MPI processes
00087     int n_prc;
00088     /// number of MPI process
00089     int my_prc;
00090     /// personal communicator of the lattice
00091     MPI_Comm comm;
00092     /// function to create and deploy the cartesian communicator
00093     void initializeCommunicator(const int Nx, const int Ny,
00094         const int Nz, const bool per);
00095     /// default construction
00096     Lattice(const int St0);
00097     /// SUNContext object
00098     SUNContext sunctx;
00099     /// component function for resizing the discrete dimensions of the lattice
00100     void setDiscreteDimensions(const sunindextype _nx,
00101         const sunindextype _ny, const sunindextype _nz);
00102     /// component function for resizing the physical size of the lattice
00103     void setPhysicalDimensions(const sunrealtype _lx,
00104         const sunrealtype _ly, const sunrealtype _lz);
00105     // /**
00106     /** @brief getter function */
00107     [[nodiscard]] const sunrealtype &get_tot_lx() const { return tot_lx; }
00108     [[nodiscard]] const sunrealtype &get_tot_ly() const { return tot_ly; }
00109     [[nodiscard]] const sunrealtype &get_tot_lz() const { return tot_lz; }
00110     [[nodiscard]] const sunindextype &get_tot_nx() const { return tot_nx; }
00111     [[nodiscard]] const sunindextype &get_tot_ny() const { return tot_ny; }
00112     [[nodiscard]] const sunindextype &get_tot_nz() const { return tot_nz; }
00113     [[nodiscard]] const sunindextype &get_tot_noP() const { return tot_noP; }
00114     [[nodiscard]] const sunrealtype &get_dx() const { return dx; }
00115     [[nodiscard]] const sunrealtype &get_dy() const { return dy; }
00116     [[nodiscard]] const sunrealtype &get_dz() const { return dz; }
00117     [[nodiscard]] constexpr int get_dataPointDimension() const {
00118         return dataPointDimension;
00119     }
00120 }
00121 [[nodiscard]] const int &get_stencilOrder() const { return stencilOrder; }
00122 [[nodiscard]] const int &get_ghostLayerWidth() const {
00123     return ghostLayerWidth;
00124 }
00125 // /**
00126 };
00127
00128 /** @brief LatticePatch class for the construction of the patches in the
00129 * enveloping lattice */
00130 class LatticePatch {
00131 private:
00132     /// origin of the patch in physical space; x-coordinate
00133     sunrealtype x0;
00134     /// origin of the patch in physical space; y-coordinate
00135     sunrealtype y0;
00136     /// origin of the patch in physical space; z-coordinate
00137     sunrealtype z0;
00138     /// inner position of lattice-patch in the lattice patchwork; x-points
00139     sunindextype LIx;
00140     /// inner position of lattice-patch in the lattice patchwork; y-points
00141     sunindextype LIy;
00142     /// inner position of lattice-patch in the lattice patchwork; z-points
00143     sunindextype LIz;
00144     /// physical size of the lattice-patch in the x-dimension
00145     sunrealtype lx;
00146     /// physical size of the lattice-patch in the y-dimension
00147     sunrealtype ly;
00148     /// physical size of the lattice-patch in the z-dimension
00149     sunrealtype lz;

```

```

00150  /// number of points in the lattice patch in the x-dimension
00151  sunindextype nx;
00152  /// number of points in the lattice patch in the y-dimension
00153  sunindextype ny;
00154  /// number of points in the lattice patch in the z-dimension
00155  sunindextype nz;
00156  /// physical distance between lattice points in x-direction
00157  sunrealtype dx;
00158  /// physical distance between lattice points in y-direction
00159  sunrealtype dy;
00160  /// physical distance between lattice points in z-direction
00161  sunrealtype dz;
00162  /// lattice patch status flags
00163  unsigned int statusFlags;
00164  /// pointer to the enveloping lattice
00165  const Lattice *envelopeLattice;
00166  /// aid (auxilliarily) vector including ghost cells to compute the derivatives
00167  std::vector<sunrealtype> uAux;
00168  ///////////////////////////////////////////////////
00169  /** translocation lookup table */
00170  std::vector<sunindextype> uTox, uToy, uToz, xTou, yTou, zTou;
00171  ///////////////////////////////////////////////////
00172  ///////////////////////////////////////////////////
00173  /** buffer to save spatial derivative values */
00174  std::vector<sunrealtype> buffX, buffY, buffZ;
00175  ///////////////////////////////////////////////////
00176  ///////////////////////////////////////////////////
00177  /** buffer for passing ghost cell data */
00178  std::vector<sunrealtype> ghostCellLeft, ghostCellRight, ghostCellLeftToSend,
00179  ghostCellRightToSend, ghostCellsToSend, ghostCells;
00180  ///////////////////////////////////////////////////
00181  ///////////////////////////////////////////////////
00182  /** ghost cell translocation lookup table */
00183  std::vector<sunindextype> lgcTox, rgcTox, lgcToy, rgcToy, lgcToz, rgcToz;
00184  ///////////////////////////////////////////////////
00185  ///////////////////////////////////////////////////
00186  /** Rotate and translocate an input array according to a lookup into an output
00187  * array */
00188  inline void rotateToX(sunrealtype *outArray, const unrealtype *inArray,
00189  const std::vector<sunindextype> &lookup);
00190  inline void rotateToY(sunrealtype *outArray, const unrealtype *inArray,
00191  const std::vector<sunindextype> &lookup);
00192  inline void rotateToZ(sunrealtype *outArray, const unrealtype *inArray,
00193  const std::vector<sunindextype> &lookup);
00194  ///////////////////////////////////////////////////
00195  public:
00196  /// ID of the LatticePatch, corresponds to process number (for debugging)
00197  int ID;
00198  /// NVector for saving field components u=(E,B) in lattice points
00199  N_Vector uLocal, u;
00200  /// NVector for saving temporal derivatives of the field data
00201  N_Vector duLocal, du;
00202  /// pointer to field data
00203  unrealtype *uData;
00204  /// pointer to time-derivative data
00205  unrealtype *duData;
00206  /// pointer to auxiliary data vector
00207  unrealtype *uAuxData;
00208  ///////////////////////////////////////////////////
00209  /** pointer to halo data */
00210  unrealtype *gCLData, *gCRData;
00211  ///////////////////////////////////////////////////
00212  /** pointer to spatial derivative data buffers
00213  std::array<sunrealtype *, 3> buffData;
00214  /// constructor setting up a default first lattice patch
00215  LatticePatch();
00216  /// destructor freeing parallel vectors
00217  ~LatticePatch();
00218  /// friend function for creating the patchwork slicing of the overall lattice
00219  friend int generatePatchwork(const Lattice &envelopeLattice,
00220  LatticePatch &patchToMold, const int DLx,
00221  const int DLy, const int DLz);
00222  /// function to get the discrete size of the LatticePatch
00223  sunindextype discreteSize(int dir=0) const;
00224  /// function to get the origin of the patch
00225  unrealtype origin(const int dir) const;
00226  /// function to get distance between points
00227  unrealtype getDelta(const int dir) const;
00228  /// function to fill out the lookup tables for cache efficiency
00229  void generateTranslocationLookup();
00230  /// function to rotate u into Z-matrix eigenraum
00231  void rotateIntoEigen(const int dir);
00232  /// function to derotate uAux into dudata lattice direction of x
00233  void derotate(int dir, unrealtype *buffOut);
00234  /// initialize buffers to save derivatives
00235  void initializeBuffers();
00236  /// function to exchange ghost cells

```

```

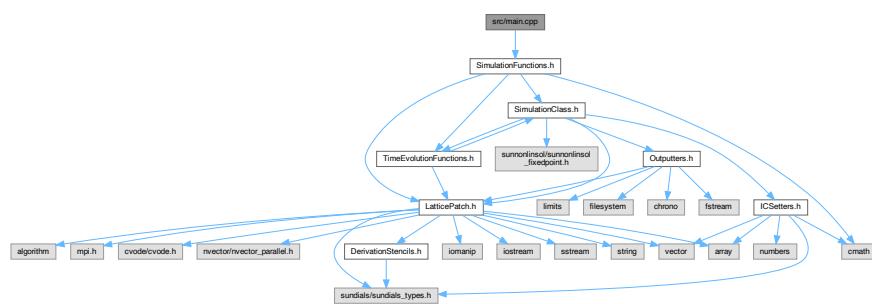
00237 void exchangeGhostCells(const int dir);
00238 /// function to derive the centered values in uAux and save them noncentered
00239 void derive(const int dir);
00240 /// function to check if a flag has been set and if not abort
00241 void checkFlag(unsigned int flag) const;
00242 };
00243
00244 /// helper function for error messages
00245 void errorKill(const std::string & errorMessage);
00246
00247 /// helper function to check MPI errors
00248 int check_error(int error, const char *funcname, int id);
00249
00250 /// helper function to check CVode errors
00251 int check_retval(void *returnvalue, const char *funcname, int opt, int id);
00252

```

6.44 src/main.cpp File Reference

Main function to configure the user's simulation settings.

```
#include "SimulationFunctions.h"
Include dependency graph for main.cpp:
```



Functions

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

6.44.1 Detailed Description

Main function to configure the user's simulation settings.

Definition in file [main.cpp](#).

6.44.2 Function Documentation

6.44.2.1 main()

```
int main (
    int argc,
    char * argv[] )
```

Determine the output directory.

A "SimResults" folder will be created if non-existent with a subdirectory named in the identifier format "yy-mm-dd_hh-MM-ss" that contains the csv files

A 1D simulation with specified

A 2D simulation with specified

A 3D simulation with specified

Definition at line 9 of file [main.cpp](#).

```
00010 {
00011     /** Determine the output directory.
00012      * A "SimResults" folder will be created if non-existent
00013      * with a subdirectory named in the identifier format
00014      * "yy-mm-dd_hh-MM-ss" that contains the csv files      */
00015     constexpr auto outputDirectory = "/path/to/directory/";
00016
00017     if(!filesystem::exists(outputDirectory)) {
00018         cerr<<"\nOutput directory nonexistent.\n";
00019         exit(1);
00020     }
00021
00022     // Initialize MPI environment
00023     int provided;
00024     MPI_Init_thread(&argc, &argv, MPI_THREAD_SINGLE, &provided);
00025
00026
00027     //----- BEGIN OF CONFIGURATION -----//
00028
00029     ////////////// -- 1D -- ///////////////////
00030     /** A 1D simulation with specified */
00031     /*
00032     // Specify your settings here //
00033     constexpr array <sunrealtype,2> CVodeTolerances={1.0e-16,1.0e-16}; // - relative and absolute
tolerances of the CVode solver
00034     constexpr int StencilOrder=13;                                         // - accuracy order of the
stencils in the range 1-13
00035     constexpr unrealtype physical_sidelength=300e-6;                   // - physical length of the
lattice in meters
00036     constexpr sunindextype latticepoints=6e3;                          // - number of lattice points
00037     constexpr bool periodic=true;                                     // - periodic or vanishing
boundary values
00038     int processOrder=3;                                              // - included processes of the
weak-field expansion, see README.md
00039     constexpr unrealtype simulationTime=100.0e-61;                  // - physical total
simulation time
00040     constexpr int numberOfSteps=100;                                    // - discrete time steps
00041     constexpr int outputStep=100;                                     // - output step multiples
00042     constexpr char outputStyle='c';                                    // - output in csv (c) or
binary (b) format
00043
00044     // Add electromagnetic waves.
00045     planewave plane1;                                                 // - A plane wave with
00046     plane1.k = {1e5,0,0};                                            // - wavevector (normalized to 1/\lambda )
00047     plane1.p = {0,0,0.1};                                           // - amplitude/polarization
00048     plane1.phi = {0,0,0};                                           // - phase shift
00049     planewave plane2;                                               // - Another plane wave with
00050     plane2.k = {-1e6,0,0};                                            // - wavevector (normalized to 1/\lambda )
00051     plane2.p = {0,0,0.5};                                           // - amplitude/polarization
00052     plane2.phi = {0,0,0};                                           // - phase shift
00053     // Do not comment out this vector, even if no plane wave is used. But if, emplace used plane
waves.
00054     vector<planewave> planewaves;
00055     //planewaves.emplace_back(plane1);
00056     //planewaves.emplace_back(plane2);
00057
00058     gaussian1D gauss1;                                              // - A Gaussian wave with
00059     gauss1.k = {1.0e6,0,0};                                            // - wavevector (normalized to 1/\lambda )
00060     gauss1.p = {0,0,0.1};                                           // - polarization/amplitude
00061     gauss1.x0 = {100e-6,0,0};                                         // - shift from origin
```

```

00062     gauss1.phig = 5e-6;           /// - width
00063     gauss1.phi = {0,0,0};        /// - phase shift
00064     gaussian1D gauss2;          /// Another Gaussian with
00065     gauss2.k = {-0.2e6,0,0};    /// - wavevector (normalized to  $\lambda$ )
00066     gauss2.p = {0,0,0.5};       /// - polarization/amplitude
00067     gauss2.x0 = {200e-6,0,0};   /// - shift from origin
00068     gauss2.phig = 15e-6;       /// - width
00069     gauss2.phi = {0,0,0};       /// - phase shift
00070 // Do not comment out this vector, even if no Gaussian wave is used. But if, emplace used Gaussian
waves.
00071     vector<gaussian1D> Gaussians1D;
00072     Gaussians1D.emplace_back(gauss1);
00073     Gaussians1D.emplace_back(gauss2);
00074
00075 ///////////////////////////////////////////////////////////////////
00076     int *interactions = &processOrder;
00077     Sim1D(CVodeTolerances, StencilOrder, physical_sidelength, latticepoints,
00078             periodic, interactions, simulationtime, numberOfSteps,
00079             outputDirectory, outputStep, outputStyle,
00080             planewaves, Gaussians1D);
00081 */
00082 ///////////////////////////////////////////////////////////////////
00083
00084
00085 /////////////////////////////////////////////////////////////////// -- 2D -- ///////////////////////////////////////////////////////////////////
00086 /* A 2D simulation with specified */
00087 /*
00088 ///////////////////////////////////////////////////////////////////
00089     constexpr array<sunrealtype,2> CVodeTolerances={1.0e-12,1.0e-12}; // relative and absolute
tolerances of the CVode solver
00090     constexpr int StencilOrder=13;                                         // accuracy order of the
stencils in the range 1-13
00091     constexpr array<sunrealtype,2> physical_sidelengths={80e-6,80e-6}; // physical length of the
lattice in the given dimensions in meters
00092     constexpr array<sunindextype,2> latticepoints_per_dim={800,800}; // number of lattice points
per dimension
00093     constexpr array<int,2> patches_per_dim={2,2};                      // slicing of discrete
dimensions into patches
00094     constexpr bool periodic=true;                                         // periodic or vanishing
boundary values
00095     int processOrder=3;                                                 // included processes of the
weak-field expansion, see README.md
00096     constexpr unrealtype simulationTime=40e-6l;                         // physical total simulation
time
00097     constexpr int numberOfSteps=100;                                       // discrete time steps
00098     constexpr int outputStep=100;                                         // output step multiples
00099     constexpr char outputStyle='c';                                       // output in csv (c) or
binary (b) format
0100
0101 // Add electromagnetic waves.
0102     planewave plane1;                                              // A plane wave with
0103     plane1.k = {1e5,0,0};                                            // - wavevector (normalized to  $\lambda$ )
0104     plane1.p = {0,0,0.1};                                           // - amplitude/polarization
0105     plane1.phi = {0,0,0};                                            // - phase shift
0106     planewave plane2;                                              // Another plane wave with
0107     plane2.k = {-1e6,0,0};                                           // - wavevector
0108     plane2.p = {0,0,0.5};                                           // - amplitude/polarization
0109     plane2.phi = {0,0,0};                                            // - phase shift
0110 // Do not comment out this vector, even if no plane wave is used. But if, emplace used plane
waves.
0111     vector<planewave> planewaves;
0112     //planewaves.emplace_back(plane1);
0113     //planewaves.emplace_back(plane2);
0114
0115     gaussian2D gauss1;                                              // A Gaussian wave with
0116     gauss1.x0 = {40e-6,40e-6};                                         // - center it approaches
0117     gauss1.axis = {1,0};                                              // - normalized direction _from_ which the wave approaches the
center
0118     gauss1.amp = 0.5;                                                 // - amplitude
0119     gauss1.phip = 2*atan(0);                                         // - polarization rotation from TE-mode (z-axis)
0120     gauss1.w0 = 2.3e-6;                                              // - taille
0121     gauss1.zr = 16.619e-6;                                           // - Rayleigh length
0122 // the wavelength is determined by the relation  $\lambda = \pi w_0^2 / z_R$ 
0123     gauss1.ph0 = 2e-5;                                                // - beam center
0124     gauss1.phA = 0.45e-5;                                             // - beam length
0125     gaussian2D gauss2;                                              // Another Gaussian wave with
0126     gauss2.x0 = {40e-6,40e-6};                                         // - center it approaches
0127     gauss2.axis = {-0.7071,0.7071};                                    // - normalized direction from which the wave approaches the
center
0128     gauss2.amp = 0.5;                                                 // - amplitude
0129     gauss2.phip = 2*atan(0);                                         // - polarization rotation from TE-mode (z-axis)
0130     gauss2.w0 = 2.3e-6;                                              // - taille
0131     gauss2.zr = 16.619e-6;                                           // - Rayleigh length
0132     gauss2.ph0 = 2e-5;                                                // - beam center
0133     gauss2.phA = 0.45e-5;                                             // - beam length
0134 // Do not comment out this vector, even if no Gaussian wave is used. But if, emplace used Gaussian
waves.

```

```

00135     vector<gaussian2D> Gaussians2D;
00136     Gaussians2D.emplace_back(gauss1);
00137     Gaussians2D.emplace_back(gauss2);
00138
00139     //// Do not change this below ////
00140     static_assert(latticepoints_per_dim[0]*patches_per_dim[0]==0 &&
00141         latticepoints_per_dim[1]*patches_per_dim[1]==0,
00142         "The number of lattice points in each dimension must be "
00143         "divisible by the number of patches in that direction.");
00144     int * interactions = &processOrder;
00145     Sim2D(CVodeTolerances,StencilOrder,physical_sidelengths,
00146             latticepoints_per_dim,patches_per_dim,periodic,interactions,
00147             simulationTime,numberOfSteps,outputDirectory,outputStep,
00148             outputStyle,planewaves,Gaussians2D);
00149
00150 *****
00151
00152
00153 ////////// -- 3D -- ///////////
00154 /** A 3D simulation with specified */
00155 /*
00156     // Specify your settings here ///////////////////////////////////////////////////////////////////
00157     constexpr array<sunrealtype,2> CVodeTolerances={1.0e-12,1.0e-12}; /// - relative and
00158     absolute tolerances of the CVode solver /// - accuracy order of
00159     constexpr int StencilOrder=13; the stencils in the range 1-13 /// - physical dimensions
00160     constexpr array<sunrealtype,3> physical_sidelengths={80e-6,80e-6,20e-6}; in meters /// - number of lattice
00161     constexpr array<sunindextype,3> latticepoints_per_dim={800,800,200}; points in any dimension /// - slicing of discrete
00162     constexpr array<int,3> patches_per_dim= {8,8,2}; dimensions into patches /// - periodic or
00163     constexpr bool periodic=true; non-periodic boundaries /// - processes of the
00164     int processOrder=3; weak-field expansion, see README.md /// - physical total
00165     constexpr unrealtype simulationTime=40e-6; simulation time /// - discrete time steps
00166     constexpr int numberOfSteps=40; /// - output step
00167
00168     constexpr int outputStep=20; multiples /// - output in csv (c)
00169     constexpr char outputStyle='b'; or binary (b) format
00170
00171     /// Add electromagnetic waves.
00172     planewave plane1; /// A plane wave with
00173     plane1.k = {1e5,0,0}; /// - wavevector (normalized to \f$ 1/\lambda \f$)
00174     plane1.p = {0,0,0.1}; /// - amplitude/polarization
00175     plane1.phi = {0,0,0}; /// - phase shift
00176     planewave plane2; /// Another plane wave with
00177     plane2.k = {-1e6,0,0}; /// - wavevector (normalized to \f$ 1/\lambda \f$)
00178     plane2.p = {0,0,0.5}; /// - amplitude/polarization
00179     plane2.phi = {0,0,0}; /// - phase shift
00180     // Do not comment out this vector, even if no plane wave is used. But if, emplace used plane
00181     waves.
00182     vector<planewave> planewaves;
00183     //planewaves.emplace_back(plane1);
00184     //planewaves.emplace_back(plane2);
00185
00186     gaussian3D gauss1; /// A Gaussian wave with
00187     gauss1.x0 = {40e-6,40e-6,10e-6}; /// - center it approaches
00188     gauss1.axis = {1,0,0}; /// - normalized direction _from_ which the wave approaches
00189     the center
00190     gauss1.amp = 0.05; /// - amplitude
00191     gauss1.phip = 2*atan(0); /// - polarization rotation from TE-mode (z-axis)
00192     gauss1.w0 = 2.3e-6; /// - taille
00193     gauss1.zr = 16.619e-6; /// - Rayleigh length
00194     /// the wavelength is determined by the relation \f$ \lambda = \pi w_0^2/z_R \f$
00195     gauss1.ph0 = 2e-5; /// - beam center
00196     gauss1.phA = 0.45e-5; /// - beam length
00197     gaussian3D gauss2; /// Another Gaussian wave with
00198     gauss2.x0 = {40e-6,40e-6,10e-6}; /// - center it approaches
00199     gauss2.axis = {0,1,0}; /// - normalized direction from which the wave approaches the
00200     center
00201     gauss2.amp = 0.05; /// - amplitude
00202     gauss2.phip = 2*atan(0); /// - polarization rotation from TE-mode (z-axis)
00203     gauss2.w0 = 2.3e-6; /// - taille
00204     gauss2.zr = 16.619e-6; /// - Rayleigh length
00205     gauss2.ph0 = 2e-5; /// - beam center
00206     gauss2.phA = 0.45e-5; /// - beam length
00207     // Do not comment out this vector, even if no Gaussian wave is used. But if, emplace used Gaussian
00208     waves.
00209     vector<gaussian3D> Gaussians3D;
00210     Gaussians3D.emplace_back(gauss1);
00211     Gaussians3D.emplace_back(gauss2);
00212

```

```

00207     //// Do not change this below /////
00208     static_assert(latticepoints_per_dim[0]%patches_per_dim[0]==0 &&
00209         latticepoints_per_dim[1]%patches_per_dim[1]==0 &&
00210         latticepoints_per_dim[2]%patches_per_dim[2]==0,
00211         "The number of lattice points in each dimension must be "
00212         "divisible by the number of patches in that direction.");
00213     static_assert(latticepoints_per_dim[0]/patches_per_dim[0] ==
00214         latticepoints_per_dim[1]/patches_per_dim[1] &&
00215         latticepoints_per_dim[0]/patches_per_dim[0] ==
00216         latticepoints_per_dim[2]/patches_per_dim[2],
00217         "At 3D simulations you are forced to make patches cubic in terms of "
00218         "lattice points as this is decisive for computational efficiency.");
00219     int *interactions = &processOrder;
00220     Sim3D(CVodeTolerances,StencilOrder,physical_sidelengths,
00221         latticepoints_per_dim,patches_per_dim,periodic,interactions,
00222         simulationTime,numberOfSteps,outputDirectory,outputStep,
00223         outputStyle,planewaves,Gaussians3D);
00224     */
00225     //////////////////////////////// END OF CONFIGURATION /////////////////////
00226
00227 //----- END OF CONFIGURATION -----//
00228
00229 // Finalize MPI environment
00230 MPI_Finalize();
00231
00232 return 0;
00233 }

```

6.45 main.cpp

[Go to the documentation of this file.](#)

```

00001 /// @file main.cpp
00002 /// @brief Main function to configure the user's simulation settings
00003
00004
00005 #include "SimulationFunctions.h" /* complete simulation functions and all headers */
00006
00007 using namespace std;
00008
00009 int main(int argc, char *argv[])
0010 {
0011     /** Determine the output directory.
0012      * A "SimResults" folder will be created if non-existent
0013      * with a subdirectory named in the identifier format
0014      * "yy-mm-dd_hh-MM-ss" that contains the csv files      */
0015     constexpr auto outputDirectory = "/path/to/directory/";
0016
0017     if(!filesystem::exists(outputDirectory)) {
0018         cerr<<"\nOutput directory nonexistent.\n";
0019         exit(1);
0020     }
0021
0022     // Initialize MPI environment
0023     int provided;
0024     MPI_Init_thread(&argc, &argv, MPI_THREAD_SINGLE, &provided);
0025
0026
0027 //----- BEGIN OF CONFIGURATION -----//
0028
0029 ////////////////// -- 1D -- //////////////////
0030 /** A 1D simulation with specified */
0031 /*
0032     //// Specify your settings here ////
0033     constexpr array <sunrealtype,2> CVodeTolerances={1.0e-16,1.0e-16}; // - relative and absolute
0034     tolerances of the CVode solver
0035     constexpr int StencilOrder=13; // - accuracy order of the
0036     stencils in the range 1-13
0037     constexpr unrealtype physical_sidelength=300e-6; // - physical length of the
0038     lattice in meters
0039     constexpr sunindextype latticepoints=6e3; // - number of lattice points
0040     constexpr bool periodic=true; // - periodic or vanishing
0041     boundary values
0042     int processOrder=3; // - included processes of the
0043     weak-field expansion, see README.md
0044     constexpr unrealtype simulationTime=100.0e-61; // - physical total
0045     simulation time
0046     constexpr int numberOfSteps=100; // - discrete time steps
0047     constexpr int outputStep=100; // - output step multiples
0048     constexpr char outputStyle='c'; // - output in csv (c) or
0049     binary (b) format
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
0999

```

```

00044  /// Add electromagnetic waves.
00045  planewave plane1;           /// A plane wave with
00046  plane1.k = {1e5,0,0};      /// - wavevector (normalized to  $\mathbf{k} / \lambda$ )
00047  plane1.p = {0,0,0.1};     /// - amplitude/polarization
00048  plane1.phi = {0,0,0};     /// - phase shift
00049  planewave plane2;         /// Another plane wave with
00050  plane2.k = {-1e6,0,0};    /// - wavevector (normalized to  $\mathbf{k} / \lambda$ )
00051  plane2.p = {0,0,0.5};    /// - amplitude/polarization
00052  plane2.phi = {0,0,0};    /// - phase shift
00053  // Do not comment out this vector, even if no plane wave is used. But if, emplace used plane
waves.
00054  vector<planewave> planewaves;
00055  //planewaves.emplace_back(plane1);
00056  //planewaves.emplace_back(plane2);
00057
00058  gaussian1D gauss1;        /// A Gaussian wave with
00059  gauss1.k = {1.0e6,0,0};   /// - wavevector (normalized to  $\mathbf{k} / \lambda$ )
00060  gauss1.p = {0,0,0.1};     /// - polarization/amplitude
00061  gauss1.x0 = {100e-6,0,0}; /// - shift from origin
00062  gauss1.phig = 5e-6;      /// - width
00063  gauss1.phi = {0,0,0};    /// - phase shift
00064  gaussian1D gauss2;       /// Another Gaussian with
00065  gauss2.k = {-0.2e6,0,0};  /// - wavevector (normalized to  $\mathbf{k} / \lambda$ )
00066  gauss2.p = {0,0,0.5};    /// - polarization/amplitude
00067  gauss2.x0 = {200e-6,0,0}; /// - shift from origin
00068  gauss2.phig = 15e-6;    /// - width
00069  gauss2.phi = {0,0,0};    /// - phase shift
00070  // Do not comment out this vector, even if no Gaussian wave is used. But if, emplace used Gaussian
waves.
00071  vector<gaussian1D> Gaussians1D;
00072  Gaussians1D.emplace_back(gauss1);
00073  Gaussians1D.emplace_back(gauss2);
00074
00075  //////////////////////////////////////////////////////////////////
00076  int *interactions = &processOrder;
00077  Sim1D(CVodeTolerances, StencilOrder, physical_sidelength, latticepoints,
00078      periodic, interactions, simulationTime, numberOfSteps,
00079      outputDirectory, outputStep, outputStyle,
00080      planewaves, Gaussians1D);
00081  */
00082  //////////////////////////////////////////////////////////////////
00083
00084
00085  ////////////////////////////////////////////////////////////////// -- 2D -- //////////////////////////////////////////////////////////////////
00086  /** A 2D simulation with specified */
00087  /*
00088  //////////////////////////////////////////////////////////////////
00089  constexpr array<sunrealtype,2> CVodeTolerances={1.0e-12,1.0e-12};  /// - relative and absolute
tolerances of the CVode solver
00090  constexpr int StencilOrder=13;                                         /// - accuracy order of the
stencils in the range 1-13
00091  constexpr array<sunrealtype,2> physical_sidelengths={80e-6,80e-6};  /// - physical length of the
lattice in the given dimensions in meters
00092  constexpr array<sunindextype,2> latticepoints_per_dim={800,800};   /// - number of lattice points
per dimension
00093  constexpr array<int,2> patches_per_dim={2,2};                      /// - slicing of discrete
dimensions into patches
00094  constexpr bool periodic=true;                                         /// - periodic or vanishing
boundary values
00095  int processOrder=3;                                                 /// - included processes of the
weak-field expansion, see README.md
00096  constexpr sunrealtype simulationTime=40e-61;                      /// - physical total simulation
time
00097  constexpr int numberOfSteps=100;                                       /// - discrete time steps
00098  constexpr int outputStep=100;                                         /// - output step multiples
00099  constexpr char outputStyle='c';                                       /// - output in csv (c) or
binary (b) format
00100
00101  //////////////////////////////////////////////////////////////////
00102  planewave plane1;           /// A plane wave with
00103  plane1.k = {1e5,0,0};      /// - wavevector (normalized to  $\mathbf{k} / \lambda$ )
00104  plane1.p = {0,0,0.1};     /// - amplitude/polarization
00105  plane1.phi = {0,0,0};     /// - phase shift
00106  planewave plane2;         /// Another plane wave with
00107  plane2.k = {-1e6,0,0};    /// - wavevector
00108  plane2.p = {0,0,0.5};    /// - amplitude/polarization
00109  plane2.phi = {0,0,0};    /// - phase shift
00110  // Do not comment out this vector, even if no plane wave is used. But if, emplace used plane
waves.
00111  vector<planewave> planewaves;
00112  //planewaves.emplace_back(plane1);
00113  //planewaves.emplace_back(plane2);
00114
00115  gaussian2D gauss1;        /// A Gaussian wave with
00116  gauss1.x0 = {40e-6,40e-6}; /// - center it approaches
00117  gauss1.axis = {1,0};       /// - normalized direction _from_ which the wave approaches the
center

```

```

00118 gauss1.amp = 0.5;                                /// - amplitude
00119 gauss1.phip = 2*pi*atan(0);                      /// - polarization rotation from TE-mode (z-axis)
00120 gauss1.w0 = 2.3e-6;                             /// - taille
00121 gauss1.zr = 16.619e-6;                          /// - Rayleigh length
00122 /// the wavelength is determined by the relation \f$ \lambda = \pi * w_0^2 / z_R \f$
00123 gauss1.ph0 = 2e-5;                            /// - beam center
00124 gauss1.phA = 0.45e-5;                          /// - beam length
00125 gaussian2D gauss2;                           /// Another Gaussian wave with
00126 gauss2.x0 = {40e-6,40e-6};                     /// - center it approaches
00127 gauss2.axis = {-0.7071,0.7071};                /// - normalized direction from which the wave approaches the
00128 center
00129 gauss2.amp = 0.5;                                /// - amplitude
00130 gauss2.phip = 2*pi*atan(0);                      /// - polarization rotation from TE-mode (z-axis)
00131 gauss2.w0 = 2.3e-6;                             /// - taille
00132 gauss2.zr = 16.619e-6;                          /// - Rayleigh length
00133 gauss2.ph0 = 2e-5;                            /// - beam center
00134 gauss2.phA = 0.45e-5;                          /// - beam length
00135 // Do not comment out this vector, even if no Gaussian wave is used. But if, emplace used Gaussian
00136 waves.
00137     vector<gaussian2D> Gaussians2D;
00138     Gaussians2D.emplace_back(gauss1);
00139     Gaussians2D.emplace_back(gauss2);
00140
00141 /// Do not change this below ///
00142 static_assert(latticepoints_per_dim[0]*patches_per_dim[0]==0 &
00143               latticepoints_per_dim[1]*patches_per_dim[1]==0,
00144               "The number of lattice points in each dimension must be "
00145               "divisible by the number of patches in that direction.");
00146 int * interactions = &processOrder;
00147 Sim2D(CVodeTolerances, StencilOrder, physical_sidelengths,
00148         latticepoints_per_dim, patches_per_dim, periodic, interactions,
00149         simulationTime,numberOfSteps,outputDirectory,outputStep,
00150         outputStyle,planewaves,Gaussians2D);
00151
00152
00153 ////////// -- 3D -- //////////
00154 /** A 3D simulation with specified */
00155 /*
00156 /// Specify your settings here ///
00157 constexpr array<sunrealtype,2> CVodeTolerances={1.0e-12,1.0e-12};      /// - relative and
00158 absolute tolerances of the CVode solver
00159 constexpr int StencilOrder=13;                                         /// - accuracy order of
00160 the stencils in the range 1-13
00161 constexpr array<sunrealtype,3> physical_sidelengths={80e-6,80e-6,20e-6}; // - physical dimensions
00162 in meters
00163 constexpr array<sunindextype,3> latticepoints_per_dim={800,800,200};    /// - number of lattice
00164 points in any dimension
00165 constexpr array<int,3> patches_per_dim= {8,8,2};                      /// - slicing of discrete
00166 dimensions into patches
00167 constexpr bool periodic=true;                                         /// - periodic or
00168 non-periodic boundaries
00169 int processOrder=3;                                                 /// - processes of the
00170 weak-field expansion, see README.md
00171 constexpr unrealtype simulationTime=40e-6;                         /// - physical total
00172 simulation time
00173 constexpr int numberOfSteps=40;                                       /// - discrete time steps
00174 constexpr int outputStep=20;                                         /// - output step
00175 multiples
00176 constexpr char outputStyle='b';                                     /// - output in csv (c)
00177 or binary (b) format
00178
00179 /// Add electromagnetic waves.
00180 planewave plane1;                                              /// A plane wave with
00181 plane1.k = {1e5,0,0};                                         /// - wavevector (normalized to \f$ 1/\lambda \f$)
00182 plane1.p = {0,0,0.1};                                         /// - amplitude/polarization
00183 plane1.phi = {0,0,0};                                         /// - phase shift
00184 planewave plane2;                                              /// Another plane wave with
00185 plane2.k = {-1e6,0,0};                                         /// - wavevector (normalized to \f$ 1/\lambda \f$)
00186 plane2.p = {0,0,0.5};                                         /// - amplitude/polarization
00187 plane2.phi = {0,0,0};                                         /// - phase shift
00188 // Do not comment out this vector, even if no plane wave is used. But if, emplace used plane
00189 waves.
00190     vector<planewave> planewaves;
00191     //planewaves.emplace_back(plane1);
00192     //planewaves.emplace_back(plane2);
00193
00194     gaussian3D gauss1;                                         /// A Gaussian wave with
00195     gauss1.x0 = {40e-6,40e-6,10e-6};                         /// - center it approaches
00196     gauss1.axis = {1,0,0};                                      /// - normalized direction _from_ which the wave approaches
00197     the center
00198     gauss1.amp = 0.05;                                         /// - amplitude
00199     gauss1.phip = 2*pi*atan(0);                      /// - polarization rotation from TE-mode (z-axis)
00200     gauss1.w0 = 2.3e-6;                             /// - taille
00201     gauss1.zr = 16.619e-6;                          /// - Rayleigh length
00202 /// the wavelength is determined by the relation \f$ \lambda = \pi * w_0^2 / z_R \f$

```

```

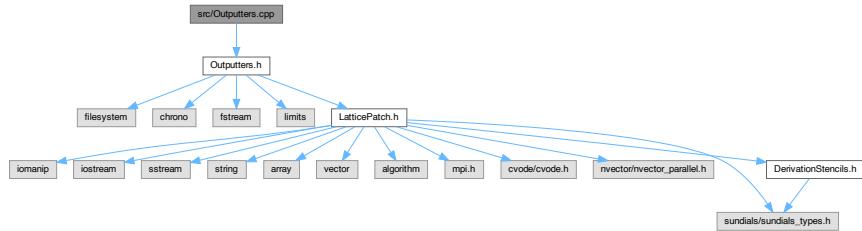
00191     gauss1.ph0 = 2e-5;           /// - beam center
00192     gauss1.phA = 0.45e-5;       /// - beam length
00193     gaussian3D gauss2;         /// Another Gaussian wave with
00194     gauss2.x0 = {40e-6, 40e-6, 10e-6}; // - center it approaches
00195     gauss2.axis = {0,1,0};      /// - normalized direction from which the wave approaches the
00196     center
00197     gauss2.amp = 0.05;          /// - amplitude
00198     gauss2.phip = 2*pi*atan(0); // - polarization rotation from TE-mode (z-axis)
00199     gauss2.w0 = 2.3e-6;         /// - taille
00200     gauss2.zr = 16.619e-6;      /// - Rayleigh length
00201     gauss2.ph0 = 2e-5;          /// - beam center
00202     gauss2.phA = 0.45e-5;       /// - beam length
00203     // Do not comment out this vector, even if no Gaussian wave is used. But if, emplace used Gaussian
00204     // waves.
00205     vector<gaussian3D> Gaussians3D;
00206     Gaussians3D.emplace_back(gauss1);
00207     Gaussians3D.emplace_back(gauss2);
00208
00209     ///////////////////////////////////////////////////////////////////
00210     static_assert(latticepoints_per_dim[0]*patches_per_dim[0]==0 &&
00211                   latticepoints_per_dim[1]*patches_per_dim[1]==0 &&
00212                   latticepoints_per_dim[2]*patches_per_dim[2]==0,
00213                   "The number of lattice points in each dimension must be "
00214                   "divisible by the number of patches in that direction.");
00215     static_assert(latticepoints_per_dim[0]/patches_per_dim[0] ==
00216                   latticepoints_per_dim[1]/patches_per_dim[1] &&
00217                   latticepoints_per_dim[0]/patches_per_dim[0] ==
00218                   latticepoints_per_dim[2]/patches_per_dim[2],
00219                   "At 3D simulations you are forced to make patches cubic in terms of "
00220                   "lattice points as this is decisive for computational efficiency.");
00221     int *interactions = &processOrder;
00222     Sim3D(CVodeTolerances, StencilOrder, physical_sidelengths,
00223           latticepoints_per_dim, patches_per_dim, periodic, interactions,
00224           simulationTime, numberOfSteps, outputDirectory, outputStep,
00225           outputStyle, planewaves, Gaussians3D);
00226
00227     //----- END OF CONFIGURATION -----
00228
00229     // Finalize MPI environment
00230     MPI_Finalize();
00231
00232     return 0;
00233 }

```

6.46 src/Outputters.cpp File Reference

Generation of output writing to disk.

```
#include "Outputters.h"
Include dependency graph for Outputters.cpp:
```



6.46.1 Detailed Description

Generation of output writing to disk.

Definition in file [Outputters.cpp](#).

6.47 Outputters.cpp

[Go to the documentation of this file.](#)

```

00001 ///////////////////////////////////////////////////////////////////
00002 /// @file Outputters.cpp
00003 /// @brief Generation of output writing to disk
00004 ///////////////////////////////////////////////////////////////////
00005
00006 #include "Outputters.h"
00007
00008 namespace fs = std::filesystem;
00009 namespace chrono = std::chrono;
00010
00011 /// Directly generate the simCode at construction
00012 OutputManager::OutputManager() {
00013     simCode = SimCodeGenerator();
00014     outputStyle = 'c';
00015 }
00016
00017 /// Generate the identifier number reverse from year to minute in the format
00018 /// yy-mm-dd_hh-MM-ss
00019 std::string OutputManager::SimCodeGenerator() {
00020     const chrono::time_point<chrono::system_clock> now{
00021         chrono::system_clock::now()};
00022     const chrono::year_month_day ymd{chrono::floor<chrono::days>(now)};
00023     const auto tod = now - chrono::floor<chrono::days>(now);
00024     const chrono::hh_mm_ss hms{tod};
00025
00026     std::stringstream temp;
00027     temp << std::setfill('0') << std::setw(2)
00028         << static_cast<int>(ymd.year() - chrono::years(2000)) << "-"
00029         << std::setfill('0') << std::setw(2)
00030         << static_cast<unsigned>(ymd.month()) << "-"
00031         << std::setfill('0') << std::setw(2)
00032         << static_cast<unsigned>(ymd.day()) << "_"
00033         << std::setfill('0') << std::setw(2) << hms.hours().count()
00034         << "-" << std::setfill('0')
00035         << std::setw(2) << hms.minutes().count() << "-"
00036         << std::setfill('0') << std::setw(2)
00037         << hms.seconds().count();
00038     //< "_" << hms.subseconds().count(); // subseconds render the filename
00039     // too large
00040     return temp.str();
00041 }
00042
00043 /** Generate the folder to save the data to by one process:
00044 * In the given directory it creates a direcor "SimResults" and a directory
00045 * with the simCode. The relevant part of the main file is written to a
00046 * "config.txt" file in that directory to log the settings. */
00047 void OutputManager::generateOutputFolder(const std::string &dir) {
00048     // Do this only once for the first process
00049     int myPrc;
00050     MPI_Comm_rank(MPI_COMM_WORLD, &myPrc);
00051     if (myPrc == 0) {
00052         if (!fs::is_directory(dir))
00053             fs::create_directory(dir);
00054         if (!fs::is_directory(dir + "/SimResults"))
00055             fs::create_directory(dir + "/SimResults");
00056         if (!fs::is_directory(dir + "/SimResults/" + simCode))
00057             fs::create_directory(dir + "/SimResults/" + simCode);
00058     }
00059     // path variable for the output generation
00060     Path = dir + "/SimResults/" + simCode + "/";
00061
00062     // Logging configurations from main.cpp
00063     std::ifstream fin("main.cpp");
00064     std::ofstream fout(Path + "config.txt");
00065     std::string line;
00066     int begin=1000;
00067     for (int i = 1; !fin.eof(); i++) {
00068         getline(fin, line);
00069         if (line.starts_with("    //----- B")) {
00070             begin=i;
00071         }
00072         if (i < begin) {
00073             continue;
00074         }
00075         fout << line << std::endl;
00076         if (line.starts_with("    //----- E")) {
00077             break;
00078         }
00079     }
00080     return;
00081 }
00082

```

```

00083 void OutputManager::set_outputStyle(const char _outputStyle){
00084     outputStyle = _outputStyle;
00085 }
00086
00087 /** Write the field data either in csv format to one file per each process
00088 * (patch) or in binary form to a single file. Files are stores inthe simCode
00089 * directory. For csv files the state (simulation step) denotes the
00090 * prefix and the suffix after an underscore is given by the process/patch
00091 * number. Binary files are simply named after the step number. */
00092 void OutputManager::outUState(const int &state, const Lattice &lattice,
00093     const LatticePatch &latticePatch) {
00094     switch(outputStyle){
00095         case 'c': { // one csv file per process
00096             std::ofstream ofs;
00097             ofs.open(Path + std::to_string(state) + "_"
00098                     + std::to_string(lattice.my_prc) + ".csv");
00099             // Precision of sunrealtype in significant decimal digits; 15 for IEEE double
00100             ofs << std::setprecision(std::numeric_limits<sunrealtype>::digits10);
00101
00102             // Walk through each lattice point
00103             const sunindextype totalNP = latticePatch.discreteSize();
00104             for (sunindextype i = 0; i < totalNP * 6; i += 6) {
00105                 // Six columns to contain the field data: Ex,Ey,Ez,Bx,By,Bz
00106                 ofs << latticePatch.uData[i + 0] << "," << latticePatch.uData[i + 1] << ","
00107                 << latticePatch.uData[i + 2] << "," << latticePatch.uData[i + 3] << ","
00108                 << latticePatch.uData[i + 4] << "," << latticePatch.uData[i + 5]
00109                 << std::endl;
00110             }
00111             ofs.close();
00112             break;
00113         }
00114
00115         case 'b': { // a single binary file
00116             // Open the output file
00117             MPI_File fh;
00118             const std::string filename = Path+std::to_string(state);
00119             MPI_File_open(lattice.comm,&filename[0],MPI_MODE_WRONLY|MPI_MODE_CREATE,
00120                         MPI_INFO_NULL,&fh);
00121             // number of datapoints in the patch with process offset
00122             const sunindextype count = latticePatch.discreteSize() *
00123                 lattice.get_dataPointDimension();
00124             MPI_Offset offset = lattice.my_prc*count*sizeof(MPI_SUNREALTYPE);
00125             // Go to offset in file and write data to it; maximal precision in
00126             // "native" representation
00127             MPI_File_set_view(fh,offset,MPI_SUNREALTYPE,MPI_SUNREALTYPE,"native",
00128                             MPI_INFO_NULL);
00129             MPI_Request write_request;
00130             MPI_File_iwrite_all(fh,latticePatch.uData,count,MPI_SUNREALTYPE,
00131                                 &write_request);
00132             MPI_Wait(&write_request,MPI_STATUS_IGNORE);
00133             MPI_File_close(&fh);
00134             break;
00135         }
00136         default: {
00137             errorKill("No valid output style defined."
00138                     " Choose between (c): one csv file per process,"
00139                     " (b) one binary file");
00140             break;
00141         }
00142     }
00143 }
00144

```

6.48 src/Outputters.h File Reference

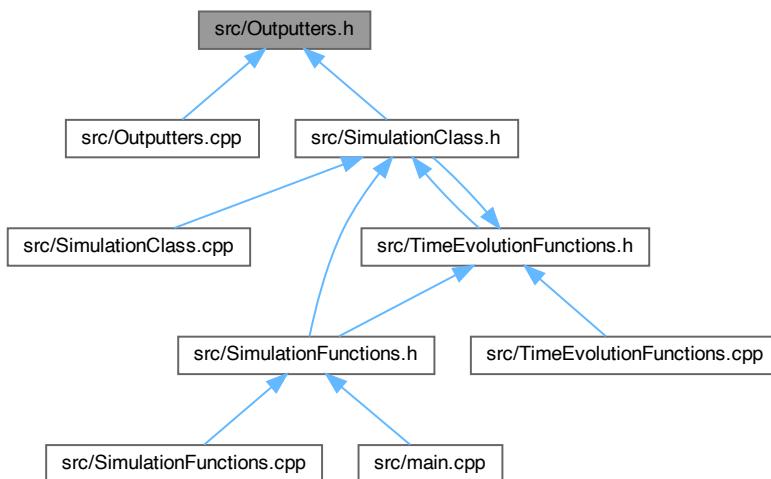
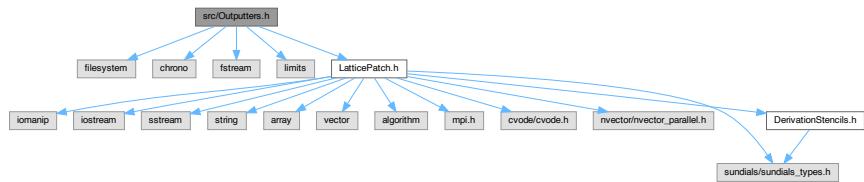
[OutputManager](#) class to outstream simulation data.

```

#include <filesystem>
#include <chrono>
#include <fstream>
#include <limits>
#include "LatticePatch.h"

```

Include dependency graph for Outputters.h:



Data Structures

- class [OutputManager](#)
Output Manager class to generate and coordinate output writing to disk.

6.48.1 Detailed Description

[OutputManager](#) class to outstream simulation data.

Definition in file [Outputters.h](#).

6.49 Outputters.h

[Go to the documentation of this file.](#)

```

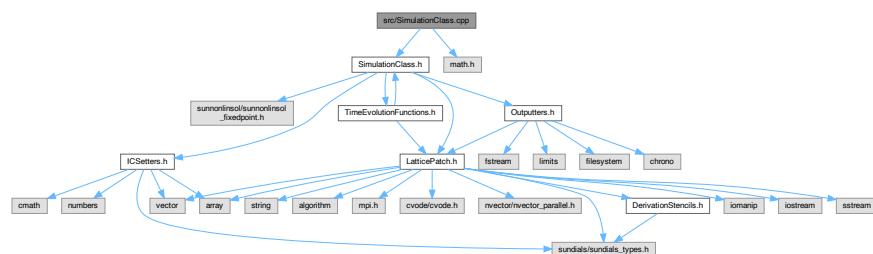
00001 ///////////////////////////////////////////////////////////////////
00002 /// @file Outputters.h
00003 /// @brief OutputManager class to outstream simulation data
00004 ///////////////////////////////////////////////////////////////////
00005
00006 #pragma once
00007
00008 // perform operations on the filesystem
00009 #include <filesystem>
00010
00011 // output controlling with limits and timestep
00012 #include <chrono>
00013 #include <fstream>
00014 #include <limits>
00015
00016 // project subfile header
00017 #include "LatticePatch.h"
00018
00019 /** @brief Output Manager class to generate and coordinate output writing to
00020 * disk */
00021 class OutputManager {
00022 private:
00023     /// function to create the Code of the Simulations
00024     static std::string SimCodeGenerator();
00025     /// variable to safe the SimCode generated at execution
00026     std::string simCode;
00027     /// variable for the path to the output folder
00028     std::string Path;
00029     /// output style; csv or binary
00030     char outputStyle;
00031 public:
00032     /// default constructor
00033     OutputManager();
00034     /// function that creates folder to save simulation data
00035     void generateOutputFolder(const std::string &dir);
00036     /// set the output style
00037     void set_outputStyle(const char _outputStyle);
00038     /// function to write data to disk in specified way
00039     void outUState(const int &state, const Lattice &lattice,
00040                     const LatticePatch &latticePatch);
00041     /// simCode getter function
00042     [[nodiscard]] const std::string &getSimCode() const { return simCode; }
00043 };
00044

```

6.50 src/SimulationClass.cpp File Reference

Interface to the whole [Simulation](#) procedure: from wave settings over lattice construction, time evolution and outputs (also all relevant CVODE steps are performed here)

```
#include "SimulationClass.h"
#include <math.h>
Include dependency graph for SimulationClass.cpp:
```



6.50.1 Detailed Description

Interface to the whole [Simulation](#) procedure: from wave settings over lattice construction, time evolution and outputs (also all relevant CVODE steps are performed here)

Definition in file [SimulationClass.cpp](#).

6.51 SimulationClass.cpp

[Go to the documentation of this file.](#)

```

00001 ///////////////////////////////////////////////////////////////////
00002 /// @file SimulationClass.cpp
00003 /// @brief Interface to the whole Simulation procedure:
00004 /// from wave settings over lattice construction, time evolution and outputs
00005 /// (also all relevant CVODE steps are performed here)
00006 ///////////////////////////////////////////////////////////////////
00007
00008 #include "SimulationClass.h"
00009
00010 #include <math.h>
00011
00012 /// Along with the simulation object, create the cartesian communicator and
00013 /// SUNContext object
00014 Simulation::Simulation(const int Nx, const int Ny, const int Nz,
00015     const int StencilOrder, const bool periodicity) :
00016     lattice(StencilOrder) {
00017     statusFlags = 0;
00018     t = 0;
00019     // Initialize the cartesian communicator
00020     lattice.initializeCommunicator(Nx, Ny, Nz, periodicity);
00021
00022     // Create the SUNContext object associated with the thread of execution
00023     int retval = 0;
00024     retval = SUNContext_Create(&lattice.comm, &lattice.sunctx);
00025     if (check(retval, "SUNContext_Create", 1, lattice.my_prc))
00026         MPI_Abort(lattice.comm, 1);
00027 }
00028
00029 /// Free the CVode solver memory and Sundials context object with the finish of
00030 /// the simulation
00031 Simulation::~Simulation() {
00032     // Free solver memory
00033     if (statusFlags & CvodeObjectSetUp) {
00034         CVodeFree(&cvode_mem);
00035         SUNNonlinSolFree(NLS);
00036         SUNContext_Free(&lattice.sunctx);
00037     }
00038 }
00039
00040 /// Set the discrete dimensions, the number of points per dimension
00041 void Simulation::setDiscreteDimensionsOfLattice(const sunindextype nx,
00042     const sunindextype ny, const sunindextype nz) {
00043     checkNoFlag(LatticePatchworkSetUp);
00044     lattice.setDiscreteDimensions(nx, ny, nz);
00045     statusFlags |= LatticeDiscreteSetUp;
00046 }
00047
00048 /// Set the physical dimensions with lenghts in micro meters
00049 void Simulation::setPhysicalDimensionsOfLattice(const sunrealtype lx,
00050     const sunrealtype ly, const sunrealtype lz) {
00051     checkNoFlag(LatticePatchworkSetUp);
00052     lattice.setPhysicalDimensions(lx, ly, lz);
00053     statusFlags |= LatticePhysicalSetUp;
00054 }
00055
00056 /// Check that the lattice dimensions are set up and generate the patchwork
00057 void Simulation::initializePatchwork(const int nx, const int ny,
00058     const int nz) {
00059     checkFlag(LatticeDiscreteSetUp);
00060     checkFlag(LatticePhysicalSetUp);
00061
00062     // Generate the patchwork
00063     generatePatchwork(lattice, latticePatch, nx, ny, nz);
00064     latticePatch.initializeBuffers();
00065
00066     statusFlags |= LatticePatchworkSetUp;
00067 }
00068

```

```

00069 // Configure CVODE
00070 void Simulation::initializeCVODEObject(const sunrealtypes reltol,
00071     const sunrealtypes abstol) {
00072     checkFlag(SimulationStarted);
00073
00074     // CVode settings return value
00075     int retval = 0;
00076
00077     // Create CVODE object -- returns a pointer to the cvode memory structure
00078     // with Adams method (Adams-Moulton formula) solver chosen for non-stiff ODE
00079     cvode_mem = CVodeCreate(CV_ADAMS, lattice.sunctx);
00080
00081     // Specify user data and attach it to the main cvode memory block
00082     retval = CVodeSetUserData(
00083         cvode_mem,
00084         &latticePatch); // patch contains the user data as used in CVRhsFn
00085     if (check(retval, "CVodeSetUserData", 1, lattice.my_prc))
00086         MPI_Abort(lattice.comm, 1);
00087
00088     // Initialize CVODE solver
00089     retval = CVodeInit(cvode_mem, TimeEvolution::f, 0,
00090                         latticePatch.u); // allocate memory, CVRhsFn f, t_i=0, u
00091                         // contains the initial values
00092     if (check(retval, "CVodeInit", 1, lattice.my_prc))
00093         MPI_Abort(lattice.comm, 1);
00094
00095     // Create fixed point nonlinear solver object (suitable for non-stiff ODE) and
00096     // attach it to CVode
00097     NLS = SUNNonlinSol_FixedPoint(latticePatch.u, 0, lattice.sunctx);
00098     retval = CVodeSetNonlinearSolver(cvode_mem, NLS);
00099     if (check(retval, "CVodeSetNonlinearSolver", 1, lattice.my_prc))
00100         MPI_Abort(lattice.comm, 1);
00101
00102     // Anderson damping factor
00103     retval = SUNNonlinSolSetDamping_FixedPoint(NLS, 1);
00104     if (check(retval, "SUNNonlinSolSetDamping_FixedPoint", 1,
00105             lattice.my_prc)) MPI_Abort(lattice.comm, 1);
00106
00107     // Specify integration tolerances -- a scalar relative tolerance and scalar
00108     // absolute tolerance
00109     retval = CVodeSStolerances(cvode_mem, reltol, abstol);
00110     if (check(retval, "CVodeSStolerances", 1, lattice.my_prc))
00111         MPI_Abort(lattice.comm, 1);
00112
00113     // Specify the maximum number of steps to be taken by the solver in its
00114     // attempt to reach the next tout
00115     retval = CVodeSetMaxNumSteps(cvode_mem, 10000);
00116     if (check(retval, "CVodeSetMaxNumSteps", 1, lattice.my_prc))
00117         MPI_Abort(lattice.comm, 1);
00118
00119     // maximum number of warnings for too small h
00120     retval = CVodeSetMaxHnilWarns(cvode_mem, 3);
00121     if (check(retval, "CVodeSetMaxHnilWarns", 1, lattice.my_prc))
00122         MPI_Abort(lattice.comm, 1);
00123
00124     statusFlags |= CvodeObjectSetUp;
00125 }
00126
00127 /// Check if the lattice patchwork is set up and set the initial conditions
00128 void Simulation::start() {
00129     checkFlag(LatticeDiscreteSetUp);
00130     checkFlag(LatticePhysicalSetUp);
00131     checkFlag(LatticePatchworkSetUp);
00132     checkNoFlag(SimulationStarted);
00133     checkNoFlag(CvodeObjectSetUp);
00134     setInitialConditions();
00135     statusFlags |= SimulationStarted;
00136 }
00137
00138 /// Set initial conditions: Fill the lattice points with the initial field
00139 /// values
00140 void Simulation::setInitialConditions() {
00141     const sunrealtypes dx = latticePatch.getDelta(1);
00142     const sunrealtypes dy = latticePatch.getDelta(2);
00143     const sunrealtypes dz = latticePatch.getDelta(3);
00144     const sunindextypes nx = latticePatch.discreteSize(1);
00145     const sunindextypes ny = latticePatch.discreteSize(2);
00146     const sunindextypes totalNP = latticePatch.discreteSize();
00147     const sunrealtypes x0 = latticePatch.origin(1);
00148     const sunrealtypes y0 = latticePatch.origin(2);
00149     const sunrealtypes z0 = latticePatch.origin(3);
00150     sunindextypes px = 0, py = 0, pz = 0;
00151     #pragma omp parallel for default(none) \
00152     shared(nx, ny, totalNP, dx, dy, dz, x0, y0, z0) \
00153     firstprivate(px, py, pz) schedule(static)
00154     for (sunindextypes i = 0; i < totalNP * 6; i += 6) {
00155         px = (i / 6) % nx;

```

```

00156     py = ((i / 6) / nx) % ny;
00157     pz = ((i / 6) / nx) / ny;
00158     // Call the 'eval' function to fill the lattice points with the field data
00159     icsettings.eval(static_cast<sunrealtype>(px) * dx + x0,
00160                     static_cast<sunrealtype>(py) * dy + y0,
00161                     static_cast<sunrealtype>(pz) * dz + z0, &latticePatch.uData[i]);
00162   }
00163   return;
00164 }
00165
00166 /// Use parameters to add periodic IC layers
00167 void Simulation::addInitialConditions(const sunindextype xm,
00168                                         const sunindextype ym,
00169                                         const sunindextype zm /* zm=0 always */ ) {
00170   const unrealtype dx = latticePatch.getDelta(1);
00171   const unrealtype dy = latticePatch.getDelta(2);
00172   const unrealtype dz = latticePatch.getDelta(3);
00173   const sunindextype nx = latticePatch.discreteSize(1);
00174   const sunindextype ny = latticePatch.discreteSize(2);
00175   const sunindextype totalNP = latticePatch.discreteSize();
00176   // Correct for demanded displacement, rest as for setInitialConditions
00177   const unrealtype x0 = latticePatch.origin(1) + xm*lattice.get_tot_lx();
00178   const unrealtype y0 = latticePatch.origin(2) + ym*lattice.get_tot_ly();
00179   const unrealtype z0 = latticePatch.origin(3) + zm*lattice.get_tot_lz();
00180   sunindextype px = 0, py = 0, pz = 0;
00181   for (sunindextype i = 0; i < totalNP * 6; i += 6) {
00182     px = (i / 6) % nx;
00183     py = ((i / 6) / nx) % ny;
00184     pz = ((i / 6) / nx) / ny;
00185     icsettings.add(static_cast<sunrealtype>(px) * dx + x0,
00186                     static_cast<sunrealtype>(py) * dy + y0,
00187                     static_cast<sunrealtype>(pz) * dz + z0, &latticePatch.uData[i]);
00188   }
00189   return;
00190 }
00191
00192 /// Add initial conditions in one dimension
00193 void Simulation::addPeriodicICLayerInX() {
00194   addInitialConditions(-1, 0, 0);
00195   addInitialConditions(1, 0, 0);
00196   return;
00197 }
00198
00199 /// Add initial conditions in two dimensions
00200 void Simulation::addPeriodicICLayerInXY() {
00201   addInitialConditions(-1, -1, 0);
00202   addInitialConditions(-1, 0, 0);
00203   addInitialConditions(-1, 1, 0);
00204   addInitialConditions(0, 1, 0);
00205   addInitialConditions(0, -1, 0);
00206   addInitialConditions(1, -1, 0);
00207   addInitialConditions(1, 0, 0);
00208   addInitialConditions(1, 1, 0);
00209   return;
00210 }
00211
00212 /// Advance the solution in time -> integrate the ODE over an interval t
00213 void Simulation::advanceToTime(const unrealtype &tEnd) {
00214   checkFlag(SimulationStarted);
00215   int retval = 0;
00216   retval = CVode(cvode_mem, tEnd, latticePatch.u, &t,
00217                  CV_NORMAL); // CV_NORMAL: internal steps to reach tEnd, then
00218                  // interpolate to return latticePatch.u, return time
00219                  // reached by the solver as t
00220   if (check_retval(&retval, "CVode", 1, lattice.my_prc))
00221     MPI_Abort(lattice.comm, 1);
00222 }
00223
00224 /// Write specified simulation steps to disk
00225 void Simulation::outAllFieldData(const int & state) {
00226   checkFlag(SimulationStarted);
00227   outputManager.outUState(state, lattice, latticePatch);
00228 }
00229
00230 /// Check presence of configuration flags
00231 void Simulation::checkFlag(unsigned int flag) const {
00232   if (!(statusFlags & flag)) {
00233     std::string errorMessage;
00234     switch (flag) {
00235       case LatticeDiscreteSetUp:
00236         errorMessage = "The discrete size of the Simulation has not been set up";
00237         break;
00238       case LatticePhysicalSetUp:
00239         errorMessage = "The physical size of the Simulation has not been set up";
00240         break;
00241       case LatticePatchworkSetUp:
00242         errorMessage = "The patchwork for the Simulation has not been set up";
00243     }
00244   }
}

```

```

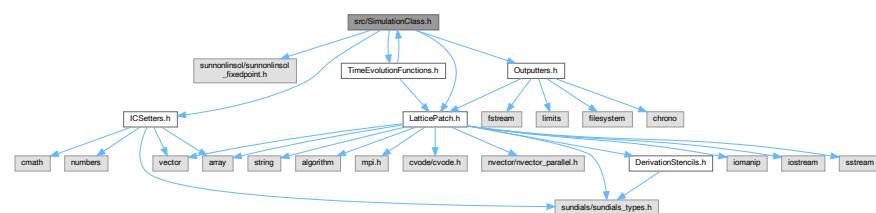
00243     break;
00244     case CvodeObjectSetUp:
00245         errorMessage = "The CVODE object has not been initialized";
00246         break;
00247     case SimulationStarted:
00248         errorMessage = "The Simulation has not been started";
00249         break;
00250     default:
00251         errorMessage = "Uppss, you've made a non-standard error, sadly I can't "
00252             "help you there";
00253         break;
00254     }
00255     errorKill(errorMessage);
00256 }
00257 return;
00258 }
00259
00260 // Check absence of configuration flags
00261 void Simulation::checkNoFlag(unsigned int flag) const {
00262     if ((statusFlags & flag)) {
00263         std::string errorMessage;
00264         switch (flag) {
00265             case LatticeDiscreteSetUp:
00266                 errorMessage =
00267                     "The discrete size of the Simulation has already been set up";
00268                 break;
00269             case LatticePhysicalSetUp:
00270                 errorMessage =
00271                     "The physical size of the Simulation has already been set up";
00272                 break;
00273             case LatticePatchworkSetUp:
00274                 errorMessage = "The patchwork for the Simulation has already been set up";
00275                 break;
00276             case CvodeObjectSetUp:
00277                 errorMessage = "The CVODE object has already been initialized";
00278                 break;
00279             case SimulationStarted:
00280                 errorMessage = "The simulation has already started, some changes are no "
00281                     "longer possible";
00282                 break;
00283             default:
00284                 errorMessage = "Uppss, you've made a non-standard error, sadly I can't "
00285                     "help you there";
00286                 break;
00287         }
00288         errorKill(errorMessage);
00289     }
00290     return;
00291 }

```

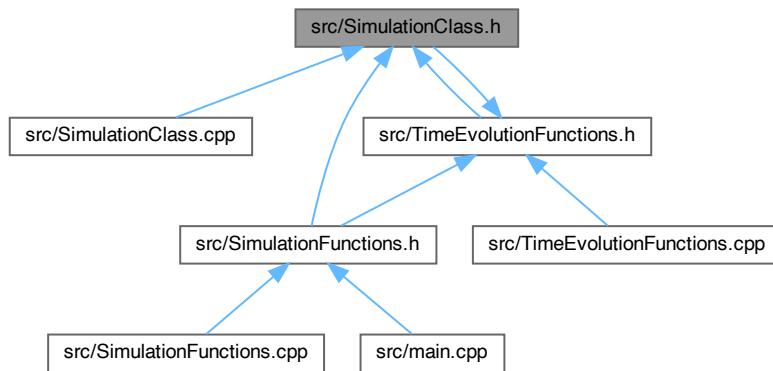
6.52 src/SimulationClass.h File Reference

Class for the **Simulation** object calling all functionality: from wave settings over lattice construction, time evolution and outputs initialization of the CVode object.

```
#include "sunnonlinsol/sunnonlinsol_fixedpoint.h"
#include "ICSetters.h"
#include "LatticePatch.h"
#include "Outputters.h"
#include "TimeEvolutionFunctions.h"
Include dependency graph for SimulationClass.h:
```



This graph shows which files directly or indirectly include this file:



Data Structures

- class [Simulation](#)
Simulation class to instantiate the whole walkthrough of a [Simulation](#).

Variables

- constexpr unsigned int [LatticeDiscreteSetUp](#) = 0x01
- constexpr unsigned int [LatticePhysicalSetUp](#) = 0x02
- constexpr unsigned int [LatticePatchworkSetUp](#) = 0x04
- constexpr unsigned int [CvodeObjectSetUp](#) = 0x08
- constexpr unsigned int [SimulationStarted](#) = 0x10

6.52.1 Detailed Description

Class for the [Simulation](#) object calling all functionality: from wave settings over lattice construction, time evolution and outputs initialization of the Cvode object.

Definition in file [SimulationClass.h](#).

6.52.2 Variable Documentation

6.52.2.1 CvodeObjectSetUp

```
constexpr unsigned int CvodeObjectSetUp = 0x08 [constexpr]
```

simulation checking flag

Definition at line 24 of file [SimulationClass.h](#).

Referenced by [Simulation::checkFlag\(\)](#), [Simulation::checkNoFlag\(\)](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::start\(\)](#), and [Simulation::~Simulation\(\)](#).

6.52.2.2 LatticeDiscreteSetUp

```
constexpr unsigned int LatticeDiscreteSetUp = 0x01 [constexpr]
```

simulation checking flag

Definition at line 21 of file [SimulationClass.h](#).

Referenced by [Simulation::checkFlag\(\)](#), [Simulation::checkNoFlag\(\)](#), [Simulation::initializePatchwork\(\)](#), [Simulation::setDiscreteDimensions\(\)](#), and [Simulation::start\(\)](#).

6.52.2.3 LatticePatchworkSetUp

```
constexpr unsigned int LatticePatchworkSetUp = 0x04 [constexpr]
```

simulation checking flag

Definition at line 23 of file [SimulationClass.h](#).

Referenced by [Simulation::checkFlag\(\)](#), [Simulation::checkNoFlag\(\)](#), [Simulation::initializePatchwork\(\)](#), [Simulation::setDiscreteDimensions\(\)](#), [Simulation::setPhysicalDimensionsOfLattice\(\)](#), and [Simulation::start\(\)](#).

6.52.2.4 LatticePhysicalSetUp

```
constexpr unsigned int LatticePhysicalSetUp = 0x02 [constexpr]
```

simulation checking flag

Definition at line 22 of file [SimulationClass.h](#).

Referenced by [Simulation::checkFlag\(\)](#), [Simulation::checkNoFlag\(\)](#), [Simulation::initializePatchwork\(\)](#), [Simulation::setPhysicalDimensions\(\)](#), and [Simulation::start\(\)](#).

6.52.2.5 SimulationStarted

```
constexpr unsigned int SimulationStarted = 0x10 [constexpr]
```

simulation checking flag

Definition at line 25 of file [SimulationClass.h](#).

Referenced by [Simulation::advanceToTime\(\)](#), [Simulation::checkFlag\(\)](#), [Simulation::checkNoFlag\(\)](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::outAllFieldData\(\)](#), and [Simulation::start\(\)](#).

6.53 SimulationClass.h

[Go to the documentation of this file.](#)

```
00001 //////////////////////////////////////////////////////////////////
00002 /// @file SimulationClass.h
00003 /// @brief Class for the Simulation object calling all functionality:
00004 /// from wave settings over lattice construction, time evolution and outputs
00005 /// initialization of the CVode object
00006 //////////////////////////////////////////////////////////////////
00007
00008 #pragma once
00009
00010 /* access to the fixed point SUNNonlinearSolver */
00011 #include "sunnonlinsol/sunnonlinsol_fixedpoint.h"
00012
00013 // project subfile headers
00014 #include "ICSetters.h"
00015 #include "LatticePatch.h"
00016 #include "Outputters.h"
00017 #include "TimeEvolutionFunctions.h"
00018
00019 /**
00020 ** simulation checking flag */
00021 constexpr unsigned int LatticeDiscreteSetUp = 0x01;
00022 constexpr unsigned int LatticePhysicalSetUp = 0x02;
00023 constexpr unsigned int LatticePatchworkSetUp = 0x04; // not used anymore
00024 constexpr unsigned int CvodeObjectSetUp = 0x08;
00025 constexpr unsigned int SimulationStarted = 0x10;
00026 /**
00027
00028 /** @brief Simulation class to instantiate the whole walkthrough of a Simulation
00029 */
00030 class Simulation {
00031 private:
00032     /// Lattice object
00033     Lattice lattice;
00034     /// LatticePatch object
00035     LatticePatch latticePatch;
00036     /// current time of the simulation
00037     sunrealtype t;
00038     /// simulation status flags
00039     unsigned int statusFlags;
00040
00041 public:
00042     /// IC Setter object
00043     ICSetter icsettings;
00044     /// Output Manager object
00045     OutputManager outputManager;
00046     /// pointer to CVode memory object
00047     void *cvode_mem;
00048     /// nonlinear solver object
00049     SUNNonlinearSolver NLS;
00050     /// constructor function for the creation of the cartesian communicator
00051     Simulation(const int Nx, const int Ny, const int Nz, const int StencilOrder,
00052                 const bool periodicity);
00053     /// destructor function freeing CVode memory and Sundials context
00054     ~Simulation();
00055     /// reference to the cartesian communicator of the lattice (for debugging)
00056     MPI_Comm *get_cart_comm() { return &lattice.comm; }
00057     /// function to set discrete dimensions of the lattice
00058     void setDiscreteDimensionsOfLattice(const sunindextype _tot_nx,
00059                                         const sunindextype _tot_ny, const sunindextype _tot_nz);
00060     /// function to set physical dimensions of the lattice
00061     void setPhysicalDimensionsOfLattice(const sunrealtype lx,
00062                                         const sunrealtype ly, const sunrealtype lz);
```

```

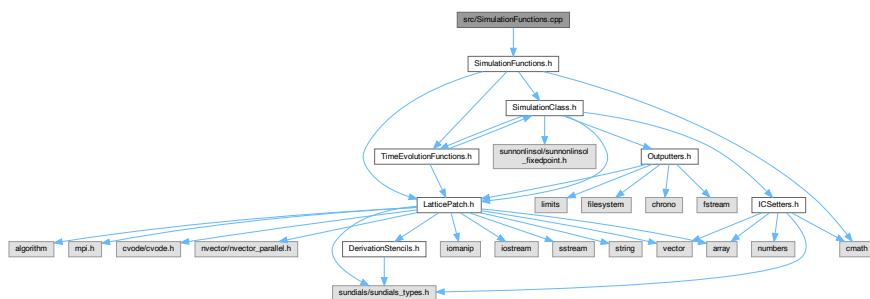
00063  /// function to initialize the Patchwork
00064  void initializePatchwork(const int nx, const int ny, const int nz);
00065  /// function to initialize the CVODE object with all requirements
00066  void initializeCVODEobject(const sunrealtype reltol,
00067           const sunrealtype abstol);
00068  /// function to start the simulation for time iteration
00069  void start();
00070  /// functions to set the initial field configuration onto the lattice
00071  void setInitialConditions();
00072  /// functions to add initial periodic field configurations
00073  void addInitialConditions(const sunindextype xm, const sunindextype ym,
00074           const sunindextype zm = 0);
00075  /// function to add a periodic IC layer in one dimension
00076  void addPeriodicICLayerInX();
00077  /// function to add periodic IC layers in two dimensions
00078  void addPeriodicICLayerInXY();
00079  /// function to advance solution in time with CVODE
00080  void advanceToTime(const sunrealtype &tEnd);
00081  /// function to write field data to disk
00082  void outAllFieldData(const int & state);
00083  /// function to check if flag has been set
00084  void checkFlag(unsigned int flag) const;
00085  /// function to check if flag has not been set
00086  // message and cause an abort on all ranks
00087  void checkNoFlag(unsigned int flag) const;
00088 };
00089

```

6.54 src/SimulationFunctions.cpp File Reference

Implementation of the complete simulation functions for 1D, 2D, and 3D, as called in the main function.

```
#include "SimulationFunctions.h"
Include dependency graph for SimulationFunctions.cpp:
```



Functions

- void **timer** (double &t1, double &t2)

MPI timer function.
- void **Sim1D** (const std::array< sunrealtype, 2 > CVodeTol, const int StencilOrder, const sunrealtype phys_← dim, const sunindextype disc_dim, const bool periodic, int *interactions, const sunrealtype endTime, const int numberOfSteps, const std::string outputDirectory, const int outputStep, const char outputStyle, const std← ::vector< **planewave** > &planes, const std::vector< **gaussian1D** > &gaussians)

complete 1D Simulation function
- void **Sim2D** (const std::array< sunrealtype, 2 > CVodeTol, int const StencilOrder, const std::array< sunrealtype, 2 > phys_dims, const std::array< sunindextype, 2 > disc_dims, const std::array< int, 2 > patches, const bool periodic, int *interactions, const sunrealtype endTime, const int numberOfSteps, const std::string outputDirectory, const int outputStep, const char outputStyle, const std::vector< **planewave** > &planes, const std::vector< **gaussian2D** > &gaussians)

- void **Sim3D** (const std::array< sunrealtype, 2 > CVodeTol, const int StencilOrder, const std::array< sunrealtype, 3 > phys_dims, const std::array< sunindextype, 3 > disc_dims, const std::array< int, 3 > patches, const bool periodic, int *interactions, const sunrealtype endTime, const int numberofSteps, const std::string outputDirectory, const int outputStep, const char outputStyle, const std::vector< planewave > &planes, const std::vector< gaussian3D > &gaussians)
- complete 3D Simulation function*

6.54.1 Detailed Description

Implementation of the complete simulation functions for 1D, 2D, and 3D, as called in the main function.

Definition in file [SimulationFunctions.cpp](#).

6.54.2 Function Documentation

6.54.2.1 Sim1D()

```
void Sim1D (
    const std::array< sunrealtype, 2 > CVodeTol,
    const int StencilOrder,
    const sunrealtype phys_dim,
    const sunindextype disc_dim,
    const bool periodic,
    int * interactions,
    const sunrealtype endTime,
    const int numberofSteps,
    const std::string outputDirectory,
    const int outputStep,
    const char outputStyle,
    const std::vector< planewave > & planes,
    const std::vector< gaussian1D > & gaussians )
```

complete 1D [Simulation function](#)

Conduct the complete 1D simulation process

Definition at line 21 of file [SimulationFunctions.cpp](#).

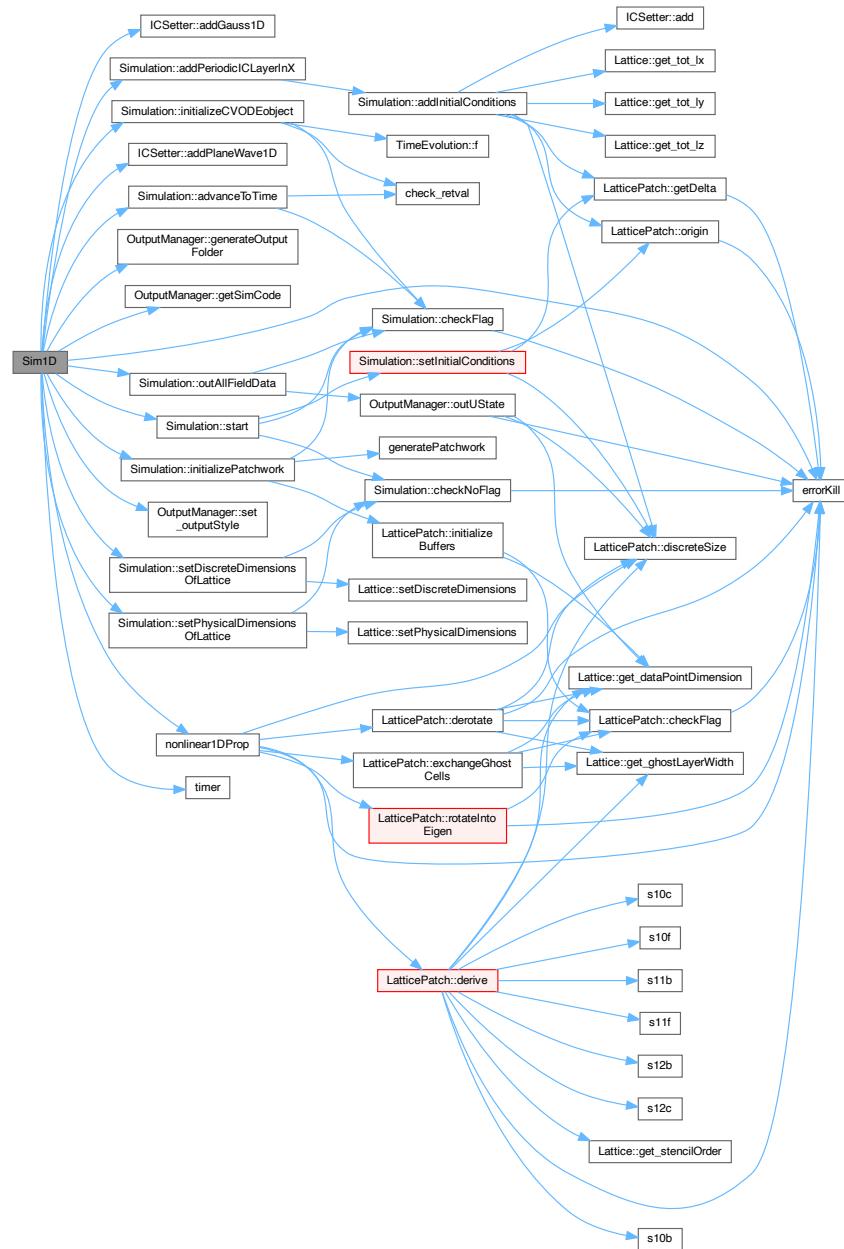
```
00028
00029
00030 // MPI data
00031 double ts = MPI_Wtime();
00032 int myPrc = 0, nPrc = 0;
00033 MPI_Comm_size(MPI_COMM_WORLD, &nPrc);
00034 MPI_Comm_rank(MPI_COMM_WORLD, &myPrc);
00035
00036 // Check feasibility of the patchwork decomposition
00037 if (myPrc == 0) {
00038     if (disc_dim % nPrc != 0) {
00039         errorKill("The number of lattice points must be "
00040                 "divisible by the number of processes.");
00041     }
00042 }
00043
00044 // Initialize the simulation, set up the cartesian communicator
00045 std::array<int, 3> patches = {nPrc, 1, 1};
00046 Simulation sim(patches[0], patches[1], patches[2], StencilOrder, periodic);
```

```

00047
00048 // Configure the patchwork
00049 sim.setPhysicalDimensionsOfLattice(phys_dim,1,1);
00050 sim.setDiscreteDimensionsOfLattice(disc_dim,1,1);
00051 sim.initializePatchwork(patches[0], patches[1], patches[2]);
00052
00053 // Add em-waves
00054 for (const auto &gauss : gaussians)
00055     sim.icsettings.addGauss1D(gauss.k, gauss.p, gauss.x0, gauss.phig,
00056                                 gauss.phi);
00057 for (const auto &plane : planes)
00058     sim.icsettings.addPlaneWave1D(plane.k, plane.p, plane.phi);
00059
00060 // Check that the patchwork is ready and set the initial conditions
00061 sim.start();
00062 sim.addPeriodicICLayerInX();
00063
00064 // Initialize CVode with abs and rel tolerances
00065 sim.initializeCVODEobject(CVodeTol[0], CVodeTol[1]);
00066
00067 // Configure the time evolution function
00068 TimeEvolution::c = interactions;
00069 TimeEvolution::TimeEvolver = nonlinear1DProp;
00070
00071 // Configure the output
00072 sim.outputManager.generateOutputFolder(outputDirectory);
00073 if (!myPrc) {
00074     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00075             << std::endl;
00076 }
00077 sim.outputManager.set_outputStyle(outputStyle);
00078
00079 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00080 //sim.outAllFieldData(0); // output of initial state
00081 // Conduct the propagation in space and time
00082 for (int step = 1; step <= numberOfSteps; step++) {
00083     sim.advanceToTime(endTime / numberOfSteps * step);
00084     if (step % outputStep == 0) {
00085         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00086         sim.outAllFieldData(step);
00087     }
00088     double tn = MPI_Wtime();
00089     if (!myPrc) {
00090         std::cout << "\rStep " << step << "\t\t" << std::flush;
00091         timer(ts, tn);
00092     }
00093 }
00094 return;
00095 }
```

References [ICSetter::addGauss1D\(\)](#), [Simulation::addPeriodicICLayerInX\(\)](#), [ICSetter::addPlaneWave1D\(\)](#), [Simulation::advanceToTime\(\)](#), [TimeEvolution::c](#), [errorKill\(\)](#), [OutputManager::generateOutputFolder\(\)](#), [OutputManager::getSimCode\(\)](#), [Simulation::icsettings](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::initializePatchwork\(\)](#), [nonlinear1DProp\(\)](#), [Simulation::outAllFieldData\(\)](#), [Simulation::outputManager](#), [OutputManager::set_outputStyle\(\)](#), [Simulation::setDiscreteDimensionsOfLattice\(\)](#), [Simulation::setPhysicalDimensionsOfLattice\(\)](#), [Simulation::start\(\)](#), [TimeEvolution::TimeEvolver](#), and [timer\(\)](#).

Here is the call graph for this function:



6.54.2.2 Sim2D()

```

void Sim2D (
    const std::array< sunrealtype, 2 > CVodeTol,
    int const StencilOrder,
    const std::array< sunrealtype, 2 > phys_dims,
    const std::array< sunindextype, 2 > disc_dims,
    const std::array< int, 2 > patches,

```

```

        const bool periodic,
        int * interactions,
        const sunrealtype endTime,
        const int numberOfSteps,
        const std::string outputDirectory,
        const int outputStep,
        const char outputStyle,
        const std::vector< planewave > & planes,
        const std::vector< gaussian2D > & gaussians )

```

complete 2D Simulation function

Conduct the complete 2D simulation process

Definition at line 99 of file [SimulationFunctions.cpp](#).

```

00107
00108
00109 // MPI data
00110 double ts = MPI_Wtime();
00111 int myPrc = 0, nPrc = 0; // Get process rank and number of processes
00112 MPI_Comm_rank(MPI_COMM_WORLD,
00113                 &myPrc); // Return process rank, number \in [1,nPrc]
00114 MPI_Comm_size(MPI_COMM_WORLD,
00115                 &nPrc); // Return number of processes (communicator size)
00116
00117 // Check feasibility of the patchwork decomposition
00118 if (myPrc == 0) {
00119     if (nPrc != patches[0] * patches[1]) {
00120         errorKill(
00121             "The number of MPI processes must match the number of patches.");
00122     }
00123 }
00124
00125 // Initialize the simulation, set up the cartesian communicator
00126 Simulation sim(patches[0], patches[1], 1, StencilOrder, periodic);
00127
00128 // Configure the patchwork
00129 sim.setPhysicalDimensionsOfLattice(phys_dims[0],
00130                                     phys_dims[1],
00131                                     1); // spacing of the lattice
00132 sim.setDiscreteDimensionsOfLattice(
00133     disc_dims[0], disc_dims[1], 1); // Spacing equivalence to points
00134 sim.initializePatchwork(patches[0], patches[1], 1);
00135
00136 // Add em-waves
00137 for (const auto &gauss : gaussians)
00138     sim.icsettings.addGauss2D(gauss.x0, gauss.axis, gauss.amp, gauss.phip,
00139                               gauss.w0, gauss.zr, gauss.ph0, gauss.phA);
00140     for (const auto &plane : planes)
00141         sim.icsettings.addPlaneWave2D(plane.k, plane.p, plane.phi);
00142
00143 // Check that the patchwork is ready and set the initial conditions
00144 sim.start(); // Check if the lattice is set up, set initial field
00145             // configuration
00146 sim.addPeriodicICLayerInXY(); // insure periodicity in propagation directions
00147
00148 // Initialize CVode with rel and abs tolerances
00149 sim.initializeCVODEObject(CVodeTol[0], CVodeTol[1]);
00150
00151 // Configure the time evolution function
00152 TimeEvolution::c = interactions;
00153 TimeEvolution::TimeEvolver = nonlinear2DProp;
00154
00155 // Configure the output
00156 sim.outputManager.generateOutputFolder(outputDirectory);
00157 if (!myPrc) {
00158     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00159             << std::endl;
00160 }
00161 sim.outputManager.set_outputStyle(outputStyle);
00162
00163 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00164 //sim.outAllFieldData(0); // output of initial state
00165 // Conduct the propagation in space and time
00166 for (int step = 1; step <= numberOfSteps; step++) {
00167     sim.advanceToTime(endTime / numberOfSteps * step);
00168     if (step % outputStep == 0) {
00169         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00170         sim.outAllFieldData(step);
00171     }

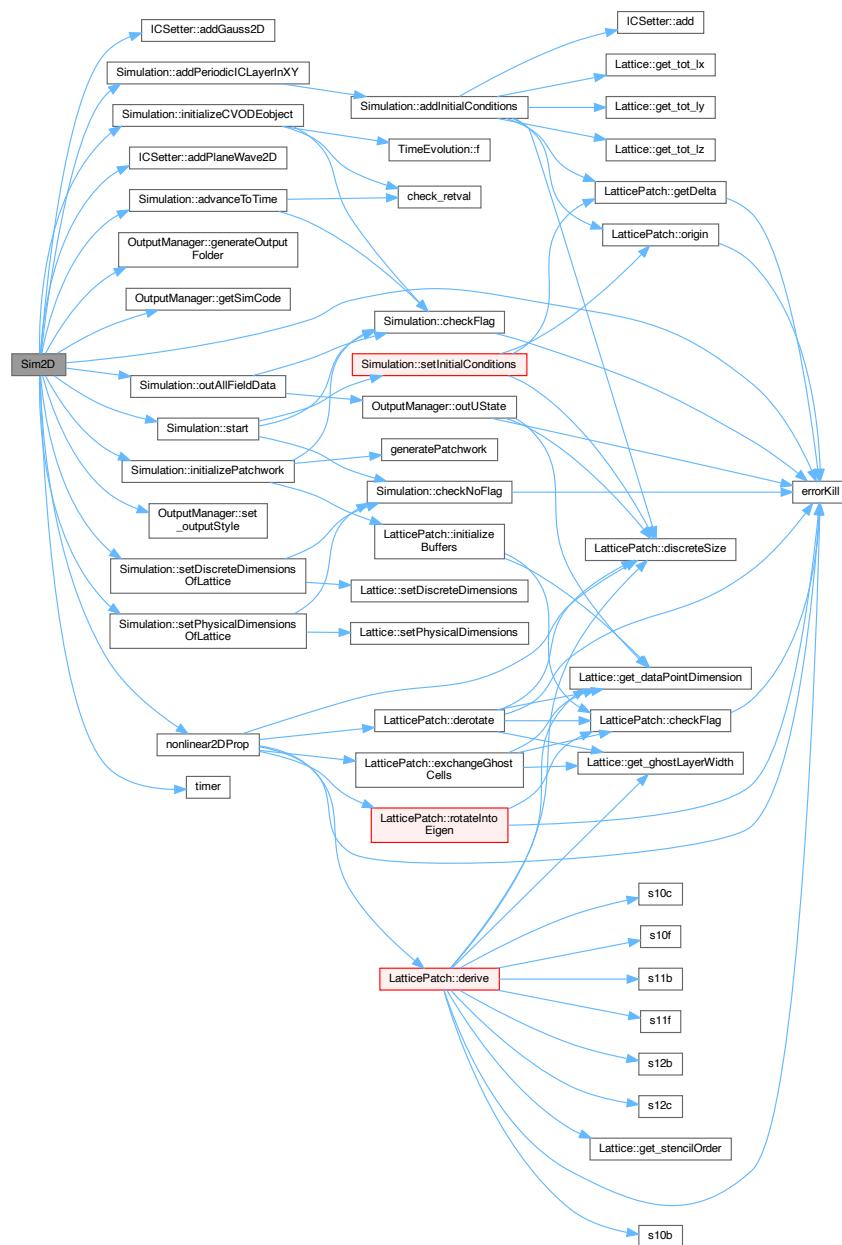
```

```

00172     double tn = MPI_Wtime();
00173     if (!myPrc) {
00174         std::cout << "\rStep " << step << "\t\t" << std::flush;
00175         timer(ts, tn);
00176     }
00177 }
00178
00179 return;
00180 }
```

References [ICSetter::addGauss2D\(\)](#), [Simulation::addPeriodicCLayerInXY\(\)](#), [ICSetter::addPlaneWave2D\(\)](#), [Simulation::advanceToTime\(\)](#), [TimeEvolution::c_errorKill\(\)](#), [OutputManager::generateOutputFolder\(\)](#), [OutputManager::getSimCode\(\)](#), [Simulation::icsettings](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::initializePatchwork\(\)](#), [nonlinear2DProp\(\)](#), [Simulation::outAllFieldData\(\)](#), [Simulation::outputManager](#), [OutputManager::set_outputStyle\(\)](#), [Simulation::setDiscreteDimensionsOfLattice\(\)](#), [Simulation::setPhysicalDimensionsOfLattice\(\)](#), [Simulation::start\(\)](#), [TimeEvolution::TimeEvolver](#), and [timer\(\)](#).

Here is the call graph for this function:



6.54.2.3 Sim3D()

```
void Sim3D (
    const std::array< sunrealtype, 2 > CVodeTol,
    const int StencilOrder,
    const std::array< sunrealtype, 3 > phys_dims,
    const std::array< sunindextype, 3 > disc_dims,
    const std::array< int, 3 > patches,
    const bool periodic,
    int * interactions,
    const sunrealtype endTime,
    const int numberofSteps,
    const std::string outputDirectory,
    const int outputStep,
    const char outputStyle,
    const std::vector< planewave > & planes,
    const std::vector< gaussian3D > & gaussians )
```

complete 3D Simulation function

Conduct the complete 3D simulation process

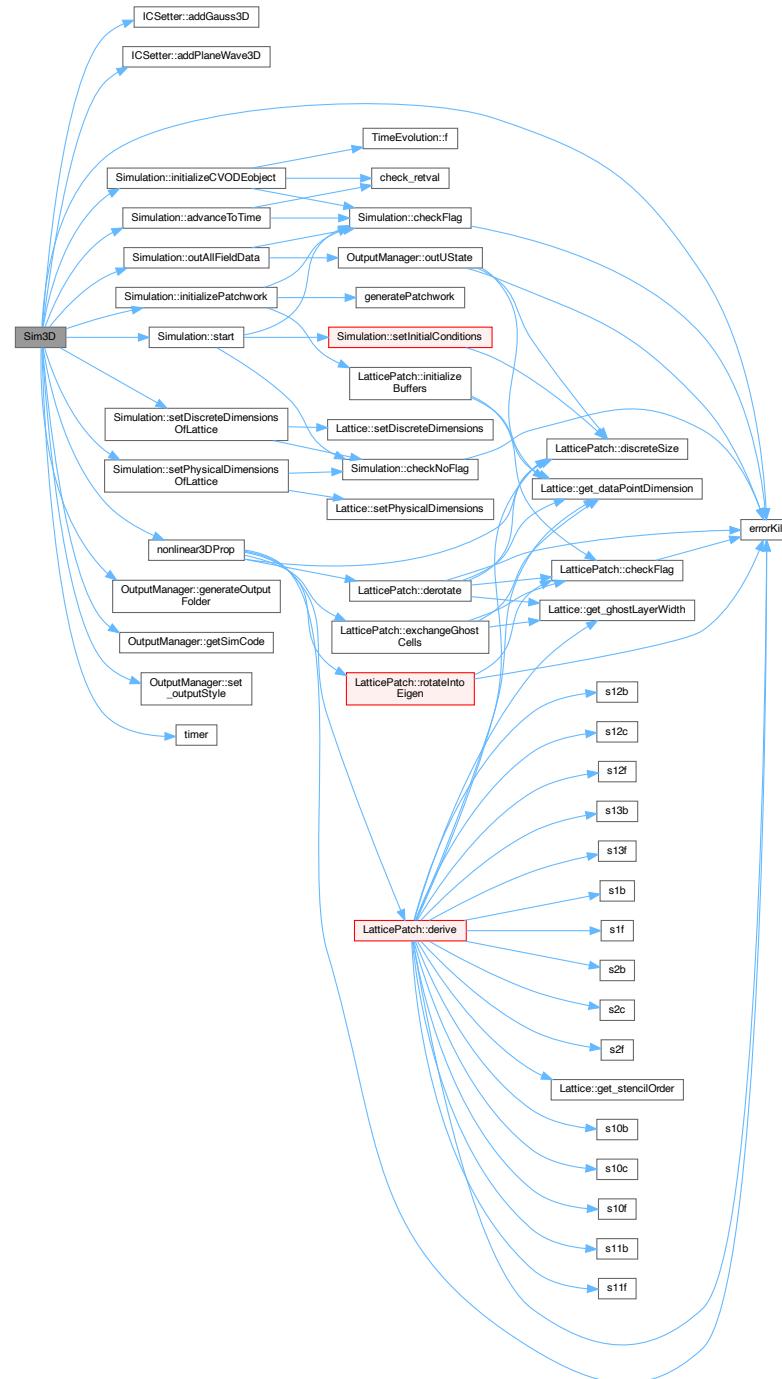
Definition at line 183 of file [SimulationFunctions.cpp](#).

```
00191
00192
00193 // MPI data
00194 double ts = MPI_Wtime();
00195 int myPrc = 0, nPrc = 0; // Get process rank and numer of process
00196 MPI_Comm_rank(MPI_COMM_WORLD,
00197     &myPrc); // rank of the process inside the world communicator
00198 MPI_Comm_size(MPI_COMM_WORLD,
00199     &nPrc); // Size of the communicator is the number of processes
00200
00201 // Check feasibility of the patchwork decomposition
00202 if (myPrc == 0) {
00203     if (nPrc != patches[0] * patches[1] * patches[2]) {
00204         errorKill(
00205             "The number of MPI processes must match the number of patches.");
00206     }
00207     /*
00208     if ( ( disc_dims[0] / patches[0] != disc_dims[1] / patches[1] ) ||
00209         ( disc_dims[0] / patches[0] != disc_dims[2] / patches[2] ) ) {
00210         std::clog
00211             << "\nWarning: Patches should be cubic in terms of the lattice "
00212             "points for the computational efficiency of larger simulations.\n";
00213     }
00214 */
00215 }
00216
00217 // Initialize the simulation, set up the cartesian communicator
00218 Simulation sim(patches[0], patches[1], patches[2],
00219                 StencilOrder, periodic); // simulation object with slicing
00220
00221 // Create the SUNContext object associated with the thread of execution
00222 sim.setPhysicalDimensionsOfLattice(phys_dims[0], phys_dims[1],
00223                                     phys_dims[2]); // spacing of the box
00224 sim.setDiscreteDimensionsOfLattice(
00225     disc_dims[0], disc_dims[1],
00226     disc_dims[2]); // Spacing equivalence to points
00227 sim.initializePatchwork(patches[0], patches[1], patches[2]);
00228
00229 // Add em-waves
00230 for (const auto &plane : planes)
00231     sim.icsettings.addPlaneWave3D(plane.k, plane.p, plane.phi);
00232 for (const auto &gauss : gaussians)
00233     sim.icsettings.addGauss3D(gauss.x0, gauss.axis, gauss.amp, gauss.phip,
00234                               gauss.w0, gauss.zr, gauss.ph0, gauss.phA);
00235
00236 // Check that the patchwork is ready and set the initial conditions
00237 sim.start();
00238
```

```
00239 // Initialize CVode with abs and rel tolerances
00240 sim.initializeCVODEobject(CVodeTol[0], CVodeTol[1]);
00241
00242 // Configure the time evolution function
00243 TimeEvolution::c = interactions;
00244 TimeEvolution::TimeEvolver = nonlinear3DProp;
00245
00246 // Configure the output
00247 sim.outputManager.generateOutputFolder(outputDirectory);
00248 if (!myPrc) {
00249     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00250         << std::endl;
00251 }
00252 sim.outputManager.set_outputStyle(outputStyle);
00253
00254 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00255 //sim.outAllFieldData(0); // output of initial state
00256 // Conduct the propagation in space and time
00257 for (int step = 1; step <= number_of_Steps; step++) {
00258     sim.advanceToTime(endTime / number_of_Steps * step);
00259     if (step % outputStep == 0) {
00260         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00261         sim.outAllFieldData(step);
00262     }
00263     double tn = MPI_Wtime();
00264     if (!myPrc) {
00265         std::cout << "\rStep " << step << "\t\t" << std::flush;
00266         timer(ts, tn);
00267     }
00268 }
00269 return;
00270 }
```

References [ICSetter::addGauss3D\(\)](#), [ICSetter::addPlaneWave3D\(\)](#), [Simulation::advanceToTime\(\)](#), [TimeEvolution::c](#), [errorKill\(\)](#), [OutputManager::generateOutputFolder\(\)](#), [OutputManager::getSimCode\(\)](#), [Simulation::icsettings](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::initializePatchwork\(\)](#), [nonlinear3DProp\(\)](#), [Simulation::outAllFieldData\(\)](#), [Simulation::outputManager](#), [OutputManager::set_outputStyle\(\)](#), [Simulation::setDiscreteDimensionsOfLattice\(\)](#), [Simulation::setPhysicalDimensionsOfLattice\(\)](#), [Simulation::start\(\)](#), [TimeEvolution::TimeEvolver](#), and [timer\(\)](#).

Here is the call graph for this function:



6.54.2.4 timer()

```
void timer (
    double & t1,
    double & t2 ) [inline]
```

MPI timer function.

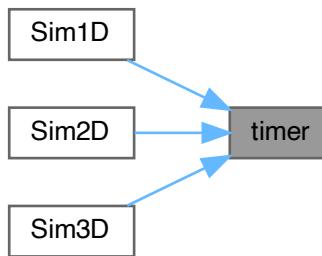
Calculate and print the total simulation time

Definition at line 10 of file [SimulationFunctions.cpp](#).

```
00010     printf("Elapsed time: %fs\n", (t2 - t1));
```

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the caller graph for this function:



6.55 SimulationFunctions.cpp

[Go to the documentation of this file.](#)

```
00001 //////////////////////////////////////////////////////////////////
00002 /// @file SimulationFunctions.cpp
00003 /// @brief Implementation of the complete simulation functions for
00004 /// 1D, 2D, and 3D, as called in the main function
00005 //////////////////////////////////////////////////////////////////
00006
00007 #include "SimulationFunctions.h"
00008
00009 /** Calculate and print the total simulation time */
00010 inline void timer(double &t1, double &t2) {
00011     printf("Elapsed time: %fs\n", (t2 - t1));
00012 }
00013
00014 // Instantiate and preliminarily initialize the time evolver
00015 // non-const statics to be defined in actual simulation process
00016 int *TimeEvolution::c = nullptr;
00017 void (*TimeEvolution::TimeEvolver)(LatticePatch *, N_Vector, N_Vector,
00018                                     int *) = nonlinear1DProp;
00019
00020 /** Conduct the complete 1D simulation process */
00021 void Sim1D(const std::array<sunrealtype,2> CVodeTol, const int StencilOrder,
00022             const unrealtype phys_dim, const sunindextype disc_dim,
00023             const bool periodic, int *interactions,
00024             const unrealtype endTime, const int numberOfSteps,
00025             const std::string outputDirectory, const int outputStep,
00026             const char outputStyle,
00027             const std::vector<planewave> &planes,
00028             const std::vector<gaussian1D> &gaussians) {
00029
00030     // MPI data
00031     double ts = MPI_Wtime();
00032     int myPrc = 0, nPrc = 0;
00033     MPI_Comm_size(MPI_COMM_WORLD, &nPrc);
00034     MPI_Comm_rank(MPI_COMM_WORLD, &myPrc);
00035
00036     // Check feasibility of the patchwork decomposition
00037     if (myPrc == 0) {
```

```

00038     if (disc_dim % nPrc != 0) {
00039         errorKill("The number of lattice points must be "
00040                 "divisible by the number of processes.");
00041     }
00042 }
00043
00044 // Initialize the simulation, set up the cartesian communicator
00045 std::array<int, 3> patches = {nPrc, 1, 1};
00046 Simulation sim(patches[0], patches[1], patches[2], StencilOrder, periodic);
00047
00048 // Configure the patchwork
00049 sim.setPhysicalDimensionsOfLattice(phys_dim,1,1);
00050 sim.setDiscreteDimensionsOfLattice(disc_dim,1,1);
00051 sim.initializePatchwork(patches[0], patches[1], patches[2]);
00052
00053 // Add em-waves
00054 for (const auto &gauss : gaussians)
00055     sim.icsettings.addGauss1D(gauss.k, gauss.p, gauss.x0, gauss.phig,
00056                                 gauss.phi);
00057 for (const auto &plane : planes)
00058     sim.icsettings.addPlaneWave1D(plane.k, plane.p, plane.phi);
00059
00060 // Check that the patchwork is ready and set the initial conditions
00061 sim.start();
00062 sim.addPeriodicICLayerInX();
00063
00064 // Initialize CVode with abs and rel tolerances
00065 sim.initializeCVODEObject(CVodeTol[0], CVodeTol[1]);
00066
00067 // Configure the time evolution function
00068 TimeEvolution::c = interactions;
00069 TimeEvolution::TimeEvolver = nonlinear1DProp;
00070
00071 // Configure the output
00072 sim.outputManager.generateOutputFolder(outputDirectory);
00073 if (!myPrc) {
00074     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00075             << std::endl;
00076 }
00077 sim.outputManager.setOutputStyle(outputStyle);
00078
00079 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00080 //sim.outAllFieldData(0); // output of initial state
00081 //Conduct the propagation in space and time
00082 for (int step = 1; step <= numberofSteps; step++) {
00083     sim.advanceToTime(endTime / numberofSteps * step);
00084     if (step % outputStep == 0) {
00085         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00086         sim.outAllFieldData(step);
00087     }
00088     double tn = MPI_Wtime();
00089     if (!myPrc) {
00090         std::cout << "\rStep " << step << "\t\t" << std::flush;
00091         timer(ts, tn);
00092     }
00093 }
00094
00095 return;
00096 }
00097
00098 /** Conduct the complete 2D simulation process */
00099 void Sim2D(const std::array<sunrealtype,2> CVodeTol, int const StencilOrder,
00100             const std::array<sunrealtype,2> phys_dims,
00101             const std::array<sunindextype,2> disc_dims,
00102             const std::array<int,2> patches, const bool periodic, int *interactions,
00103             const unrealtype endTime, const int numberofSteps,
00104             const std::string outputDirectory, const int outputStep,
00105             const char outputStyle,
00106             const std::vector<planewave> &planes,
00107             const std::vector<gaussian2D> &gaussians) {
00108
00109 // MPI data
00110 double ts = MPI_Wtime();
00111 int myPrc = 0, nPrc = 0; // Get process rank and number of processes
00112 MPI_Comm_rank(MPI_COMM_WORLD,
00113                 &myPrc); // Return process rank, number \in [1,nPrc]
00114 MPI_Comm_size(MPI_COMM_WORLD,
00115                 &nPrc); // Return number of processes (communicator size)
00116
00117 // Check feasibility of the patchwork decomposition
00118 if (myPrc == 0) {
00119     if (nPrc != patches[0] * patches[1]) {
00120         errorKill(
00121             "The number of MPI processes must match the number of patches.");
00122     }
00123 }
00124

```

```

00125 // Initialize the simulation, set up the cartesian communicator
00126 Simulation sim(patches[0], patches[1], 1, StencilOrder, periodic);
00127
00128 // Configure the patchwork
00129 sim.setPhysicalDimensionsOfLattice(phys_dims[0],
00130                                     phys_dims[1],
00131                                     1); // spacing of the lattice
00132 sim.setDiscreteDimensionsOfLattice(
00133     disc_dims[0], disc_dims[1], 1); // Spacing equivalence to points
00134 sim.initializePatchwork(patches[0], patches[1], 1);
00135
00136 // Add em-waves
00137 for (const auto &gauss : gaussians)
00138     sim.icsettings.addGauss2D(gauss.x0, gauss.axis, gauss.amp, gauss.phip,
00139                               gauss.w0, gauss.zr, gauss.ph0, gauss.phA);
00140 for (const auto &plane : planes)
00141     sim.icsettings.addPlaneWave2D(plane.k, plane.p, plane.phi);
00142
00143 // Check that the patchwork is ready and set the initial conditions
00144 sim.start(); // Check if the lattice is set up, set initial field
00145             // configuration
00146 sim.addPeriodicICLayerInXY(); // insure periodicity in propagation directions
00147
00148 // Initialize CVode with rel and abs tolerances
00149 sim.initializeCVODEObject(CVodeTol[0], CVodeTol[1]);
00150
00151 // Configure the time evolution function
00152 TimeEvolution::c = interactions;
00153 TimeEvolution::TimeEvolver = nonlinear2DProp;
00154
00155 // Configure the output
00156 sim.outputManager.generateOutputFolder(outputDirectory);
00157 if (!myPrc) {
00158     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00159             << std::endl;
00160 }
00161 sim.outputManager.setOutputStyle(outputStyle);
00162
00163 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00164 //sim.outAllFieldData(0); // output of initial state
00165 // Conduct the propagation in space and time
00166 for (int step = 1; step <= numberofSteps; step++) {
00167     sim.advanceToTime(endTime / numberofSteps * step);
00168     if (step % outputStep == 0) {
00169         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00170         sim.outAllFieldData(step);
00171     }
00172     double tn = MPI_Wtime();
00173     if (!myPrc) {
00174         std::cout << "\rStep " << step << "\t\t" << std::flush;
00175         timer(ts, tn);
00176     }
00177 }
00178
00179 return;
00180 }
00181
00182 /** Conduct the complete 3D simulation process */
00183 void Sim3D(const std::array<sunrealtype,2> CVodeTol, const int StencilOrder,
00184             const std::array<sunrealtype,3> phys_dims,
00185             const std::array<sunindextype,3> disc_dims,
00186             const std::array<int,3> patches,
00187             const bool periodic, int *interactions, const unrealtype endTime,
00188             const int numberofSteps, const std::string outputDirectory,
00189             const int outputStep, const char outputStyle,
00190             const std::vector<planewave> &planes,
00191             const std::vector<gaussian3D> &gaussians) {
00192
00193 // MPI data
00194 double ts = MPI_Wtime();
00195 int myPrc = 0, nPrc = 0; // Get process rank and numer of process
00196 MPI_Comm_rank(MPI_COMM_WORLD,
00197                 &myPrc); // rank of the process inside the world communicator
00198 MPI_Comm_size(MPI_COMM_WORLD,
00199                 &nPrc); // Size of the communicator is the number of processes
00200
00201 // Check feasibility of the patchwork decomposition
00202 if (myPrc == 0) {
00203     if (nPrc != patches[0] * patches[1] * patches[2]) {
00204         errorKill(
00205             "The number of MPI processes must match the number of patches.");
00206     }
00207     /*
00208     if ( ( disc_dims[0] / patches[0] != disc_dims[1] / patches[1] ) ||
00209         ( disc_dims[0] / patches[0] != disc_dims[2] / patches[2] ) ) {
00210         std::clog
00211             << "\nWarning: Patches should be cubic in terms of the lattice "

```

```

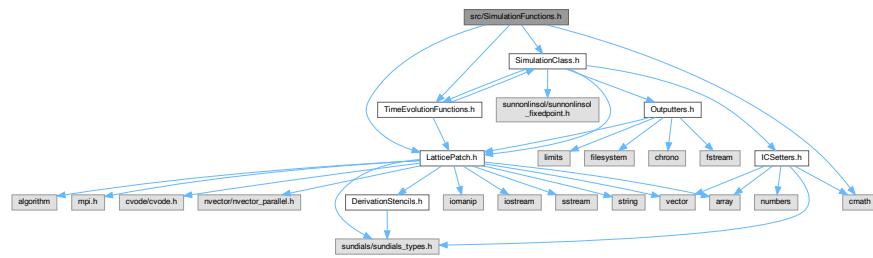
00212         "points for the computational efficiency of larger simulations.\n";
00213     }
00214     */
00215 }
00216
00217 // Initialize the simulation, set up the cartesian communicator
00218 Simulation sim(patches[0], patches[1], patches[2],
00219                 StencilOrder, periodic); // Simulation object with slicing
00220
00221 // Create the SUNContext object associated with the thread of execution
00222 sim.setPhysicalDimensionsOfLattice(phys_dims[0], phys_dims[1],
00223                                     phys_dims[2]); // spacing of the box
00224 sim.setDiscreteDimensionsOfLattice(
00225     disc_dims[0], disc_dims[1],
00226     disc_dims[2]); // Spacing equivalence to points
00227 sim.initializePatchwork(patches[0], patches[1], patches[2]);
00228
00229 // Add em-waves
00230 for (const auto &plane : planes)
00231     sim.icsettings.addPlaneWave3D(plane.k, plane.p, plane.phi);
00232 for (const auto &gauss : gaussians)
00233     sim.icsettings.addGauss3D(gauss.x0, gauss.axis, gauss.amp, gauss.phip,
00234                                gauss.w0, gauss.zr, gauss.ph0, gauss.phA);
00235
00236 // Check that the patchwork is ready and set the initial conditions
00237 sim.start();
00238
00239 // Initialize CVode with abs and rel tolerances
00240 sim.initializeCVODEObject(CVodeTol[0], CVodeTol[1]);
00241
00242 // Configure the time evolution function
00243 TimeEvolution::c = interactions;
00244 TimeEvolution::TimeEvolver = nonlinear3DProp;
00245
00246 // Configure the output
00247 sim.outputManager.generateOutputFolder(outputDirectory);
00248 if (!myPrc) {
00249     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00250             << std::endl;
00251 }
00252 sim.outputManager.setOutputStyle(outputStyle);
00253
00254 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00255 //sim.outAllFieldData(0); // output of initial state
00256 // Conduct the propagation in space and time
00257 for (int step = 1; step <= number_of_steps; step++) {
00258     sim.advanceToTime(endTime / number_of_steps * step);
00259     if (step % output_step == 0) {
00260         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00261         sim.outAllFieldData(step);
00262     }
00263     double tn = MPI_Wtime();
00264     if (!myPrc) {
00265         std::cout << "\rStep " << step << "\t\t" << std::flush;
00266         timer(ts, tn);
00267     }
00268 }
00269 return;
00270 }
```

6.56 src/SimulationFunctions.h File Reference

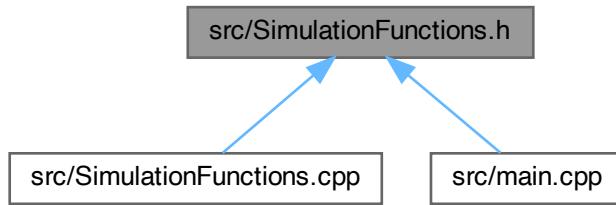
Full simulation functions for 1D, 2D, and 3D used in [main.cpp](#).

```
#include <cmath>
#include "LatticePatch.h"
#include "SimulationClass.h"
#include "TimeEvolutionFunctions.h"
```

Include dependency graph for SimulationFunctions.h:



This graph shows which files directly or indirectly include this file:



Data Structures

- struct **planewave**
plane wave structure
 - struct **gaussian1D**
1D Gaussian wave structure
 - struct **gaussian2D**
2D Gaussian wave structure
 - struct **gaussian3D**
3D Gaussian wave structure

Functions

- void **Sim1D** (const std::array< sunrealtype, 2 >, const int, const sunrealtype, const sunindextype, const bool, int *, const sunrealtype, const int, const std::string, const int, const char, const std::vector< **planewave** > &, const std::vector< **gaussian1D** > &)
complete 1D Simulation function
 - void **Sim2D** (const std::array< sunrealtype, 2 >, const int, const std::array< sunrealtype, 2 >, const std::array< sunindextype, 2 >, const std::array< int, 2 >, const bool, int *, const sunrealtype, const int, const std::string, const int, const char, const std::vector< **planewave** > &, const std::vector< **gaussian2D** > &)
complete 2D Simulation function

- void **Sim3D** (const std::array< sunrealtype, 2 >, const int, const std::array< sunrealtype, 3 >, const std::array< sunindextype, 3 >, const std::array< int, 3 >, const bool, int *, const sunrealtype, const int, const std::string, const int, const char, const std::vector< **planewave** > &, const std::vector< **gaussian3D** > &)

complete 3D Simulation function
- void **timer** (double &, double &)

MPI timer function.

6.56.1 Detailed Description

Full simulation functions for 1D, 2D, and 3D used in [main.cpp](#).

Definition in file [SimulationFunctions.h](#).

6.56.2 Function Documentation

6.56.2.1 Sim1D()

```
void Sim1D (
    const std::array< sunrealtype, 2 > CVodeTol,
    const int StencilOrder,
    const sunrealtype phys_dim,
    const sunindextype disc_dim,
    const bool periodic,
    int * interactions,
    const sunrealtype endTime,
    const int numberOfSteps,
    const std::string outputDirectory,
    const int outputStep,
    const char outputStyle,
    const std::vector< planewave > & planes,
    const std::vector< gaussian1D > & gaussians )
```

complete 1D [Simulation](#) function

Conduct the complete 1D simulation process

Definition at line 21 of file [SimulationFunctions.cpp](#).

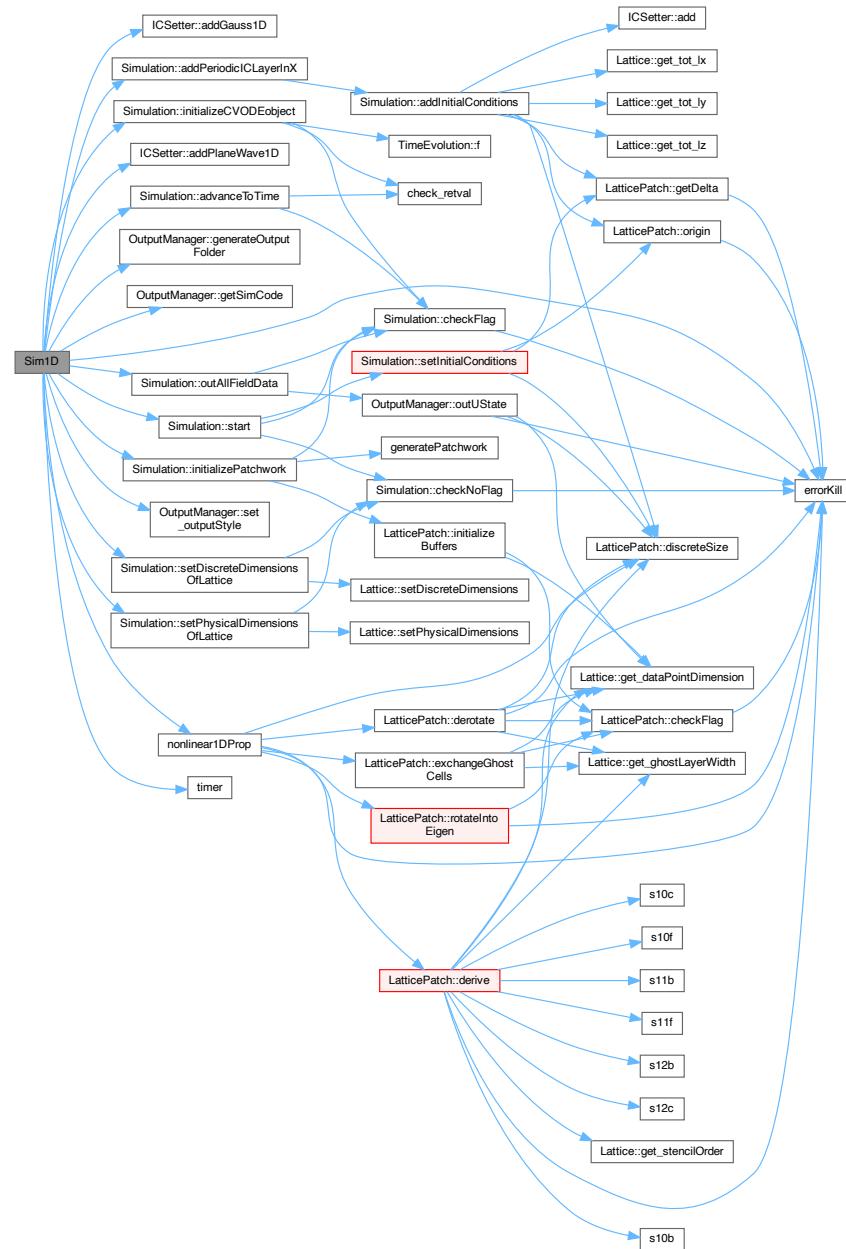
```
00028
00029
00030 // MPI data
00031 double ts = MPI_Wtime();
00032 int myPrc = 0, nPrc = 0;
00033 MPI_Comm_size(MPI_COMM_WORLD, &nPrc);
00034 MPI_Comm_rank(MPI_COMM_WORLD, &myPrc);
00035
00036 // Check feasibility of the patchwork decomposition
00037 if (myPrc == 0) {
00038     if (disc_dim % nPrc != 0) {
00039         errorKill("The number of lattice points must be "
00040                 "divisible by the number of processes.");
00041     }
00042 }
00043
00044 // Initialize the simulation, set up the cartesian communicator
00045 std::array<int, 3> patches = {nPrc, 1, 1};
00046 Simulation sim(patches[0], patches[1], patches[2], StencilOrder, periodic);
00047
```

```

00048 // Configure the patchwork
00049 sim.setPhysicalDimensionsOfLattice(phys_dim,1,1);
00050 sim.setDiscreteDimensionsOfLattice(disc_dim,1,1);
00051 sim.initializePatchwork(patches[0], patches[1], patches[2]);
00052
00053 // Add em-waves
00054 for (const auto &gauss : gaussians)
00055     sim.icsettings.addGauss1D(gauss.k, gauss.p, gauss.x0, gauss.phig,
00056                                 gauss.phi);
00057 for (const auto &plane : planes)
00058     sim.icsettings.addPlaneWave1D(plane.k, plane.p, plane.phi);
00059
00060 // Check that the patchwork is ready and set the initial conditions
00061 sim.start();
00062 sim.addPeriodicICLayerInX();
00063
00064 // Initialize CVode with abs and rel tolerances
00065 sim.initializeCVODEObject(CVodeTol[0], CVodeTol[1]);
00066
00067 // Configure the time evolution function
00068 TimeEvolution::c = interactions;
00069 TimeEvolution::TimeEvolver = nonlinear1DProp;
00070
00071 // Configure the output
00072 sim.outputManager.generateOutputFolder(outputDirectory);
00073 if (!myPrc) {
00074     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00075         << std::endl;
00076 }
00077 sim.outputManager.set_outputStyle(outputStyle);
00078
00079 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00080 //sim.outAllFieldData(0); // output of initial state
00081 // Conduct the propagation in space and time
00082 for (int step = 1; step <= numberOfSteps; step++) {
00083     sim.advanceToTime(endTime / numberOfSteps * step);
00084     if (step % outputStep == 0) {
00085         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00086         sim.outAllFieldData(step);
00087     }
00088     double tn = MPI_Wtime();
00089     if (!myPrc) {
00090         std::cout << "\rStep " << step << "\t\t" << std::flush;
00091         timer(ts, tn);
00092     }
00093 }
00094
00095 return;
00096 }
```

References [ICSetter::addGauss1D\(\)](#), [Simulation::addPeriodicICLayerInX\(\)](#), [ICSetter::addPlaneWave1D\(\)](#), [Simulation::advanceToTime\(\)](#), [TimeEvolution::c](#), [errorKill\(\)](#), [OutputManager::generateOutputFolder\(\)](#), [OutputManager::getSimCode\(\)](#), [Simulation::icsettings](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::initializePatchwork\(\)](#), [nonlinear1DProp\(\)](#), [Simulation::outAllFieldData\(\)](#), [Simulation::outputManager](#), [OutputManager::set_outputStyle\(\)](#), [Simulation::setDiscreteDimensionsOfLattice\(\)](#), [Simulation::setPhysicalDimensionsOfLattice\(\)](#), [Simulation::start\(\)](#), [TimeEvolution::TimeEvolver](#), and [timer\(\)](#).

Here is the call graph for this function:



6.56.2.2 Sim2D()

```
void Sim2D (
    const std::array< sunrealtype, 2 > CVodeTol,
    int const StencilOrder,
    const std::array< sunrealtype, 2 > phys_dims,
    const std::array< sunindextype, 2 > disc_dims,
    const std::array< int, 2 > patches,
```

```

        const bool periodic,
        int * interactions,
        const sunrealtype endTime,
        const int numberofSteps,
        const std::string outputDirectory,
        const int outputStep,
        const char outputStyle,
        const std::vector< planewave > & planes,
        const std::vector< gaussian2D > & gaussians )

```

complete 2D Simulation function

Conduct the complete 2D simulation process

Definition at line 99 of file [SimulationFunctions.cpp](#).

```

00107
00108
00109 // MPI data
00110 double ts = MPI_Wtime();
00111 int myPrc = 0, nPrc = 0; // Get process rank and number of processes
00112 MPI_Comm_rank(MPI_COMM_WORLD,
00113                 &myPrc); // Return process rank, number \in [1,nPrc]
00114 MPI_Comm_size(MPI_COMM_WORLD,
00115                 &nPrc); // Return number of processes (communicator size)
00116
00117 // Check feasibility of the patchwork decomposition
00118 if (myPrc == 0) {
00119     if (nPrc != patches[0] * patches[1]) {
00120         errorKill(
00121             "The number of MPI processes must match the number of patches.");
00122     }
00123 }
00124
00125 // Initialize the simulation, set up the cartesian communicator
00126 Simulation sim(patches[0], patches[1], 1, StencilOrder, periodic);
00127
00128 // Configure the patchwork
00129 sim.setPhysicalDimensionsOfLattice(phys_dims[0],
00130                                     phys_dims[1],
00131                                     1); // spacing of the lattice
00132 sim.setDiscreteDimensionsOfLattice(
00133     disc_dims[0], disc_dims[1], 1); // Spacing equivalence to points
00134 sim.initializePatchwork(patches[0], patches[1], 1);
00135
00136 // Add em-waves
00137 for (const auto &gauss : gaussians)
00138     sim.icsettings.addGauss2D(gauss.x0, gauss.axis, gauss.amp, gauss.phip,
00139                               gauss.w0, gauss.zr, gauss.ph0, gauss.phA);
00140     for (const auto &plane : planes)
00141         sim.icsettings.addPlaneWave2D(plane.k, plane.p, plane.phi);
00142
00143 // Check that the patchwork is ready and set the initial conditions
00144 sim.start(); // Check if the lattice is set up, set initial field
00145             // configuration
00146 sim.addPeriodicICLayerInXY(); // insure periodicity in propagation directions
00147
00148 // Initialize CVode with rel and abs tolerances
00149 sim.initializeCVODEObject(CVodeTol[0], CVodeTol[1]);
00150
00151 // Configure the time evolution function
00152 TimeEvolution::c = interactions;
00153 TimeEvolution::TimeEvolver = nonlinear2DProp;
00154
00155 // Configure the output
00156 sim.outputManager.generateOutputFolder(outputDirectory);
00157 if (!myPrc) {
00158     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00159             << std::endl;
00160 }
00161 sim.outputManager.set_outputStyle(outputStyle);
00162
00163 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00164 //sim.outAllFieldData(0); // output of initial state
00165 // Conduct the propagation in space and time
00166 for (int step = 1; step <= numberofSteps; step++) {
00167     sim.advanceToTime(endTime / numberofSteps * step);
00168     if (step % outputStep == 0) {
00169         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00170         sim.outAllFieldData(step);
00171     }

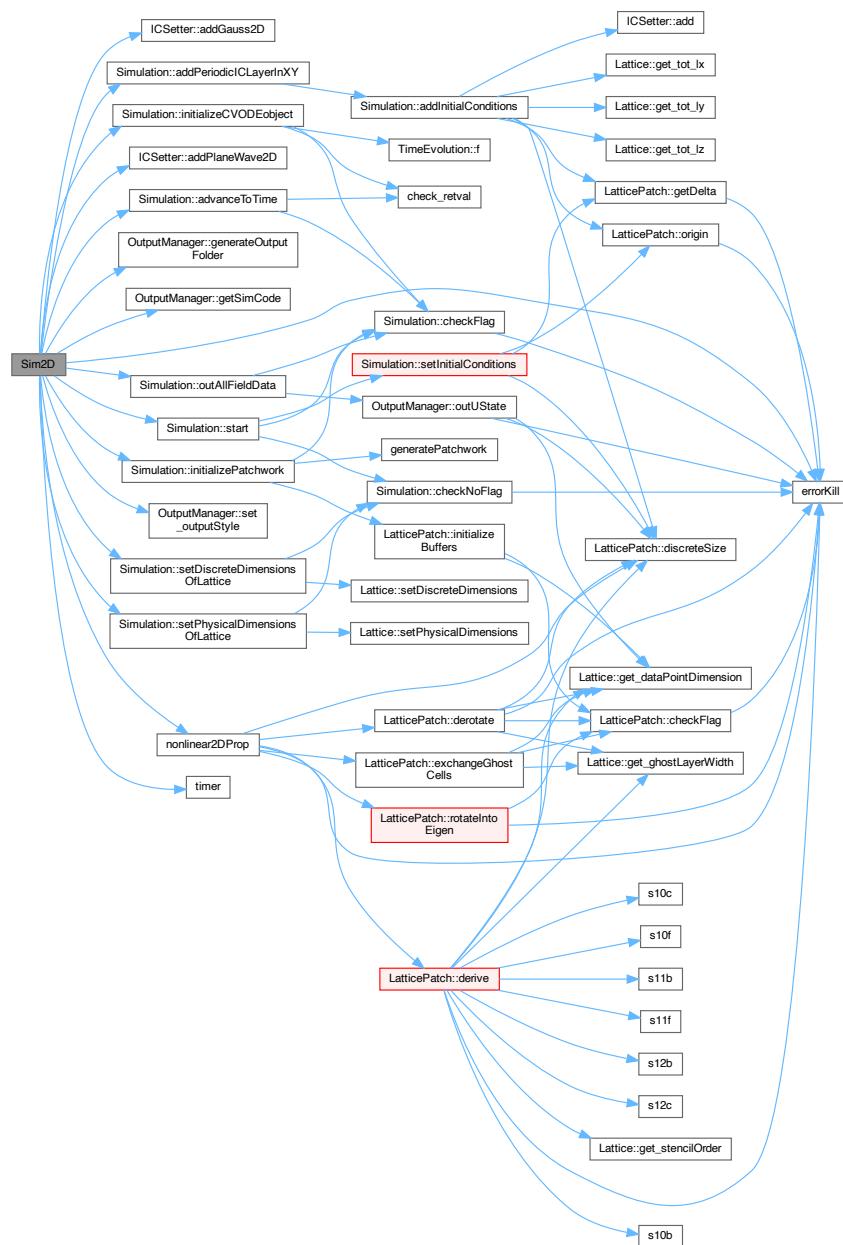
```

```

00172     double tn = MPI_Wtime();
00173     if (!myPrc) {
00174         std::cout << "\rStep " << step << "\t\t" << std::flush;
00175         timer(ts, tn);
00176     }
00177 }
00178
00179 return;
00180 }
```

References [ICSetter::addGauss2D\(\)](#), [Simulation::addPeriodicCLayerInXY\(\)](#), [ICSetter::addPlaneWave2D\(\)](#), [Simulation::advanceToTime\(\)](#), [TimeEvolution::c_errorKill\(\)](#), [OutputManager::generateOutputFolder\(\)](#), [OutputManager::getSimCode\(\)](#), [Simulation::icsettings](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::initializePatchwork\(\)](#), [nonlinear2DProp\(\)](#), [Simulation::outAllFieldData\(\)](#), [Simulation::outputManager](#), [OutputManager::set_outputStyle\(\)](#), [Simulation::setDiscreteDimensionsOfLattice\(\)](#), [Simulation::setPhysicalDimensionsOfLattice\(\)](#), [Simulation::start\(\)](#), [TimeEvolution::TimeEvolver](#), and [timer\(\)](#).

Here is the call graph for this function:



6.56.2.3 Sim3D()

```
void Sim3D (
    const std::array< sunrealtype, 2 > CVodeTol,
    const int StencilOrder,
    const std::array< sunrealtype, 3 > phys_dims,
    const std::array< sunindextype, 3 > disc_dims,
    const std::array< int, 3 > patches,
    const bool periodic,
    int * interactions,
    const sunrealtype endTime,
    const int numberofSteps,
    const std::string outputDirectory,
    const int outputStep,
    const char outputStyle,
    const std::vector< planewave > & planes,
    const std::vector< gaussian3D > & gaussians )
```

complete 3D Simulation function

Conduct the complete 3D simulation process

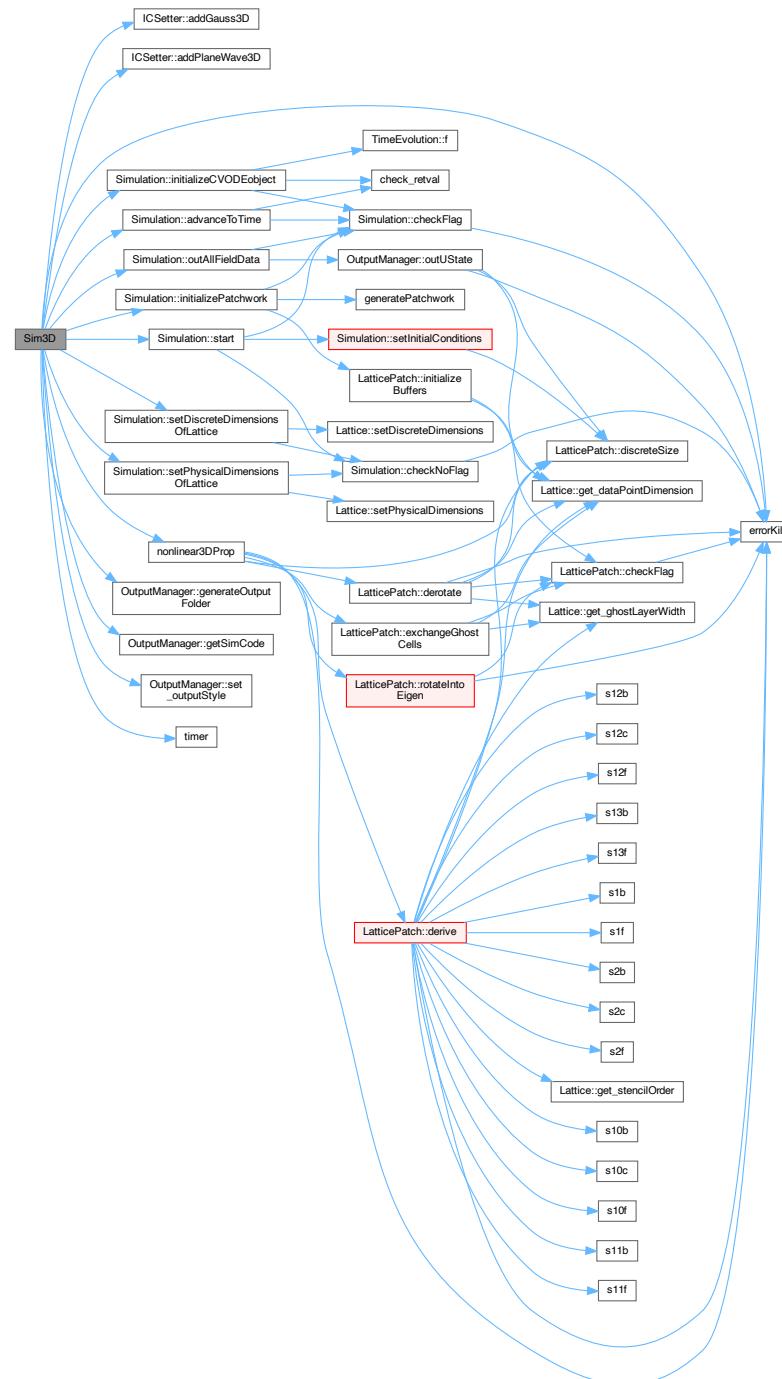
Definition at line 183 of file [SimulationFunctions.cpp](#).

```
00191
00192
00193 // MPI data
00194 double ts = MPI_Wtime();
00195 int myPrc = 0, nPrc = 0; // Get process rank and numer of process
00196 MPI_Comm_rank(MPI_COMM_WORLD,
00197     &myPrc); // rank of the process inside the world communicator
00198 MPI_Comm_size(MPI_COMM_WORLD,
00199     &nPrc); // Size of the communicator is the number of processes
00200
00201 // Check feasibility of the patchwork decomposition
00202 if (myPrc == 0) {
00203     if (nPrc != patches[0] * patches[1] * patches[2]) {
00204         errorKill(
00205             "The number of MPI processes must match the number of patches.");
00206     }
00207     /*
00208     if ( ( disc_dims[0] / patches[0] != disc_dims[1] / patches[1] ) ||
00209         ( disc_dims[0] / patches[0] != disc_dims[2] / patches[2] ) ) {
00210         std::clog
00211             << "\nWarning: Patches should be cubic in terms of the lattice "
00212             "points for the computational efficiency of larger simulations.\n";
00213     }
00214 */
00215 }
00216
00217 // Initialize the simulation, set up the cartesian communicator
00218 Simulation sim(patches[0], patches[1], patches[2],
00219                 StencilOrder, periodic); // simulation object with slicing
00220
00221 // Create the SUNContext object associated with the thread of execution
00222 sim.setPhysicalDimensionsOfLattice(phys_dims[0], phys_dims[1],
00223                                     phys_dims[2]); // spacing of the box
00224 sim.setDiscreteDimensionsOfLattice(
00225     disc_dims[0], disc_dims[1],
00226     disc_dims[2]); // Spacing equivalence to points
00227 sim.initializePatchwork(patches[0], patches[1], patches[2]);
00228
00229 // Add em-waves
00230 for (const auto &plane : planes)
00231     sim.icsettings.addPlaneWave3D(plane.k, plane.p, plane.phi);
00232 for (const auto &gauss : gaussians)
00233     sim.icsettings.addGauss3D(gauss.x0, gauss.axis, gauss.amp, gauss.phip,
00234                               gauss.w0, gauss.zr, gauss.ph0, gauss.phA);
00235
00236 // Check that the patchwork is ready and set the initial conditions
00237 sim.start();
00238
```

```
00239 // Initialize CVode with abs and rel tolerances
00240 sim.initializeCVODEobject(CVodeTol[0], CVodeTol[1]);
00241
00242 // Configure the time evolution function
00243 TimeEvolution::c = interactions;
00244 TimeEvolution::TimeEvolver = nonlinear3DProp;
00245
00246 // Configure the output
00247 sim.outputManager.generateOutputFolder(outputDirectory);
00248 if (!myPrc) {
00249     std::cout << "Simulation code: " << sim.outputManager.getSimCode()
00250         << std::endl;
00251 }
00252 sim.outputManager.set_outputStyle(outputStyle);
00253
00254 //MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00255 //sim.outAllFieldData(0); // output of initial state
00256 // Conduct the propagation in space and time
00257 for (int step = 1; step <= number_of_Steps; step++) {
00258     sim.advanceToTime(endTime / number_of_Steps * step);
00259     if (step % outputStep == 0) {
00260         MPI_Barrier(MPI_COMM_WORLD); // insure correct output
00261         sim.outAllFieldData(step);
00262     }
00263     double tn = MPI_Wtime();
00264     if (!myPrc) {
00265         std::cout << "\rStep " << step << "\t\t" << std::flush;
00266         timer(ts, tn);
00267     }
00268 }
00269 return;
00270 }
```

References [ICSetter::addGauss3D\(\)](#), [ICSetter::addPlaneWave3D\(\)](#), [Simulation::advanceToTime\(\)](#), [TimeEvolution::c](#), [errorKill\(\)](#), [OutputManager::generateOutputFolder\(\)](#), [OutputManager::getSimCode\(\)](#), [Simulation::icsettings](#), [Simulation::initializeCVODEobject\(\)](#), [Simulation::initializePatchwork\(\)](#), [nonlinear3DProp\(\)](#), [Simulation::outAllFieldData\(\)](#), [Simulation::outputManager](#), [OutputManager::set_outputStyle\(\)](#), [Simulation::setDiscreteDimensionsOfLattice\(\)](#), [Simulation::setPhysicalDimensionsOfLattice\(\)](#), [Simulation::start\(\)](#), [TimeEvolution::TimeEvolver](#), and [timer\(\)](#).

Here is the call graph for this function:



6.56.2.4 timer()

```

void timer (
    double & t1,
    double & t2 ) [inline]

```

MPI timer function.

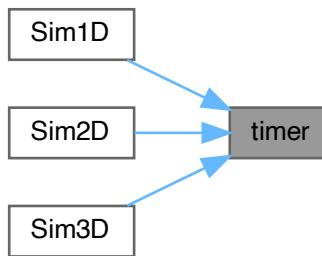
Calculate and print the total simulation time

Definition at line 10 of file [SimulationFunctions.cpp](#).

```
00010     printf("Elapsed time: %fs\n", (t2 - t1));
```

Referenced by [Sim1D\(\)](#), [Sim2D\(\)](#), and [Sim3D\(\)](#).

Here is the caller graph for this function:



6.57 SimulationFunctions.h

[Go to the documentation of this file.](#)

```
00001 //////////////////////////////////////////////////////////////////
00002 /// @file SimulationFunctions.h
00003 /// @brief Full simulation functions for 1D, 2D, and 3D used in main.cpp
00004 //////////////////////////////////////////////////////////////////
00005
00006 #pragma once
00007
00008 // math
00009 #include <cmath>
00010
00011 // project subfile headers
00012 #include "LatticePatch.h"
00013 #include "SimulationClass.h"
00014 #include "TimeEvolutionFunctions.h"
00015
00016 /***** EM-wave structures *****/
00017
00018 /// plane wave structure
00019 struct planewave {
00020     std::array<sunrealtype, 3> k;    /**< wavevector (normalized to \f$ 1/\lambda \f$) */
00021     std::array<sunrealtype, 3> p;    /**< amplitude & polarization vector */
00022     std::array<sunrealtype, 3> phi;  /**< phase shift */
00023 };
00024
00025 /// 1D Gaussian wave structure
00026 struct gaussian1D {
00027     std::array<sunrealtype, 3> k;    /**< wavevector (normalized to \f$ 1/\lambda \f$) */
00028     std::array<sunrealtype, 3> p;    /**< amplitude & polarization vector */
00029     std::array<sunrealtype, 3> x0;   /**< shift from origin */
00030     unrealtype phig;           /**< width */
00031     std::array<sunrealtype, 3> phi;  /**< phase shift */
00032 };
00033
00034 /// 2D Gaussian wave structure
00035 struct gaussian2D {
00036     std::array<sunrealtype, 3> x0;   /**< center */
00037     std::array<sunrealtype, 3> axis;  /**< direction from where it comes */
  
```

```

00038     sunrealtypesamp;           /**< amplitude */
00039     sunrealtypesphiP;         /**< polarization rotation */
00040     sunrealtypesw0;           /**< taille */
00041     sunrealtypeszr;           /**< Rayleigh length */
00042     sunrealtypesph0;          /**< beam center */
00043     sunrealtypesphA;          /**< beam length */
00044 };
00045
00046 /// 3D Gaussian wave structure
00047 struct gaussian3D {
00048     std::array<sunrealtypes, 3> x0;    /**< center */
00049     std::array<sunrealtypes, 3> axis;   /**< direction from where it comes */
00050     sunrealtypesamp;           /**< amplitude */
00051     sunrealtypesphiP;          /**< polarization rotation */
00052     sunrealtypesw0;           /**< taille */
00053     sunrealtypeszr;           /**< Rayleigh length */
00054     sunrealtypesph0;          /**< beam center */
00055     sunrealtypesphA;          /**< beam length */
00056 };
00057
00058 /***** simulation function declarations *****/
00059
00060 /// complete 1D Simulation function
00061 void Sim1D(const std::array<sunrealtypes, 2>, const int, const sunrealtypes,
00062             const sunindextypes, const bool, int *, const sunrealtypes, const int,
00063             const std::string, const int, const char,
00064             const std::vector<planewave> &,
00065             const std::vector<gaussian1D> &);
00066 /// complete 2D Simulation function
00067 void Sim2D(const std::array<sunrealtypes, 2>, const int,
00068             const std::array<sunrealtypes, 2>,
00069             const std::array<sunindextypes, 2>, const std::array<int, 2>,
00070             const bool, int *,
00071             const sunrealtypes, const int, const std::string,
00072             const int, const char,
00073             const std::vector<planewave> &, const std::vector<gaussian2D> &);
00074 /// complete 3D Simulation function
00075 void Sim3D(const std::array<sunrealtypes, 2>, const int,
00076             const std::array<sunrealtypes, 3>,
00077             const std::array<sunindextypes, 3>, const std::array<int, 3>,
00078             const bool, int *,
00079             const sunrealtypes, const int, const std::string,
00080             const int, const char,
00081             const std::vector<planewave> &, const std::vector<gaussian3D> &);
00082
00083 /// MPI timer function
00084 void timer(double &, double &);

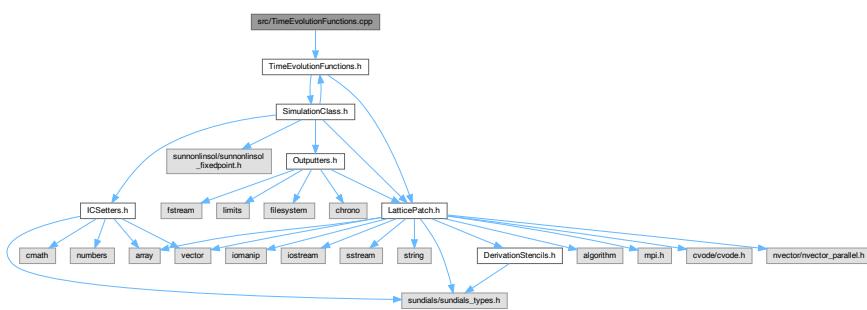

```

6.58 src/TimeEvolutionFunctions.cpp File Reference

Implementation of functions to propagate data vectors in time according to Maxwell's equations, and various orders in the HE weak-field expansion.

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

Include dependency graph for TimeEvolutionFunctions.cpp:



Functions

- void `linear1DProp (LatticePatch *data, N_Vector u, N_Vector udot, int *c)`
only under-the-hood-callable Maxwell propagation in 1D;
- void `nonlinear1DProp (LatticePatch *data, N_Vector u, N_Vector udot, int *c)`
nonlinear 1D HE propagation
- void `linear2DProp (LatticePatch *data, N_Vector u, N_Vector udot, int *c)`
only under-the-hood-callable Maxwell propagation in 2D
- void `nonlinear2DProp (LatticePatch *data, N_Vector u, N_Vector udot, int *c)`
nonlinear 2D HE propagation
- void `linear3DProp (LatticePatch *data, N_Vector u, N_Vector udot, int *c)`
only under-the-hood-callable Maxwell propagation in 3D
- void `nonlinear3DProp (LatticePatch *data, N_Vector u, N_Vector udot, int *c)`
nonlinear 3D HE propagation

6.58.1 Detailed Description

Implementation of functions to propagate data vectors in time according to Maxwell's equations, and various orders in the HE weak-field expansion.

Definition in file [TimeEvolutionFunctions.cpp](#).

6.58.2 Function Documentation

6.58.2.1 linear1DProp()

```
void linear1DProp (
    LatticePatch * data,
    N_Vector u,
    N_Vector udot,
    int * c )
```

only under-the-hood-callable Maxwell propagation in 1D;

Maxwell propagation function for 1D – only for reference.

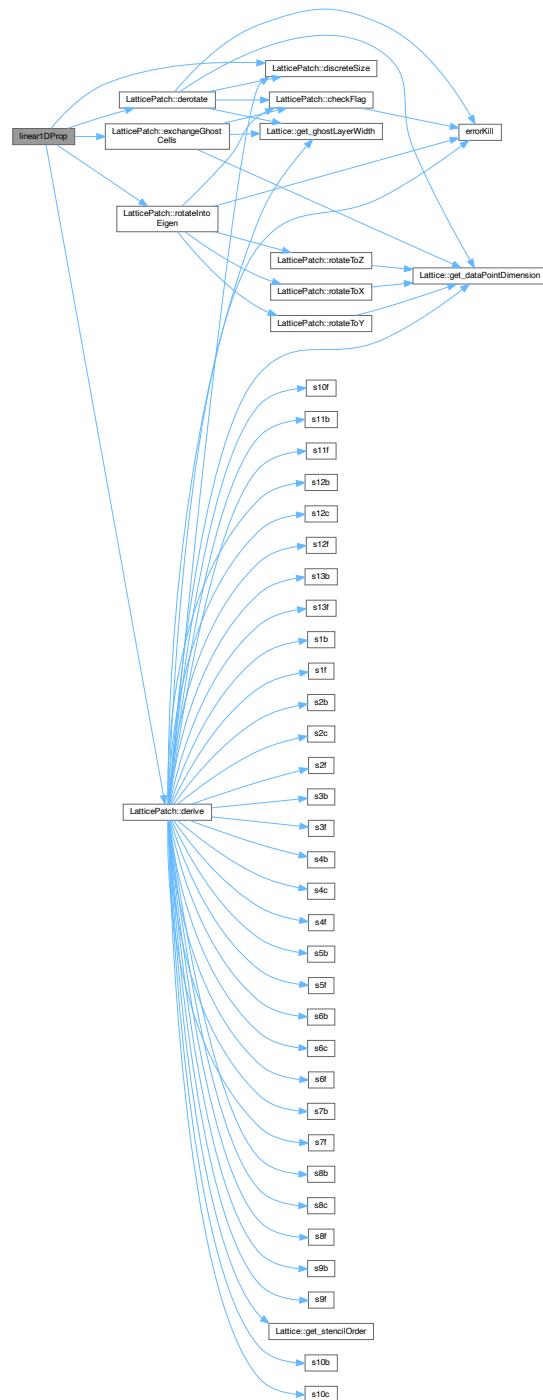
Definition at line 48 of file [TimeEvolutionFunctions.cpp](#).

```
00048
00049
00050 // pointers to temporal and spatial derivative data
00051 sunrealtype *duData = data->duData;
00052 sunrealtype *dxData = data->buffData[1 - 1];
00053
00054 // sequence along any dimension according to the scheme:
00055 data->exchangeGhostCells(1); // -> exchange halos
00056 data->rotateIntoEigen(
00057     1); // -> rotate all data to prepare derivative operation
00058 data->derive(1); // -> perform derivative approximation operation on it
00059 data->derotate(
00060     1, dxData); // -> derotate derived data for ensuing time-evolution
00061
00062 const sunindextype totalNP = data->discreteSize();
00063 sunindextype pp = 0;
00064 for (sunindextype i = 0; i < totalNP; i++) {
00065     pp = i * 6;
```

```
00066  /*
00067   simple vacuum Maxwell equations for the temporal derivatives using the
00068   spatial derivative only in x-direction without polarization or
00069   magnetization terms
00070 */
00071 duData[pp + 0] = 0;
00072 duData[pp + 1] = -dxData[pp + 5];
00073 duData[pp + 2] = dxData[pp + 4];
00074 duData[pp + 3] = 0;
00075 duData[pp + 4] = dxData[pp + 2];
00076 duData[pp + 5] = -dxData[pp + 1];
00077 }
00078 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [LatticePatch::duData](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Here is the call graph for this function:



6.58.2.2 linear2DProp()

```
void linear2DProp (
    LatticePatch * data,
```

```
N_Vector u,
N_Vector udot,
int * c )
```

only under-the-hood-callable Maxwell propagation in 2D

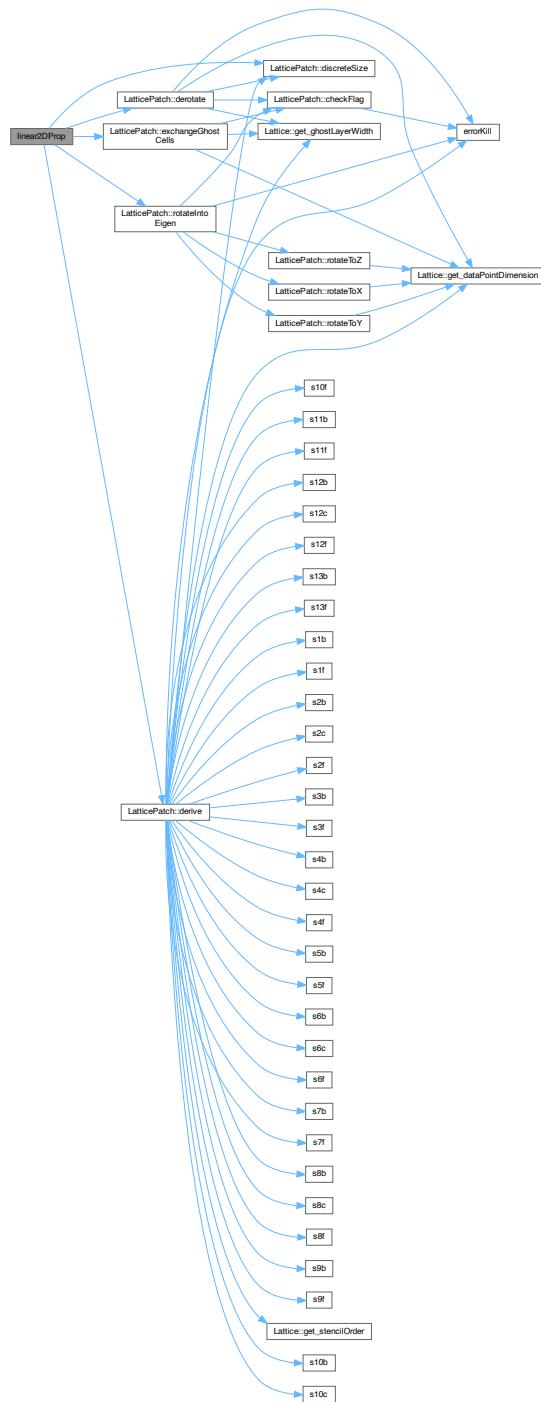
Maxwell propagation function for 2D – only for reference.

Definition at line 296 of file [TimeEvolutionFunctions.cpp](#).

```
00296
00297
00298     sunrealtype *duData = data->duData;
00299     sunrealtype *dxData = data->buffData[1 - 1];
00300     sunrealtype *dyData = data->buffData[2 - 1];
00301
00302     data->exchangeGhostCells(1);
00303     data->rotateIntoEigen(1);
00304     data->derive(1);
00305     data->derotate(1, dxData);
00306     data->exchangeGhostCells(2);
00307     data->rotateIntoEigen(2);
00308     data->derive(2);
00309     data->derotate(2, dyData);
00310
00311     const sunindextype totalNP = data->discreteSize();
00312     sunindextype pp = 0;
00313     for (sunindextype i = 0; i < totalNP; i++) {
00314         pp = i * 6;
00315         duData[pp + 0] = dyData[pp + 5];
00316         duData[pp + 1] = -dxData[pp + 5];
00317         duData[pp + 2] = -dyData[pp + 3] + dxData[pp + 4];
00318         duData[pp + 3] = -dyData[pp + 2];
00319         duData[pp + 4] = dxData[pp + 2];
00320         duData[pp + 5] = dyData[pp + 0] - dxData[pp + 1];
00321     }
00322 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [LatticePatch::duData](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Here is the call graph for this function:



6.58.2.3 linear3DProp()

```
void linear3DProp (
    LatticePatch * data,
```

```
N_Vector u,
N_Vector udot,
int * c )
```

only under-the-hood-callable Maxwell propagation in 3D

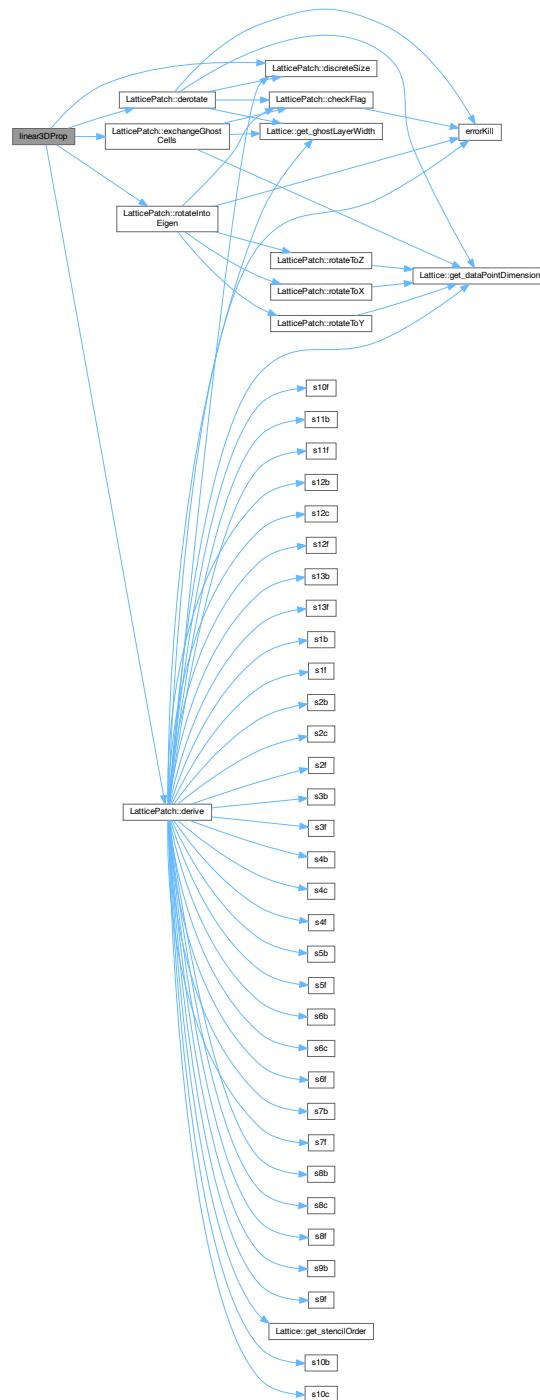
Maxwell propagation function for 3D – only for reference.

Definition at line 509 of file [TimeEvolutionFunctions.cpp](#).

```
00509
00510
00511     sunrealtype *duData = data->duData;
00512     sunrealtype *dxData = data->buffData[1 - 1];
00513     sunrealtype *dyData = data->buffData[2 - 1];
00514     sunrealtype *dzData = data->buffData[3 - 1];
00515
00516     data->exchangeGhostCells(1);
00517     data->rotateIntoEigen(1);
00518     data->derive(1);
00519     data->derotate(1, dxData);
00520     data->exchangeGhostCells(2);
00521     data->rotateIntoEigen(2);
00522     data->derive(2);
00523     data->derotate(2, dyData);
00524     data->exchangeGhostCells(3);
00525     data->rotateIntoEigen(3);
00526     data->derive(3);
00527     data->derotate(3, dzData);
00528
00529     const sunindextype totalNP = data->discreteSize();
00530     sunindextype pp = 0;
00531     for (sunindextype i = 0; i < totalNP; i++) {
00532         pp = i * 6;
00533         duData[pp + 0] = dyData[pp + 5] - dzData[pp + 4];
00534         duData[pp + 1] = dzData[pp + 3] - dxData[pp + 5];
00535         duData[pp + 2] = dxData[pp + 4] - dyData[pp + 3];
00536         duData[pp + 3] = -dyData[pp + 2] + dzData[pp + 1];
00537         duData[pp + 4] = -dzData[pp + 0] + dxData[pp + 2];
00538         duData[pp + 5] = -dxData[pp + 1] + dyData[pp + 0];
00539     }
00540 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [LatticePatch::duData](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Here is the call graph for this function:



6.58.2.4 nonlinear1DProp()

```
void nonlinear1DProp (
    LatticePatch * data,
```

```
N_Vector u,
N_Vector udot,
int * c )
```

nonlinear 1D HE propagation

HE propagation function for 1D.

Definition at line 81 of file [TimeEvolutionFunctions.cpp](#).

```
00081
00082
00083 // NVector pointers to provided field values and their temp. derivatives
00084 #if defined(_OPENMP)
00085 sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00086             *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00087 #else
00088 sunrealtype *udata = NV_DATA_P(u),
00089             *dudata = NV_DATA_P(udot);
00090 #endif
00091
00092 // pointer to spatial derivatives via pach data
00093 sunrealtype *dxData = data->buffData[1 - 1];
00094
00095 // same sequence as in the linear case
00096 data->exchangeGhostCells(1);
00097 data->rotateIntoEigen(1);
00098 data->derive(1);
00099 data->derotate(1, dxData);
00100
00101 /*
00102 F and G are nonzero in the nonlinear case,
00103 polarization and magnetization derivatives
00104 w.r.t. E- and B-field go into the e.o.m.
00105 */
00106 static sunrealtype f, g; // em field invariants F, G
00107 // derivatives of HE Lagrangian w.r.t. field invariants
00108 static sunrealtype lf, lff, lfg, lg, lgg;
00109 // matrix to hold derivatives of polarization and magnetization
00110 static std::array<sunrealtype, 21> JMM;
00111 // array to hold E^2 and B^2 components
00112 static std::array<sunrealtype, 6> Quad;
00113 // array to hold intermediate temp. derivatives of E and B
00114 static std::array<sunrealtype, 6> h;
00115 // determinant needed for explicit matrix inversion
00116 static sunrealtype detC = nan("0x12345");
00117
00118 // number of points in the patch
00119 const sunindextype totalNP = data->discreteSize();
00120 #pragma omp parallel for default(none) \
00121 private(JMM, Quad, h, detC) \
00122 shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, dxData) \
00123 schedule(static)
00124 for (sunindextype pp = 0; pp < totalNP * 6;
00125     pp += 6) { // loop over all 6dim points in the patch
00126     // em field Lorentz invariants F and G
00127     f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00128                 (Quad[1] = udata[pp + 1] * udata[pp + 1]) +
00129                 (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00130                 (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00131                 (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00132                 (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00133     g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
00134         udata[pp + 2] * udata[pp + 5];
00135     // process/expansion order and corresponding derivative values of L
00136     // w.r.t. F, G
00137     switch (*c) {
00138     case 0: // linear Maxwell vacuum
00139         lf = 0;
00140         lff = 0;
00141         lfg = 0;
00142         lg = 0;
00143         lgg = 0;
00144         break;
00145     case 1: // only 4-photon processes
00146         lf = 0.000206527095658582755255648 * f;
00147         lff = 0.000206527095658582755255648;
00148         lfg = 0;
00149         lg = 0.0003614224174025198216973841 * g;
00150         lgg = 0.0003614224174025198216973841;
00151         break;
00152     case 2: // only 6-photon processes
00153         lf = 0.000354046449700427580438254 * f * f +
00154             0.000191775160254398272737387 * g * g;
```

```

00155     lff = 0.0007080928994008551608765075 * f;
00156     lfg = 0.0003835503205087965454747749 * g;
00157     lg = 0.0003835503205087965454747749 * f * g;
00158     lgg = 0.0003835503205087965454747749 * f;
00159     break;
00160 case 3: // 4- and 6-photon processes
00161     lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00162         f +
00163         0.000191775160254398272737387 * g * g;
00164     lff = 0.000206527095658582755255648 + 0.0007080928994008551608765075 * f;
00165     lfg = 0.0003835503205087965454747749 * g;
00166     lg = (0.0003614224174025198216973841 +
00167         0.0003835503205087965454747749 * f) *
00168         g;
00169     lgg = 0.0003614224174025198216973841 + 0.0003835503205087965454747749 * f;
00170     break;
00171 default:
00172     errorKill(
00173         "You need to specify a correct order in the weak-field expansion.");
00174 }
00175
00176 // derivatives of polarization and magnetization w.r.t. E and B
00177 // Jpx(Ex)
00178 JMM[0] = lf + lff * Quad[0] +
00179     udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00180 // Jpx(Ey)
00181 JMM[1] =
00182     lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00183     lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00184 // Jpy(Ey)
00185 JMM[2] = lf + lff * Quad[1] +
00186     udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00187 // Jpx(Ez) = Jpz(Ex)
00188 JMM[3] =
00189     lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00190     lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00191 // Jpy(Ez) = Jpz(Ey)
00192 JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00193     lfg * udata[2 + pp] * udata[4 + pp] +
00194     lfg * udata[1 + pp] * udata[5 + pp] +
00195     lgg * udata[4 + pp] * udata[5 + pp];
00196 // Jpz(Ez)
00197 JMM[5] = lf + lff * Quad[2] +
00198     udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00199 // Jpx(Bx) = Jmx(Ex)
00200 JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00201     (-lff + lgg) * udata[pp] * udata[3 + pp];
00202 // Jpy(Bx) = Jmx(Ey)
00203 JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00204     udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00205 // Jpz(Bx) = Jmx(Ez)
00206 JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00207     udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00208 // Jmx(Bx)
00209 JMM[9] = -lf + lgg * Quad[0] +
00210     udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00211 // Jpx(By) = Jmy(Ex)
00212 JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00213     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00214 // Jpy(By) = Jmy(Ey)
00215 JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00216     (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00217 // Jpz(By) = Jmy(Ez)
00218 JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00219     udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00220 // Jmx(By) = Jmy(Bx)
00221 JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00222     lff * udata[3 + pp] * udata[4 + pp] -
00223     lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00224 // Jmy(By)
00225 JMM[14] = -lf + lgg * Quad[1] +
00226     udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00227 // Jmz(Ex) = Jpx(Bz)
00228 JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00229     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00230 // Jmz(Ey) = Jpy(Bz)
00231 JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00232     (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00233 // Jpz(Bz) = Jmz(Ez)
00234 JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00235     (-lff + lgg) * udata[2 + pp] * udata[5 + pp];
00236 // Jmz(Bx) = Jmx(Bz)
00237 JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00238     lff * udata[3 + pp] * udata[5 + pp] -
00239     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00240 // Jmy(Bz) = Jmz(By)
00241 JMM[19] =

```

```

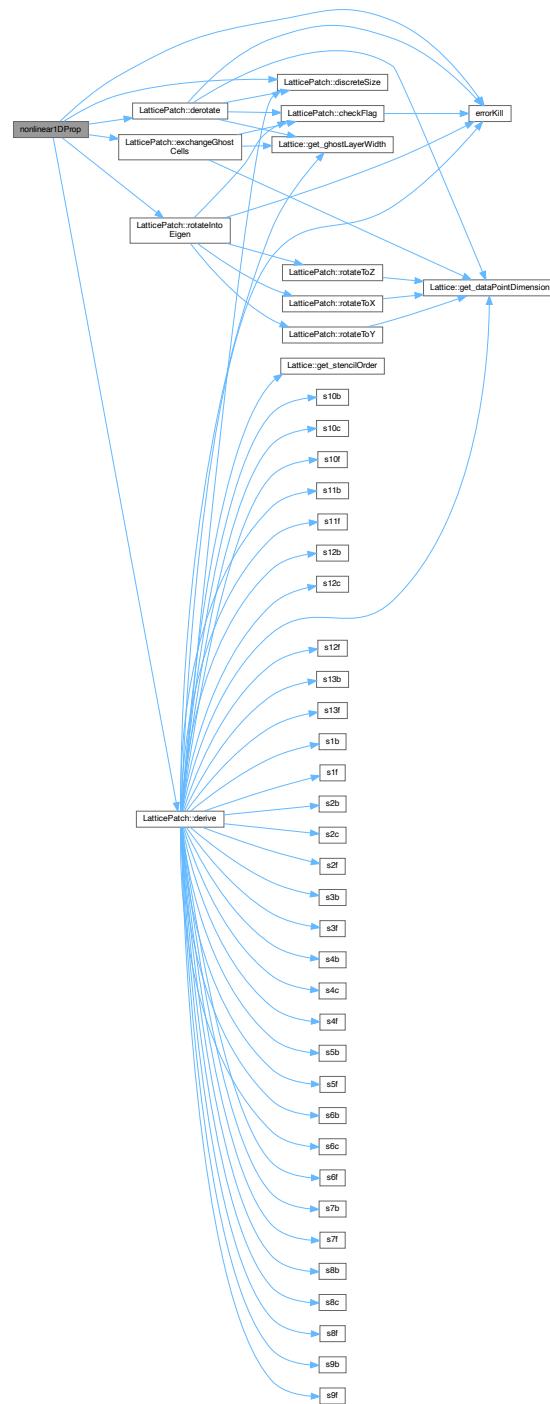
00242     lgg * udata[1 + pp] * udata[2 + pp] +
00243     lff * udata[4 + pp] * udata[5 + pp] -
00244     lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]));
00245 // Jmz(Bz)
00246 JMM[20] = -lf + lgg * Quad[2] +
00247     udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00248
00249 // apply Z
00250 // top block: -QJm(E)*E, Q-QJm(B)*B
00251 h[0] = 0;
00252 h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00253     dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00254     dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00255 h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00256     dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00257     dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00258 // bottom blocks: -Q*E
00259 h[3] = 0;
00260 h[4] = dxData[2 + pp];
00261 h[5] = -dxData[1 + pp];
00262 // (1+A)^-1 applies only to E components
00263 // -Jp(B)*B
00264 h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00265 h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00266 h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00267 // apply C^-1 explicitly, with C=1+Jp(E)
00268 dudata[pp + 0] =
00269     h[2] * (-(JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00270     h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00271     h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00272 dudata[pp + 1] =
00273     h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00274     h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00275     h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00276 dudata[pp + 2] =
00277     h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00278     h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00279     h[0] * (-(1 + JMM[2]) * JMM[3]) + JMM[1] * JMM[4];
00280 detC = // determinant of C
00281     -((1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00282     (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00283     JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00284     JMM[1] * (-(JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00285 dudata[pp + 0] /= detC;
00286 dudata[pp + 1] /= detC;
00287 dudata[pp + 2] /= detC;
00288 dudata[pp + 3] = h[3];
00289 dudata[pp + 4] = h[4];
00290 dudata[pp + 5] = h[5];
00291 }
00292 return;
00293 }

```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [errorKill\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Referenced by [Sim1D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.58.2.5 nonlinear2DProp()

```

void nonlinear2DProp (
    LatticePatch * data,
    N_Vector u,
    N_Vector udot,
    int * c )

```

nonlinear 2D HE propagation

HE propagation function for 2D.

Definition at line 325 of file [TimeEvolutionFunctions.cpp](#).

```

00325
00326
00327 #if defined(_OPENMP)
00328     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00329                 *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00330 #else
00331     sunrealtype *udata = NV_DATA_P(u),
00332                 *dudata = NV_DATA_P(udot);
00333 #endif
00334
00335     sunrealtype *dxData = data->buffData[1 - 1];
00336     sunrealtype *dyData = data->buffData[2 - 1];
00337
00338     data->exchangeGhostCells(1);
00339     data->rotateIntoEigen(1);
00340     data->derive(1);
00341     data->derotate(1, dxData);
00342     data->exchangeGhostCells(2);
00343     data->rotateIntoEigen(2);
00344     data->derive(2);
00345     data->derotate(2, dyData);
00346
00347     static sunrealtype f, g;
00348     static sunrealtype lf, lff, lfg, lg, lgg;
00349     static std::array<sunrealtype, 21> JMM;
00350     static std::array<sunrealtype, 6> Quad;
00351     static std::array<sunrealtype, 6> h;
00352     static sunrealtype detC;
00353
00354     const sunindextype totalNP = data->discreteSize();
00355     #pragma omp parallel for default(none) \
00356     private(JMM, Quad, h, detC) \
00357     shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, \
00358             dxData, dyData) \
00359     schedule(static)
00360     for (sunindextype pp = 0; pp < totalNP * 6; pp += 6) {
00361         f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00362                     (Quad[1] = udata[pp + 1] * udata[pp + 1])) +
00363                     (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00364                     (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00365                     (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00366                     (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00367         g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +

```

```

00368     udata[pp + 2] * udata[pp + 5];
00369     switch (*c) {
00370     case 0:
00371         lf = 0;
00372         lff = 0;
00373         lfg = 0;
00374         lg = 0;
00375         lgg = 0;
00376         break;
00377     case 1:
00378         lf = 0.000206527095658582755255648 * f;
00379         lff = 0.000206527095658582755255648;
00380         lfg = 0;
00381         lg = 0.0003614224174025198216973841 * g;
00382         lgg = 0.0003614224174025198216973841;
00383         break;
00384     case 2:
00385         lf = 0.000354046449700427580438254 * f * f +
00386             0.000191775160254398272737387 * g * g;
00387         lff = 0.0007080928994008551608765075 * f;
00388         lfg = 0.0003835503205087965454747749 * g;
00389         lg = 0.0003835503205087965454747749 * f * g;
00390         lgg = 0.0003835503205087965454747749 * f;
00391         break;
00392     case 3:
00393         lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00394             f +
00395             0.000191775160254398272737387 * g * g;
00396         lff = 0.000206527095658582755255648 + 0.000708092899400855160876508 * f;
00397         lfg = 0.0003835503205087965454747749 * g;
00398         lg = (0.000361422417402519821697384 + 0.000383550320508796545474775 * f) *
00399             g;
00400         lgg = 0.000361422417402519821697384 + 0.000383550320508796545474775 * f;
00401         break;
00402     default:
00403         errorKill(
00404             "You need to specify a correct order in the weak-field expansion.");
00405     }
00406
00407 JMM[0] = lf + lff * Quad[0] +
00408     udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00409 JMM[1] =
00410     lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00411     lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00412 JMM[2] = lf + lff * Quad[1] +
00413     udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00414 JMM[3] =
00415     lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00416     lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00417 JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00418     lfg * udata[2 + pp] * udata[4 + pp] +
00419     lfg * udata[1 + pp] * udata[5 + pp] +
00420     lgg * udata[4 + pp] * udata[5 + pp];
00421 JMM[5] = lf + lff * Quad[2] +
00422     udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00423 JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00424     (-lff + lgg) * udata[pp] * udata[3 + pp];
00425 JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00426     udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00427 JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00428     udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00429 JMM[9] = -lf + lgg * Quad[0] +
00430     udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00431 JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00432     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00433 JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00434     (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00435 JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00436     udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00437 JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00438     lff * udata[3 + pp] * udata[4 + pp] -
00439     lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00440 JMM[14] = -lf + lgg * Quad[1] +
00441     udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00442 JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00443     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00444 JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00445     (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00446 JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00447     (-lff + lgg) * udata[2 + pp] * udata[5 + pp];
00448 JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00449     lff * udata[3 + pp] * udata[5 + pp] -
00450     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00451 JMM[19] =
00452     lgg * udata[1 + pp] * udata[2 + pp] +
00453     lff * udata[4 + pp] * udata[5 + pp] -
00454     lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]);

```

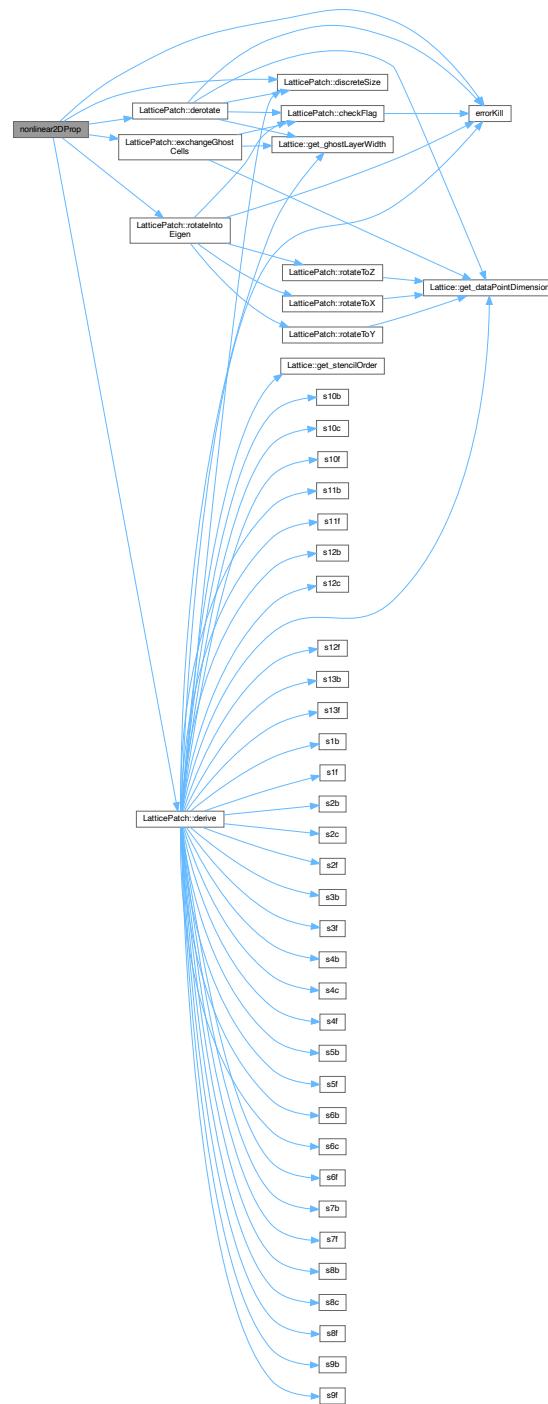
```

00455     JMM[20] = -lf + lgg * Quad[2] +
00456         udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00457
00458     h[0] = 0;
00459     h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00460         dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00461         dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00462     h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00463         dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00464         dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00465     h[3] = 0;
00466     h[4] = dxData[2 + pp];
00467     h[5] = -dxData[1 + pp];
00468     h[0] += -(dyData[pp] * JMM[15]) - dyData[1 + pp] * JMM[16] -
00469         dyData[2 + pp] * JMM[17] - dyData[3 + pp] * JMM[18] -
00470         dyData[4 + pp] * JMM[19] + dyData[5 + pp] * (1 - JMM[20]);
00471     h[1] += 0;
00472     h[2] += dyData[pp] * JMM[6] + dyData[1 + pp] * JMM[7] +
00473         dyData[2 + pp] * JMM[8] + dyData[3 + pp] * (-1 + JMM[9]) +
00474         dyData[4 + pp] * JMM[13] + dyData[5 + pp] * JMM[18];
00475     h[3] += -dyData[2 + pp];
00476     h[4] += 0;
00477     h[5] += dyData[pp];
00478     h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00479     h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00480     h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00481     dudata[pp + 0] =
00482         h[2] * (- (JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00483         h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00484         h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00485     dudata[pp + 1] =
00486         h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00487         h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00488         h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00489     dudata[pp + 2] =
00490         h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00491         h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00492         h[0] * (-((1 + JMM[2]) * JMM[3]) + JMM[1] * JMM[4]);
00493     detC =
00494         -((1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00495         (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00496         JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00497         JMM[1] * (- (JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00498     dudata[pp + 0] /= detC;
00499     dudata[pp + 1] /= detC;
00500     dudata[pp + 2] /= detC;
00501     dudata[pp + 3] = h[3];
00502     dudata[pp + 4] = h[4];
00503     dudata[pp + 5] = h[5];
00504 }
00505 return;
00506 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [errorKill\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Referenced by [Sim2D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.58.2.6 nonlinear3DProp()

```
void nonlinear3DProp (
    LatticePatch * data,
    N_Vector u,
    N_Vector udot,
    int * c )
```

nonlinear 3D HE propagation

HE propagation function for 3D.

Definition at line 543 of file [TimeEvolutionFunctions.cpp](#).

```
00543
00544
00545 #if defined(_OPENMP)
00546     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00547             *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00548 #else
00549     sunrealtype *udata = NV_DATA_P(u),
00550         *dudata = NV_DATA_P(udot);
00551 #endif
00552
00553     sunrealtype *dxData = data->buffData[1 - 1];
00554     sunrealtype *dyData = data->buffData[2 - 1];
00555     sunrealtype *dzData = data->buffData[3 - 1];
00556
00557     data->exchangeGhostCells(1);
00558     data->rotateIntoEigen(1);
00559     data->derive(1);
00560     data->derotate(1,dxData);
00561     data->exchangeGhostCells(2);
00562     data->rotateIntoEigen(2);
00563     data->derive(2);
00564     data->derotate(2,dyData);
00565     data->exchangeGhostCells(3);
00566     data->rotateIntoEigen(3);
00567     data->derive(3);
00568     data->derotate(3,dzData);
00569
00570     static sunrealtype f, g;
00571     static sunrealtype lf, lff, lfg, lg, lgg;
00572     static std::array<sunrealtype, 21> JMM;
00573     static std::array<sunrealtype, 6> Quad;
00574     static std::array<sunrealtype, 6> h;
00575     static sunrealtype detC = nan("0x12345");
00576
00577     const sunindextype totalNP = data->discreteSize();
00578     #pragma omp parallel for default(none) \
00579     private(JMM, Quad, h, detC) \
00580     shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, \
00581           dxData, dyData, dzData) \
00582     schedule(static)
00583     for (sunindextype pp = 0; pp < totalNP * 6; pp += 6) {
00584         f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00585                     (Quad[1] = udata[pp + 1] * udata[pp + 1]) +
00586                     (Quad[2] = udata[pp + 2] * udata[pp + 2]) +
00587                     (Quad[3] = udata[pp + 3] * udata[pp + 3]) +
00588                     (Quad[4] = udata[pp + 4] * udata[pp + 4]) +
00589                     (Quad[5] = udata[pp + 5] * udata[pp + 5])) / 6.0;
00590
00591         lf = (lff = (lg = (lgg = (f * f) * f) * f) * f) * f;
00592
00593         lfg = lf * lff;
00594         lg = lf * lg;
00595         lgg = lf * lgg;
00596
00597         f = (f * lf) * lf;
00598         g = (g * lf) * lf;
00599
00600         lf = lf * lf;
00601         lff = lf * lf;
00602         lfg = lf * lfg;
00603         lg = lf * lg;
00604         lgg = lf * lgg;
00605
00606         f = (f * lf) * lf;
00607         g = (g * lf) * lf;
00608
00609         lf = lf * lf;
00610         lff = lf * lf;
00611         lfg = lf * lfg;
00612         lg = lf * lg;
00613         lgg = lf * lgg;
00614
00615         f = (f * lf) * lf;
00616         g = (g * lf) * lf;
00617
00618         lf = lf * lf;
00619         lff = lf * lf;
00620         lfg = lf * lfg;
00621         lg = lf * lg;
00622         lgg = lf * lgg;
00623
00624         f = (f * lf) * lf;
00625         g = (g * lf) * lf;
00626
00627         lf = lf * lf;
00628         lff = lf * lf;
00629         lfg = lf * lfg;
00630         lg = lf * lg;
00631         lgg = lf * lgg;
00632
00633         f = (f * lf) * lf;
00634         g = (g * lf) * lf;
00635
00636         lf = lf * lf;
00637         lff = lf * lf;
00638         lfg = lf * lfg;
00639         lg = lf * lg;
00640         lgg = lf * lgg;
00641
00642         f = (f * lf) * lf;
00643         g = (g * lf) * lf;
00644
00645         lf = lf * lf;
00646         lff = lf * lf;
00647         lfg = lf * lfg;
00648         lg = lf * lg;
00649         lgg = lf * lgg;
00650
00651         f = (f * lf) * lf;
00652         g = (g * lf) * lf;
00653
00654         lf = lf * lf;
00655         lff = lf * lf;
00656         lfg = lf * lfg;
00657         lg = lf * lg;
00658         lgg = lf * lgg;
00659
00660         f = (f * lf) * lf;
00661         g = (g * lf) * lf;
00662
00663         lf = lf * lf;
00664         lff = lf * lf;
00665         lfg = lf * lfg;
00666         lg = lf * lg;
00667         lgg = lf * lgg;
00668
00669         f = (f * lf) * lf;
00670         g = (g * lf) * lf;
00671
00672         lf = lf * lf;
00673         lff = lf * lf;
00674         lfg = lf * lfg;
00675         lg = lf * lg;
00676         lgg = lf * lgg;
00677
00678         f = (f * lf) * lf;
00679         g = (g * lf) * lf;
00680
00681         lf = lf * lf;
00682         lff = lf * lf;
00683         lfg = lf * lfg;
00684         lg = lf * lg;
00685         lgg = lf * lgg;
00686
00687         f = (f * lf) * lf;
00688         g = (g * lf) * lf;
00689
00690         lf = lf * lf;
00691         lff = lf * lf;
00692         lfg = lf * lfg;
00693         lg = lf * lg;
00694         lgg = lf * lgg;
00695
00696         f = (f * lf) * lf;
00697         g = (g * lf) * lf;
00698
00699         lf = lf * lf;
00700         lff = lf * lf;
00701         lfg = lf * lfg;
00702         lg = lf * lg;
00703         lgg = lf * lgg;
00704
00705         f = (f * lf) * lf;
00706         g = (g * lf) * lf;
00707
00708         lf = lf * lf;
00709         lff = lf * lf;
00710         lfg = lf * lfg;
00711         lg = lf * lg;
00712         lgg = lf * lgg;
00713
00714         f = (f * lf) * lf;
00715         g = (g * lf) * lf;
00716
00717         lf = lf * lf;
00718         lff = lf * lf;
00719         lfg = lf * lfg;
00720         lg = lf * lg;
00721         lgg = lf * lgg;
00722
00723         f = (f * lf) * lf;
00724         g = (g * lf) * lf;
00725
00726         lf = lf * lf;
00727         lff = lf * lf;
00728         lfg = lf * lfg;
00729         lg = lf * lg;
00730         lgg = lf * lgg;
00731
00732         f = (f * lf) * lf;
00733         g = (g * lf) * lf;
00734
00735         lf = lf * lf;
00736         lff = lf * lf;
00737         lfg = lf * lfg;
00738         lg = lf * lg;
00739         lgg = lf * lgg;
00740
00741         f = (f * lf) * lf;
00742         g = (g * lf) * lf;
00743
00744         lf = lf * lf;
00745         lff = lf * lf;
00746         lfg = lf * lfg;
00747         lg = lf * lg;
00748         lgg = lf * lgg;
00749
00750         f = (f * lf) * lf;
00751         g = (g * lf) * lf;
00752
00753         lf = lf * lf;
00754         lff = lf * lf;
00755         lfg = lf * lfg;
00756         lg = lf * lg;
00757         lgg = lf * lgg;
00758
00759         f = (f * lf) * lf;
00760         g = (g * lf) * lf;
00761
00762         lf = lf * lf;
00763         lff = lf * lf;
00764         lfg = lf * lfg;
00765         lg = lf * lg;
00766         lgg = lf * lgg;
00767
00768         f = (f * lf) * lf;
00769         g = (g * lf) * lf;
00770
00771         lf = lf * lf;
00772         lff = lf * lf;
00773         lfg = lf * lfg;
00774         lg = lf * lg;
00775         lgg = lf * lgg;
00776
00777         f = (f * lf) * lf;
00778         g = (g * lf) * lf;
00779
00780         lf = lf * lf;
00781         lff = lf * lf;
00782         lfg = lf * lfg;
00783         lg = lf * lg;
00784         lgg = lf * lgg;
00785
00786         f = (f * lf) * lf;
00787         g = (g * lf) * lf;
00788
00789         lf = lf * lf;
00790         lff = lf * lf;
00791         lfg = lf * lfg;
00792         lg = lf * lg;
00793         lgg = lf * lgg;
00794
00795         f = (f * lf) * lf;
00796         g = (g * lf) * lf;
00797
00798         lf = lf * lf;
00799         lff = lf * lf;
00800         lfg = lf * lfg;
00801         lg = lf * lg;
00802         lgg = lf * lgg;
00803
00804         f = (f * lf) * lf;
00805         g = (g * lf) * lf;
00806
00807         lf = lf * lf;
00808         lff = lf * lf;
00809         lfg = lf * lfg;
00810         lg = lf * lg;
00811         lgg = lf * lgg;
00812
00813         f = (f * lf) * lf;
00814         g = (g * lf) * lf;
00815
00816         lf = lf * lf;
00817         lff = lf * lf;
00818         lfg = lf * lfg;
00819         lg = lf * lg;
00820         lgg = lf * lgg;
00821
00822         f = (f * lf) * lf;
00823         g = (g * lf) * lf;
00824
00825         lf = lf * lf;
00826         lff = lf * lf;
00827         lfg = lf * lfg;
00828         lg = lf * lg;
00829         lgg = lf * lgg;
00830
00831         f = (f * lf) * lf;
00832         g = (g * lf) * lf;
00833
00834         lf = lf * lf;
00835         lff = lf * lf;
00836         lfg = lf * lfg;
00837         lg = lf * lg;
00838         lgg = lf * lgg;
00839
00840         f = (f * lf) * lf;
00841         g = (g * lf) * lf;
00842
00843         lf = lf * lf;
00844         lff = lf * lf;
00845         lfg = lf * lfg;
00846         lg = lf * lg;
00847         lgg = lf * lgg;
00848
00849         f = (f * lf) * lf;
00850         g = (g * lf) * lf;
00851
00852         lf = lf * lf;
00853         lff = lf * lf;
00854         lfg = lf * lfg;
00855         lg = lf * lg;
00856         lgg = lf * lgg;
00857
00858         f = (f * lf) * lf;
00859         g = (g * lf) * lf;
00860
00861         lf = lf * lf;
00862         lff = lf * lf;
00863         lfg = lf * lfg;
00864         lg = lf * lg;
00865         lgg = lf * lgg;
00866
00867         f = (f * lf) * lf;
00868         g = (g * lf) * lf;
00869
00870         lf = lf * lf;
00871         lff = lf * lf;
00872         lfg = lf * lfg;
00873         lg = lf * lg;
00874         lgg = lf * lgg;
00875
00876         f = (f * lf) * lf;
00877         g = (g * lf) * lf;
00878
00879         lf = lf * lf;
00880         lff = lf * lf;
00881         lfg = lf * lfg;
00882         lg = lf * lg;
00883         lgg = lf * lgg;
00884
00885         f = (f * lf) * lf;
00886         g = (g * lf) * lf;
00887
00888         lf = lf * lf;
00889         lff = lf * lf;
00890         lfg = lf * lfg;
00891         lg = lf * lg;
00892         lgg = lf * lgg;
00893
00894         f = (f * lf) * lf;
00895         g = (g * lf) * lf;
00896
00897         lf = lf * lf;
00898         lff = lf * lf;
00899         lfg = lf * lfg;
00900         lg = lf * lg;
00901         lgg = lf * lgg;
00902
00903         f = (f * lf) * lf;
00904         g = (g * lf) * lf;
00905
00906         lf = lf * lf;
00907         lff = lf * lf;
00908         lfg = lf * lfg;
00909         lg = lf * lg;
00910         lgg = lf * lgg;
00911
00912         f = (f * lf) * lf;
00913         g = (g * lf) * lf;
00914
00915         lf = lf * lf;
00916         lff = lf * lf;
00917         lfg = lf * lfg;
00918         lg = lf * lg;
00919         lgg = lf * lgg;
00920
00921         f = (f * lf) * lf;
00922         g = (g * lf) * lf;
00923
00924         lf = lf * lf;
00925         lff = lf * lf;
00926         lfg = lf * lfg;
00927         lg = lf * lg;
00928         lgg = lf * lgg;
00929
00930         f = (f * lf) * lf;
00931         g = (g * lf) * lf;
00932
00933         lf = lf * lf;
00934         lff = lf * lf;
00935         lfg = lf * lfg;
00936         lg = lf * lg;
00937         lgg = lf * lgg;
00938
00939         f = (f * lf) * lf;
00940         g = (g * lf) * lf;
00941
00942         lf = lf * lf;
00943         lff = lf * lf;
00944         lfg = lf * lfg;
00945         lg = lf * lg;
00946         lgg = lf * lgg;
00947
00948         f = (f * lf) * lf;
00949         g = (g * lf) * lf;
00950
00951         lf = lf * lf;
00952         lff = lf * lf;
00953         lfg = lf * lfg;
00954         lg = lf * lg;
00955         lgg = lf * lgg;
00956
00957         f = (f * lf) * lf;
00958         g = (g * lf) * lf;
00959
00960         lf = lf * lf;
00961         lff = lf * lf;
00962         lfg = lf * lfg;
00963         lg = lf * lg;
00964         lgg = lf * lgg;
00965
00966         f = (f * lf) * lf;
00967         g = (g * lf) * lf;
00968
00969         lf = lf * lf;
00970         lff = lf * lf;
00971         lfg = lf * lfg;
00972         lg = lf * lg;
00973         lgg = lf * lgg;
00974
00975         f = (f * lf) * lf;
00976         g = (g * lf) * lf;
00977
00978         lf = lf * lf;
00979         lff = lf * lf;
00980         lfg = lf * lfg;
00981         lg = lf * lg;
00982         lgg = lf * lgg;
00983
00984         f = (f * lf) * lf;
00985         g = (g * lf) * lf;
00986
00987         lf = lf * lf;
00988         lff = lf * lf;
00989         lfg = lf * lfg;
00990         lg = lf * lg;
00991         lgg = lf * lgg;
00992
00993         f = (f * lf) * lf;
00994         g = (g * lf) * lf;
00995
00996         lf = lf * lf;
00997         lff = lf * lf;
00998         lfg = lf * lfg;
00999         lg = lf * lg;
01000         lgg = lf * lgg;
01001
01002         f = (f * lf) * lf;
01003         g = (g * lf) * lf;
01004
01005         lf = lf * lf;
01006         lff = lf * lf;
01007         lfg = lf * lfg;
01008         lg = lf * lg;
01009         lgg = lf * lgg;
01010
01011         f = (f * lf) * lf;
01012         g = (g * lf) * lf;
01013
01014         lf = lf * lf;
01015         lff = lf * lf;
01016         lfg = lf * lfg;
01017         lg = lf * lg;
01018         lgg = lf * lgg;
01019
01020         f = (f * lf) * lf;
01021         g = (g * lf) * lf;
01022
01023         lf = lf * lf;
01024         lff = lf * lf;
01025         lfg = lf * lfg;
01026         lg = lf * lg;
01027         lgg = lf * lgg;
01028
01029         f = (f * lf) * lf;
01030         g = (g * lf) * lf;
01031
01032         lf = lf * lf;
01033         lff = lf * lf;
01034         lfg = lf * lfg;
01035         lg = lf * lg;
01036         lgg = lf * lgg;
01037
01038         f = (f * lf) * lf;
01039         g = (g * lf) * lf;
01040
01041         lf = lf * lf;
01042         lff = lf * lf;
01043         lfg = lf * lfg;
01044         lg = lf * lg;
01045         lgg = lf * lgg;
01046
01047         f = (f * lf) * lf;
01048         g = (g * lf) * lf;
01049
01050         lf = lf * lf;
01051         lff = lf * lf;
01052         lfg = lf * lfg;
01053         lg = lf * lg;
01054         lgg = lf * lgg;
01055
01056         f = (f * lf) * lf;
01057         g = (g * lf) * lf;
01058
01059         lf = lf * lf;
01060         lff = lf * lf;
01061         lfg = lf * lfg;
01062         lg = lf * lg;
01063         lgg = lf * lgg;
01064
01065         f = (f * lf) * lf;
01066         g = (g * lf) * lf;
01067
01068         lf = lf * lf;
01069         lff = lf * lf;
01070         lfg = lf * lfg;
01071         lg = lf * lg;
01072         lgg = lf * lgg;
01073
01074         f = (f * lf) * lf;
01075         g = (g * lf) * lf;
01076
01077         lf = lf * lf;
01078         lff = lf * lf;
01079         lfg = lf * lfg;
01080         lg = lf * lg;
01081         lgg = lf * lgg;
01082
01083         f = (f * lf) * lf;
01084         g = (g * lf) * lf;
01085
01086         lf = lf * lf;
01087         lff = lf * lf;
01088         lfg = lf * lfg;
01089         lg = lf * lg;
01090         lgg = lf * lgg;
01091
01092         f = (f * lf) * lf;
01093         g = (g * lf) * lf;
01094
01095         lf = lf * lf;
01096         lff = lf * lf;
01097         lfg = lf * lfg;
01098         lg = lf * lg;
01099         lgg = lf * lgg;
01100
01101         f = (f * lf) * lf;
01102         g = (g * lf) * lf;
01103
01104         lf = lf * lf;
01105         lff = lf * lf;
01106         lfg = lf * lfg;
01107         lg = lf * lg;
01108         lgg = lf * lgg;
01109
01110         f = (f * lf) * lf;
01111         g = (g * lf) * lf;
01112
01113         lf = lf * lf;
01114         lff = lf * lf;
01115         lfg = lf * lfg;
01116         lg = lf * lg;
01117         lgg = lf * lgg;
01118
01119         f = (f * lf) * lf;
01120         g = (g * lf) * lf;
01121
01122         lf = lf * lf;
01123         lff = lf * lf;
01124         lfg = lf * lfg;
01125         lg = lf * lg;
01126         lgg = lf * lgg;
01127
01128         f = (f * lf) * lf;
01129         g = (g * lf) * lf;
01130
01131         lf = lf * lf;
01132         lff = lf * lf;
01133         lfg = lf * lfg;
01134         lg = lf * lg;
01135         lgg = lf * lgg;
01136
01137         f = (f * lf) * lf;
01138         g = (g * lf) * lf;
01139
01140         lf = lf * lf;
01141         lff = lf * lf;
01142         lfg = lf * lfg;
01143         lg = lf * lg;
01144         lgg = lf * lgg;
01145
01146         f = (f * lf) * lf;
01147         g = (g * lf) * lf;
01148
01149         lf = lf * lf;
01150         lff = lf * lf;
01151         lfg = lf * lfg;
01152         lg = lf * lg;
01153         lgg = lf * lgg;
01154
01155         f = (f * lf) * lf;
01156         g = (g * lf) * lf;
01157
01158         lf = lf * lf;
01159         lff = lf * lf;
01160         lfg = lf * lfg;
01161         lg = lf * lg;
01162         lgg = lf * lgg;
01163
01164         f = (f * lf) * lf;
01165         g = (g * lf) * lf;
01166
01167         lf = lf * lf;
01168         lff = lf * lf;
01169         lfg = lf * lfg;
01170         lg = lf * lg;
01171         lgg = lf * lgg;
01172
01173         f = (f * lf) * lf;
01174         g = (g * lf) * lf;
01175
01176         lf = lf * lf;
01177         lff = lf * lf;
01178         lfg = lf * lfg;
01179         lg = lf * lg;
01180         lgg = lf * lgg;
01181
01182         f = (f * lf) * lf;
01183         g = (g * lf) * lf;
01184
01185         lf = lf * lf;
01186         lff = lf * lf;
01187         lfg = lf * lfg;
01188         lg = lf * lg;
01189         lgg = lf * lgg;
01190
01191         f = (f * lf) * lf;
01192         g = (g * lf) * lf;
01193
01194         lf = lf * lf;
01195         lff = lf * lf;
01196         lfg = lf * lfg;
01197         lg = lf * lg;
01198         lgg = lf * lgg;
01199
01200         f = (f * lf) * lf;
01201         g = (g * lf) * lf;
01202
01203         lf = lf * lf;
01204         lff = lf * lf;
01205         lfg = lf * lfg;
01206         lg = lf * lg;
01207         lgg = lf * lgg;
01208
01209         f = (f * lf) * lf;
01210         g = (g * lf) * lf;
01211
01212         lf = lf * lf;
01213         lff = lf * lf;
01214         lfg = lf * lfg;
01215         lg = lf * lg;
01216         lgg = lf * lgg;
01217
01218         f = (f * lf) * lf;
01219         g = (g * lf) * lf;
01220
01221         lf = lf * lf;
01222         lff = lf * lf;
01223         lfg = lf * lfg;
01224         lg = lf * lg;
01225         lgg = lf * lgg;
01226
01227         f = (f * lf) * lf;
01228         g = (g * lf) * lf;
01229
01230         lf = lf * lf;
01231         lff = lf * lf;
01232         lfg = lf * lfg;
01233         lg = lf * lg;
01234         lgg = lf * lgg;
01235
01236         f = (f * lf) * lf;
01237         g = (g * lf) * lf;
01238
01239         lf = lf * lf;
01240         lff = lf * lf;
01241         lfg = lf * lfg;
01242         lg = lf * lg;
01243         lgg = lf * lgg;
01244
01245         f = (f * lf) * lf;
01246         g = (g * lf) * lf;
01247
01248         lf = lf * lf;
01249         lff = lf * lf;
01250         lfg = lf * lfg;
01251         lg = lf * lg;
01252         lgg = lf * lgg;
01253
01254         f = (f * lf) * lf;
01255         g = (g * lf) * lf;
01256
01257         lf = lf * lf;
01258         lff = lf * lf;
01259         lfg = lf * lfg;
01260         lg = lf * lg;
01261         lgg = lf * lgg;
01262
01263         f = (f * lf) * lf;
01264         g = (g * lf) * lf;
01265
01266         lf = lf * lf;
01267         lff = lf * lf;
01268         lfg = lf * lfg;
01269         lg = lf * lg;
01270         lgg = lf * lgg;
01271
01272         f = (f * lf) * lf;
01273         g = (g * lf) * lf;
01274
01275         lf = lf * lf;
01276         lff = lf * lf;
01277         lfg = lf * lfg;
01278         lg = lf * lg;
01279         lgg = lf * lgg;
01280
01281         f = (f * lf) * lf;
01282         g = (g * lf) * lf;
01283
01284         lf = lf * lf;
01285         lff = lf * lf;
01286         lfg = lf * lfg;
01287         lg = lf * lg;
01288         lgg = lf * lgg;
01289
01290         f = (f * lf) * lf;
01291         g = (g * lf) * lf;
01292
01293         lf = lf * lf;
01294         lff = lf * lf;
01295         lfg = lf * lfg;
01296         lg = lf * lg;
01297         lgg = lf * lgg;
01298
01299         f = (f * lf) * lf;
01300         g = (g * lf) * lf;
01301
01302         lf = lf * lf;
01303         lff = lf * lf;
01304         lfg = lf * lfg;
01305         lg = lf * lg;
01306         lgg = lf * lgg;
01307
01308         f = (f * lf) * lf;
01309         g = (g * lf) * lf;
01310
01311         lf = lf * lf;
01312         lff = lf * lf;
01313         lfg = lf * lfg;
01314         lg = lf * lg;
01315         lgg = lf * lgg;
01316
01317         f = (f * lf) * lf;
01318         g = (g * lf) * lf;
01319
01320         lf = lf * lf;
01321         lff = lf * lf;
01322         lfg = lf * lfg;
01323         lg = lf * lg;
01324         lgg = lf * lgg;
01325
01326         f = (f * lf) * lf;
01327         g = (g * lf) * lf;
01328
01329         lf = lf * lf;
01330         lff = lf * lf;
01331         lfg = lf * lfg;
01332         lg = lf * lg;
01333         lgg = lf * lgg;
01334
01335         f = (f * lf) * lf;
01336         g = (g * lf) * lf;
01337
01338         lf = lf * lf;
01339         lff = lf * lf;
01340         lfg = lf * lfg;
01341         lg = lf * lg;
01342         lgg = lf * lgg;
01343
01344         f = (f * lf) * lf;
01345         g = (g * lf) * lf;
01346
01347         lf = lf * lf;
01348         lff = lf * lf;
01349         lfg = lf * lfg;
01350         lg = lf * lg;
01351         lgg = lf * lgg;
01352
01353         f = (f * lf) * lf;
01354         g = (g * lf) * lf;
01355
01356         lf = lf * lf;
01357         lff = lf * lf;
01358         lfg = lf * lfg;
01359         lg = lf * lg;
01360         lgg = lf * lgg;
01361
01362         f = (f * lf) * lf;
01363         g = (g * lf) * lf;
01364
01365         lf = lf * lf;
01366         lff = lf * lf;
01367         lfg = lf * lfg;
01368         lg = lf * lg;
01369         lgg = lf * lgg;
01370
01371         f = (f * lf) * lf;
01372         g = (g * lf) * lf;
01373
01374         lf = lf * lf;
01375         lff = lf * lf;
01376         lfg = lf * lfg;
01377         lg = lf * lg;
01378         lgg = lf * lgg;
01379
01380         f = (f * lf) * lf;
01381         g = (g * lf) * lf;
01382
01383         lf = lf * lf;
01384         lff = lf * lf;
01385         lfg = lf * lfg;
01386         lg = lf * lg;
01387         lgg = lf * lgg;
01388
01389         f = (f * lf) * lf;
01390         g = (g * lf) * lf;
01391
01392         lf = lf * lf;
01393         lff = lf * lf;
01394         lfg = lf * lfg;
01395         lg = lf * lg;
01396         lgg = lf * lgg;
01397
01398         f = (f * lf) * lf;
01399         g = (g * lf) * lf;
01400
01401         lf = lf
```

```

00586             (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00587             (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00588             (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00589             (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00590     g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
00591         udata[pp + 2] * udata[pp + 5];
00592     switch (*c) {
00593     case 0:
00594         lf = 0;
00595         lff = 0;
00596         lfg = 0;
00597         lg = 0;
00598         lgg = 0;
00599         break;
00600     case 1:
00601         lf = 0.000206527095658582755255648 * f;
00602         lff = 0.000206527095658582755255648;
00603         lfg = 0;
00604         lg = 0.0003614224174025198216973841 * g;
00605         lgg = 0.0003614224174025198216973841;
00606         break;
00607     case 2:
00608         lf = 0.000354046449700427580438254 * f * f +
00609             0.000191775160254398272737387 * g * g;
00610         lff = 0.0007080928994008551608765075 * f;
00611         lfg = 0.0003835503205087965454747749 * g;
00612         lg = 0.0003835503205087965454747749 * f * g;
00613         lgg = 0.0003835503205087965454747749 * f;
00614         break;
00615     case 3:
00616         lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00617             f +
00618             0.000191775160254398272737387 * g * g;
00619         lff = 0.000206527095658582755255648 + 0.000708092899400855160876508 * f;
00620         lfg = 0.0003835503205087965454747749 * g;
00621         lg = (0.000361422417402519821697384 + 0.000383550320508796545474775 * f) *
00622             g;
00623         lgg = 0.000361422417402519821697384 + 0.000383550320508796545474775 * f;
00624         break;
00625     default:
00626         errorKill(
00627             "You need to specify a correct order in the weak-field expansion.");
00628     }
00629
00630     JMM[0] = lf + lff * Quad[0] +
00631         udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00632     JMM[1] =
00633         lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00634         lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00635     JMM[2] = lf + lff * Quad[1] +
00636         udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00637     JMM[3] =
00638         lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00639         lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00640     JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00641             lfg * udata[2 + pp] * udata[4 + pp] +
00642             lfg * udata[1 + pp] * udata[5 + pp] +
00643             lgg * udata[4 + pp] * udata[5 + pp];
00644     JMM[5] = lf + lff * Quad[2] +
00645         udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00646     JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00647             (-lff + lgg) * udata[pp] * udata[3 + pp];
00648     JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00649             udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00650     JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00651             udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00652     JMM[9] = -lf + lfg * Quad[0] +
00653         udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00654     JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00655             (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00656     JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00657             (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00658     JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00659             udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00660     JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00661         lff * udata[3 + pp] * udata[4 + pp] -
00662         lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00663     JMM[14] = -lf + lgg * Quad[1] +
00664         udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00665     JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00666             (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00667     JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00668             (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00669     JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00670             (-lff + lgg) * udata[2 + pp] * udata[5 + pp];
00671     JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00672             lff * udata[3 + pp] * udata[5 + pp] -

```

```

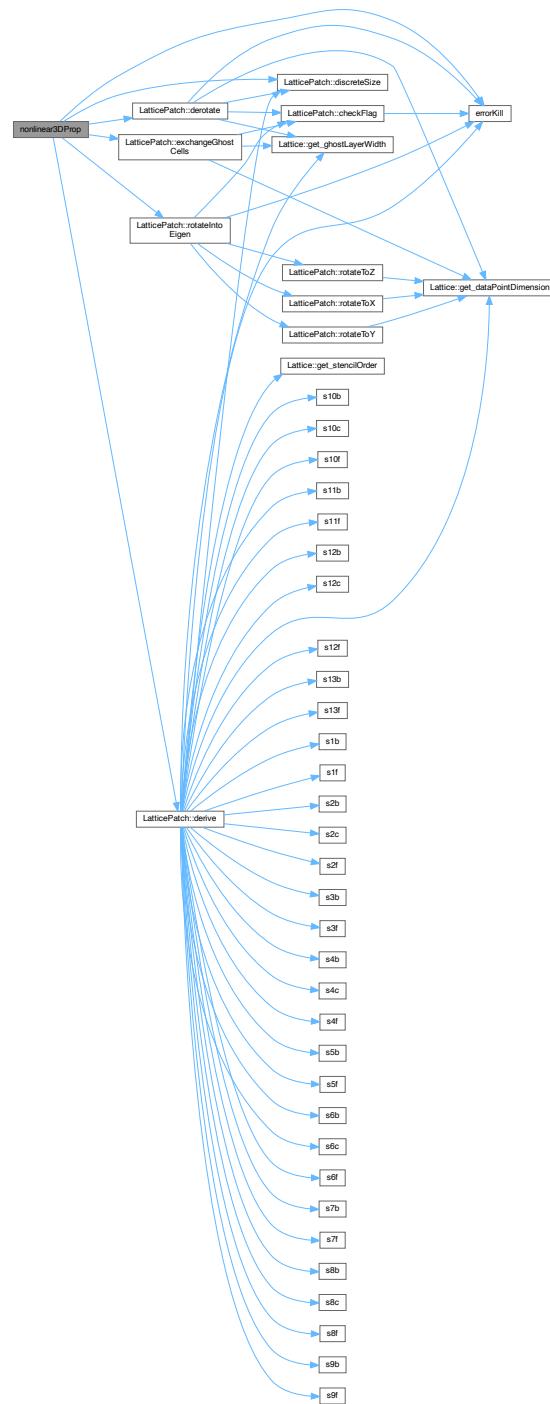
00673     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00674     JMM[19] =
00675         lgg * udata[1 + pp] * udata[2 + pp] +
00676         lff * udata[4 + pp] * udata[5 + pp] -
00677         lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]);
00678     JMM[20] = -lf + lgg * Quad[2] +
00679         udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00680
00681     h[0] = 0;
00682     h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00683         dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00684         dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00685     h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00686         dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00687         dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00688     h[3] = 0;
00689     h[4] = dxData[2 + pp];
00690     h[5] = -dxData[1 + pp];
00691     h[0] += -(dyData[pp] * JMM[15]) - dyData[1 + pp] * JMM[16] -
00692         dyData[2 + pp] * JMM[17] - dyData[3 + pp] * JMM[18] -
00693         dyData[4 + pp] * JMM[19] + dyData[5 + pp] * (1 - JMM[20]);
00694     h[1] += 0;
00695     h[2] += dyData[pp] * JMM[6] + dyData[1 + pp] * JMM[7] +
00696         dyData[2 + pp] * JMM[8] + dyData[3 + pp] * (-1 + JMM[9]) +
00697         dyData[4 + pp] * JMM[13] + dyData[5 + pp] * JMM[18];
00698     h[3] += -dyData[2 + pp];
00699     h[4] += 0;
00700     h[5] += dyData[pp];
00701     h[0] += dzData[pp] * JMM[10] + dzData[1 + pp] * JMM[11] +
00702         dzData[2 + pp] * JMM[12] + dzData[3 + pp] * JMM[13] +
00703         dzData[4 + pp] * (-1 + JMM[14]) + dzData[5 + pp] * JMM[19];
00704     h[1] += -(dzData[pp] * JMM[6]) - dzData[1 + pp] * JMM[7] -
00705         dzData[2 + pp] * JMM[8] + dzData[3 + pp] * (1 - JMM[9]) -
00706         dzData[4 + pp] * JMM[13] - dzData[5 + pp] * JMM[18];
00707     h[2] += 0;
00708     h[3] += dzData[1 + pp];
00709     h[4] += -dzData[pp];
00710     h[5] += 0;
00711     h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00712     h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00713     h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00714     dudata[pp + 0] =
00715         h[2] * (-JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00716         h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00717         h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00718     dudata[pp + 1] =
00719         h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00720         h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00721         h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00722     dudata[pp + 2] =
00723         h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00724         h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00725         h[0] * (-(1 + JMM[2]) * JMM[3] + JMM[1] * JMM[4]);
00726     detC =
00727         -(1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00728         (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00729         JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00730         JMM[1] * (-(JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00731     dudata[pp + 0] /= detC;
00732     dudata[pp + 1] /= detC;
00733     dudata[pp + 2] /= detC;
00734     dudata[pp + 3] = h[3];
00735     dudata[pp + 4] = h[4];
00736     dudata[pp + 5] = h[5];
00737 }
00738 return;
00739 }

```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [errorKill\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Referenced by [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.59 TimeEvolutionFunctions.cpp

[Go to the documentation of this file.](#)

```

00001 ///////////////////////////////////////////////////////////////////
00002 /// @file TimeEvolutionFunctions.cpp
00003 /// @brief Implementation of functions to propagate
00004 /// data vectors in time according to Maxwell's equations,
00005 /// and various orders in the HE weak-field expansion
00006 ///////////////////////////////////////////////////////////////////
00007
00008 #include "TimeEvolutionFunctions.h"
00009
00010 /// CCode right-hand-side function (CVRhsFn)
00011 int TimeEvolution::f(sunrealtype t, N_Vector u, N_Vector udot, void *data_loc) {
00012
00013     // Set recover pointer to provided lattice patch where the field data resides
00014     LatticePatch *data = static_cast<LatticePatch *>(data_loc);
00015
00016     // update circle
00017     // Access provided field values and temp. derivatievees with NVector pointers
00018 #if defined(_OPENMP)
00019     unrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00020                 *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00021 #else
00022     unrealtype *udata = NV_DATA_P(u),
00023                 *dudata = NV_DATA_P(udot);
00024 #endif
00025
00026     // Store original data location of the patch
00027     unrealtype *originaluData = data->uData,
00028                 *originalduData = data->duData;
00029
00030     // Point patch data to arguments of f
00031     data->uData = udata;
00032     data->duData = dudata;
00033
00034     // Time-evolve these arguments (the field data) with specific propagator below
00035     TimeEvolver(data, u, udot, c);
00036
00037     // Refer patch data back to original location
00038     data->uData = originaluData;
00039     data->duData = originalduData;
00040
00041     return (0);
00042 }
00043
00044 /// only under-the-hood-callable Maxwell propagation in 1D;
00045 // unused parameters 2-4 for compliance with CVRhsFn - field data is here
00046 // accessed implicitly via user data (lattice patch);
00047 // same effect as the respective nonlinear function without nonlinear terms
00048 void linear1DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c) {
00049
00050     // pointers to temporal and spatial derivative data
00051     unrealtype *duData = data->duData;
00052     unrealtype *dxData = data->buffData[1 - 1];
00053
00054     // sequence along any dimension according to the scheme:
00055     data->exchangeGhostCells(1); // -> exchange halos
00056     data->rotateIntoEigen(
00057         1); // -> rotate all data to prepare derivative operation
00058     data->derive(1); // -> perform derivative approximation operation on it
00059     data->derotate(
00060         1, dxData); // -> derotate derived data for ensuing time-evolution
00061

```

```

00062 const sunindextype totalNP = data->discreteSize();
00063 sunindextype pp = 0;
00064 for (sunindextype i = 0; i < totalNP; i++) {
00065     pp = i * 6;
00066     /*
00067         simple vacuum Maxwell equations for the temporal derivatives using the
00068         spatial derivative only in x-direction without polarization or
00069         magnetization terms
00070     */
00071     duData[pp + 0] = 0;
00072     duData[pp + 1] = -dxData[pp + 5];
00073     duData[pp + 2] = dxData[pp + 4];
00074     duData[pp + 3] = 0;
00075     duData[pp + 4] = dxData[pp + 2];
00076     duData[pp + 5] = -dxData[pp + 1];
00077 }
00078 }
00079
00080 /// nonlinear 1D HE propagation
00081 void nonlinear1DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c) {
00082
00083     // NVector pointers to provided field values and their temp. derivatives
00084 #if defined(_OPENMP)
00085     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00086                 *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00087 #else
00088     sunrealtype *udata = NV_DATA_P(u),
00089                 *dudata = NV_DATA_P(udot);
00090 #endif
00091
00092     // pointer to spatial derivatives via patch data
00093     sunrealtype *dxData = data->buffData[1 - 1];
00094
00095     // same sequence as in the linear case
00096     data->exchangeGhostCells(1);
00097     data->rotateIntoEigen(1);
00098     data->derive(1);
00099     data->derotate(1, dxData);
00100
00101     /*
00102     F and G are nonzero in the nonlinear case,
00103     polarization and magnetization derivatives
00104     w.r.t. E- and B-field go into the e.o.m.
00105     */
00106     static sunrealtype f, g; // em field invariants F, G
00107     // derivatives of HE Lagrangian w.r.t. field invariants
00108     static sunrealtype lf, lff, lfg, lg, lgg;
00109     // matrix to hold derivatives of polarization and magnetization
00110     static std::array<sunrealtype, 21> JMM;
00111     // array to hold E^2 and B^2 components
00112     static std::array<sunrealtype, 6> Quad;
00113     // array to hold intermediate temp. derivatives of E and B
00114     static std::array<sunrealtype, 6> h;
00115     // determinant needed for explicit matrix inversion
00116     static sunrealtype detC = nan("0x12345");
00117
00118     // number of points in the patch
00119     const sunindextype totalNP = data->discreteSize();
00120     #pragma omp parallel for default(none) \
00121     private(JMM, Quad, h, detC) \
00122     shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, dxData) \
00123     schedule(static)
00124     for (sunindextype pp = 0; pp < totalNP * 6;
00125          pp += 6) { // loop over all 6dim points in the patch
00126         // em field Lorentz invariants F and G
00127         f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00128                     (Quad[1] = udata[pp + 1] * udata[pp + 1]) +
00129                     (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00130                     (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00131                     (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00132                     (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00133         g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
00134             udata[pp + 2] * udata[pp + 5];
00135         // process/expansion order and corresponding derivative values of L
00136         // w.r.t. F, G
00137         switch (*c) {
00138             case 0: // linear Maxwell vacuum
00139                 lf = 0;
00140                 lff = 0;
00141                 lfg = 0;
00142                 lg = 0;
00143                 lgg = 0;
00144                 break;
00145             case 1: // only 4-photon processes
00146                 lf = 0.000206527095658582755255648 * f;
00147                 lff = 0.000206527095658582755255648;
00148                 lfg = 0;

```

```

00149     lg = 0.0003614224174025198216973841 * g;
00150     lgg = 0.0003614224174025198216973841;
00151     break;
00152 case 2: // only 6-photon processes
00153     lf = 0.000354046449700427580438254 * f * f +
00154         0.000191775160254398272737387 * g * g;
00155     lff = 0.0007080928994008551608765075 * f;
00156     lfg = 0.0003835503205087965454747749 * g;
00157     lg = 0.0003835503205087965454747749 * f * g;
00158     lgg = 0.0003835503205087965454747749 * f;
00159     break;
00160 case 3: // 4- and 6-photon processes
00161     lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00162         f +
00163         0.000191775160254398272737387 * g * g;
00164     lff = 0.000206527095658582755255648 + 0.0007080928994008551608765075 * f;
00165     lfg = 0.0003835503205087965454747749 * g;
00166     lg = (0.0003614224174025198216973841 +
00167         0.0003835503205087965454747749 * f) *
00168         g;
00169     lgg = 0.0003614224174025198216973841 + 0.0003835503205087965454747749 * f;
00170     break;
00171 default:
00172     errorKill(
00173         "You need to specify a correct order in the weak-field expansion.");
00174 }
00175
00176 // derivatives of polarization and magnetization w.r.t. E and B
00177 // Jpx(Ex)
00178 JMM[0] = lf + lff * Quad[0] +
00179     udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00180 // Jpx(Ey)
00181 JMM[1] =
00182     lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00183     lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00184 // Jpy(Ey)
00185 JMM[2] = lf + lff * Quad[1] +
00186     udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00187 // Jpx(Ez) = Jpz(Ex)
00188 JMM[3] =
00189     lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00190     lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00191 // Jpy(Ez) = Jpz(Ey)
00192 JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00193     lfg * udata[2 + pp] * udata[4 + pp] +
00194     lfg * udata[1 + pp] * udata[5 + pp] +
00195     lgg * udata[4 + pp] * udata[5 + pp];
00196 // Jpz(Ez)
00197 JMM[5] = lf + lff * Quad[2] +
00198     udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00199 // Jpx(Bx) = Jmx(Ex)
00200 JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00201     (-lff + lgg) * udata[pp] * udata[3 + pp];
00202 // Jpy(Bx) = Jmx(Ey)
00203 JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00204     udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00205 // Jpz(Bx) = Jmx(Ez)
00206 JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00207     udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00208 // Jmx(Bx)
00209 JMM[9] = -lf + lgg * Quad[0] +
00210     udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00211 // Jpx(By) = Jmy(Ex)
00212 JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00213     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00214 // Jpy(By) = Jmy(Ey)
00215 JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00216     (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00217 // Jpz(By) = Jmy(Ez)
00218 JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00219     udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00220 // Jmx(By) = Jmy(Bx)
00221 JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00222     lff * udata[3 + pp] * udata[4 + pp] -
00223     lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00224 // Jmy(By)
00225 JMM[14] = -lf + lgg * Quad[1] +
00226     udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00227 // Jmz(Ex) = Jpx(Bz)
00228 JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00229     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00230 // Jmz(Ey) = Jpy(Bz)
00231 JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00232     (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00233 // Jpz(Bz) = Jmz(Ez)
00234 JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00235     (-lff + lgg) * udata[2 + pp] * udata[5 + pp];

```

```

00236 // Jmz(Bx) = Jmx(Bz)
00237 JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00238     lff * udata[3 + pp] * udata[5 + pp] -
00239     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00240 // Jmy(Bz) = Jmz(By)
00241 JMM[19] =
00242     lgg * udata[1 + pp] * udata[2 + pp] +
00243     lff * udata[4 + pp] * udata[5 + pp] -
00244     lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]);
00245 // Jmz(Bz)
00246 JMM[20] = -lf + lgg * Quad[2] +
00247     udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00248
00249 // apply Z
00250 // top block: -QJm(E)*E, Q-QJm(B)*B
00251 h[0] = 0;
00252 h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00253     dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00254     dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00255 h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00256     dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00257     dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00258 // bottom blocks: -Q*B
00259 h[3] = 0;
00260 h[4] = dxData[2 + pp];
00261 h[5] = -dxData[1 + pp];
00262 // (1+A)^-1 applies only to E components
00263 // -Jp(B)*B
00264 h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00265 h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00266 h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00267 // apply C^-1 explicitly, with C=1+Jp(E)
00268 udata[pp + 0] =
00269     h[2] * (-JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4] +
00270     h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00271     h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00272 udata[pp + 1] =
00273     h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00274     h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00275     h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00276 udata[pp + 2] =
00277     h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00278     h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00279     h[0] * ((-1 + JMM[2]) * JMM[3]) + JMM[1] * JMM[4];
00280 detC = // determinant of C
00281     -((1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00282     (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00283     JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00284     JMM[1] * (-JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]);
00285 udata[pp + 0] /= detC;
00286 udata[pp + 1] /= detC;
00287 udata[pp + 2] /= detC;
00288 udata[pp + 3] = h[3];
00289 udata[pp + 4] = h[4];
00290 udata[pp + 5] = h[5];
00291 }
00292 return;
00293 }
00294
00295 /// only under-the-hood-callable Maxwell propagation in 2D
00296 void linear2DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c) {
00297
00298     sunrealtype *duData = data->duData;
00299     sunrealtype *dxData = data->buffData[1 - 1];
00300     sunrealtype *dyData = data->buffData[2 - 1];
00301
00302     data->exchangeGhostCells(1);
00303     data->rotateIntoEigen(1);
00304     data->derive(1);
00305     data->derotate(1, dxData);
00306     data->exchangeGhostCells(2);
00307     data->rotateIntoEigen(2);
00308     data->derive(2);
00309     data->derotate(2, dyData);
00310
00311     const sunindextype totalNP = data->discreteSize();
00312     sunindextype pp = 0;
00313     for (sunindextype i = 0; i < totalNP; i++) {
00314         pp = i * 6;
00315         duData[pp + 0] = dyData[pp + 5];
00316         duData[pp + 1] = -dxData[pp + 5];
00317         duData[pp + 2] = -dyData[pp + 3] + dxData[pp + 4];
00318         duData[pp + 3] = -dyData[pp + 2];
00319         duData[pp + 4] = dxData[pp + 2];
00320         duData[pp + 5] = dyData[pp + 0] - dxData[pp + 1];
00321     }
00322 }

```

```

00323
00324 // non-linear 2D HE propagation
00325 void nonlinear2DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c) {
00326
00327 #if defined(_OPENMP)
00328     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00329                 *udata = N_VGetArrayPointer_MPIPlusX(udot);
00330 #else
00331     sunrealtype *udata = NV_DATA_P(u),
00332                 *udata = NV_DATA_P(udot);
00333 #endif
00334
00335     sunrealtype *dxData = data->buffData[1 - 1];
00336     sunrealtype *dyData = data->buffData[2 - 1];
00337
00338     data->exchangeGhostCells(1);
00339     data->rotateIntoEigen(1);
00340     data->derive(1);
00341     data->derotate(1, dxData);
00342     data->exchangeGhostCells(2);
00343     data->rotateIntoEigen(2);
00344     data->derive(2);
00345     data->derotate(2, dyData);
00346
00347     static sunrealtype f, g;
00348     static sunrealtype lf, lff, lfg, lg, lgg;
00349     static std::array<sunrealtype, 21> JMM;
00350     static std::array<sunrealtype, 6> Quad;
00351     static std::array<sunrealtype, 6> h;
00352     static sunrealtype detC;
00353
00354     const sunindextype totalNP = data->discreteSize();
00355     #pragma omp parallel for default(none) \
00356     private(JMM, Quad, h, detC) \
00357     shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, \
00358           dxData, dyData) \
00359     schedule(static)
00360     for (sunindextype pp = 0; pp < totalNP * 6; pp += 6) {
00361         f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00362                     (Quad[1] = udata[pp + 1] * udata[pp + 1]) +
00363                     (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00364                     (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00365                     (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00366                     (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00367         g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
00368             udata[pp + 2] * udata[pp + 5];
00369         switch (*c) {
00370             case 0:
00371                 lf = 0;
00372                 lff = 0;
00373                 lfg = 0;
00374                 lg = 0;
00375                 lgg = 0;
00376                 break;
00377             case 1:
00378                 lf = 0.000206527095658582755255648 * f;
00379                 lff = 0.000206527095658582755255648;
00380                 lfg = 0;
00381                 lg = 0.0003614224174025198216973841 * g;
00382                 lgg = 0.0003614224174025198216973841;
00383                 break;
00384             case 2:
00385                 lf = 0.000354046449700427580438254 * f * f +
00386                     0.000191775160254398272737387 * g * g;
00387                 lff = 0.0007080928994008551608765075 * f;
00388                 lfg = 0.0003835503205087965454747749 * g;
00389                 lg = 0.0003835503205087965454747749 * f * g;
00390                 lgg = 0.0003835503205087965454747749 * f;
00391                 break;
00392             case 3:
00393                 lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00394                     f +
00395                     0.000191775160254398272737387 * g * g;
00396                 lff = 0.000206527095658582755255648 + 0.000708092899400855160876508 * f;
00397                 lfg = 0.0003835503205087965454747749 * g;
00398                 lg = (0.000361422417402519821697384 + 0.000383550320508796545474775 * f) *
00399                     g;
00400                 lgg = 0.000361422417402519821697384 + 0.000383550320508796545474775 * f;
00401                 break;
00402             default:
00403                 errorKill(
00404                     "You need to specify a correct order in the weak-field expansion.");
00405             }
00406
00407             JMM[0] = lf + lff * Quad[0] +
00408                 udata[3 + pp] * (2 * lfg * udata[pp] + lg * udata[3 + pp]);
00409             JMM[1] =

```

```

00410     lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00411     lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00412     JMM[2] = lf + lff * Quad[1] +
00413     udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00414     JMM[3] =
00415     lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00416     lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00417     JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00418     lfg * udata[2 + pp] * udata[4 + pp] +
00419     lfg * udata[1 + pp] * udata[5 + pp] +
00420     lgg * udata[4 + pp] * udata[5 + pp];
00421     JMM[5] = lf + lff * Quad[2] +
00422     udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00423     JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00424     (-lff + lgg) * udata[pp] * udata[3 + pp];
00425     JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00426     udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00427     JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00428     udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00429     JMM[9] = -lf + lgg * Quad[0] +
00430     udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00431     JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00432     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00433     JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00434     (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00435     JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00436     udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00437     JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00438     lff * udata[3 + pp] * udata[4 + pp] -
00439     lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00440     JMM[14] = -lf + lgg * Quad[1] +
00441     udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00442     JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00443     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00444     JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00445     (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00446     JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00447     (-lff + lgg) * udata[2 + pp] * udata[5 + pp];
00448     JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00449     lff * udata[3 + pp] * udata[5 + pp] -
00450     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00451     JMM[19] =
00452     lgg * udata[1 + pp] * udata[2 + pp] +
00453     lff * udata[4 + pp] * udata[5 + pp] -
00454     lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]);
00455     JMM[20] = -lf + lgg * Quad[2] +
00456     udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00457
00458     h[0] = 0;
00459     h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00460     dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00461     dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00462     h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00463     dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00464     dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00465     h[3] = 0;
00466     h[4] = dxData[2 + pp];
00467     h[5] = -dxData[1 + pp];
00468     h[0] += -(dyData[pp] * JMM[15]) - dyData[1 + pp] * JMM[16] -
00469     dyData[2 + pp] * JMM[17] - dyData[3 + pp] * JMM[18] -
00470     dyData[4 + pp] * JMM[19] + dyData[5 + pp] * (1 - JMM[20]);
00471     h[1] += 0;
00472     h[2] += dyData[pp] * JMM[6] + dyData[1 + pp] * JMM[7] +
00473     dyData[2 + pp] * JMM[8] + dyData[3 + pp] * (-1 + JMM[9]) +
00474     dyData[4 + pp] * JMM[13] + dyData[5 + pp] * JMM[18];
00475     h[3] += -dyData[2 + pp];
00476     h[4] += 0;
00477     h[5] += dyData[pp];
00478     h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00479     h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00480     h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00481     dudata[pp + 0] =
00482     h[2] * (-JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00483     h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00484     h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00485     dudata[pp + 1] =
00486     h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00487     h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00488     h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00489     dudata[pp + 2] =
00490     h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00491     h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00492     h[0] * (((1 + JMM[2]) * JMM[3]) + JMM[1] * JMM[4]);
00493     detC =
00494     -((1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00495     (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00496     JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -

```

```

00497     JMM[1] * (- (JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00498     dudata[pp + 0] /= detC;
00499     dudata[pp + 1] /= detC;
00500     dudata[pp + 2] /= detC;
00501     dudata[pp + 3] = h[3];
00502     dudata[pp + 4] = h[4];
00503     dudata[pp + 5] = h[5];
00504 }
00505 return;
00506 }
00507
00508 /// only under-the-hood-callable Maxwell propagation in 3D
00509 void linear3DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c) {
00510
00511     sunrealtype *duData = data->duData;
00512     sunrealtype *dxData = data->buffData[1 - 1];
00513     sunrealtype *dyData = data->buffData[2 - 1];
00514     sunrealtype *dzData = data->buffData[3 - 1];
00515
00516     data->exchangeGhostCells(1);
00517     data->rotateIntoEigen(1);
00518     data->derive(1);
00519     data->derotate(1, dxData);
00520     data->exchangeGhostCells(2);
00521     data->rotateIntoEigen(2);
00522     data->derive(2);
00523     data->derotate(2, dyData);
00524     data->exchangeGhostCells(3);
00525     data->rotateIntoEigen(3);
00526     data->derive(3);
00527     data->derotate(3, dzData);
00528
00529     const sunindextype totalNP = data->discreteSize();
00530     sunindextype pp = 0;
00531     for (sunindextype i = 0; i < totalNP; i++) {
00532         pp = i * 6;
00533         duData[pp + 0] = dyData[pp + 5] - dzData[pp + 4];
00534         duData[pp + 1] = dzData[pp + 3] - dxData[pp + 5];
00535         duData[pp + 2] = dxData[pp + 4] - dyData[pp + 3];
00536         duData[pp + 3] = -dyData[pp + 2] + dzData[pp + 1];
00537         duData[pp + 4] = -dzData[pp + 0] + dxData[pp + 2];
00538         duData[pp + 5] = -dxData[pp + 1] + dyData[pp + 0];
00539     }
00540 }
00541
00542 /// nonlinear 3D HE propagation
00543 void nonlinear3DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c) {
00544
00545 #if defined(_OPENMP)
00546     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00547                 *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00548 #else
00549     sunrealtype *udata = NV_DATA_P(u),
00550                 *dudata = NV_DATA_P(udot);
00551 #endif
00552
00553     sunrealtype *dxData = data->buffData[1 - 1];
00554     sunrealtype *dyData = data->buffData[2 - 1];
00555     sunrealtype *dzData = data->buffData[3 - 1];
00556
00557     data->exchangeGhostCells(1);
00558     data->rotateIntoEigen(1);
00559     data->derive(1);
00560     data->derotate(1, dxData);
00561     data->exchangeGhostCells(2);
00562     data->rotateIntoEigen(2);
00563     data->derive(2);
00564     data->derotate(2, dyData);
00565     data->exchangeGhostCells(3);
00566     data->rotateIntoEigen(3);
00567     data->derive(3);
00568     data->derotate(3, dzData);
00569
00570     static sunrealtype f, g;
00571     static sunrealtype lf, lff, lfg, lg, lgg;
00572     static std::array<sunrealtype, 21> JMM;
00573     static std::array<sunrealtype, 6> Quad;
00574     static std::array<sunrealtype, 6> h;
00575     static sunrealtype detC = nan("0x12345");
00576
00577     const sunindextype totalNP = data->discreteSize();
00578 #pragma omp parallel for default(none) \
00579 private(JMM, Quad, h, detC) \
00580 shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, \
00581           dxData, dyData, dzData) \
00582 schedule(static)
00583     for (sunindextype pp = 0; pp < totalNP * 6; pp += 6) {

```

```

00584     f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00585             (Quad[1] = udata[pp + 1] * udata[pp + 1]) +
00586             (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00587             (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00588             (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00589             (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00590     g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
00591         udata[pp + 2] * udata[pp + 5];
00592     switch (*c) {
00593     case 0:
00594         lf = 0;
00595         lff = 0;
00596         lfg = 0;
00597         lg = 0;
00598         lgg = 0;
00599         break;
00600     case 1:
00601         lf = 0.000206527095658582755255648 * f;
00602         lff = 0.000206527095658582755255648;
00603         lfg = 0;
00604         lg = 0.0003614224174025198216973841 * g;
00605         lgg = 0.0003614224174025198216973841;
00606         break;
00607     case 2:
00608         lf = 0.000354046449700427580438254 * f * f +
00609             0.000191775160254398272737387 * g * g;
00610         lff = 0.0007080928994008551608765075 * f;
00611         lfg = 0.0003835503205087965454747749 * g;
00612         lg = 0.0003835503205087965454747749 * f * g;
00613         lgg = 0.0003835503205087965454747749 * f;
00614         break;
00615     case 3:
00616         lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00617             f +
00618             0.000191775160254398272737387 * g * g;
00619         lff = 0.000206527095658582755255648 + 0.000708092899400855160876508 * f;
00620         lfg = 0.0003835503205087965454747749 * g;
00621         lg = (0.000361422417402519821697384 + 0.000383550320508796545474775 * f) *
00622             g;
00623         lgg = 0.000361422417402519821697384 + 0.000383550320508796545474775 * f;
00624         break;
00625     default:
00626         errorKill(
00627             "You need to specify a correct order in the weak-field expansion.");
00628     }
00629
00630     JMM[0] = lf + lff * Quad[0] +
00631         udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00632     JMM[1] =
00633         lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00634         lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00635     JMM[2] = lf + lff * Quad[1] +
00636         udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00637     JMM[3] =
00638         lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00639         lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00640     JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00641         lfg * udata[2 + pp] * udata[4 + pp] +
00642         lfg * udata[1 + pp] * udata[5 + pp] +
00643         lgg * udata[4 + pp] * udata[5 + pp];
00644     JMM[5] = lf + lff * Quad[2] +
00645         udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00646     JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00647         (-lff + lgg) * udata[pp] * udata[3 + pp];
00648     JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00649         udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00650     JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00651         udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00652     JMM[9] = -lf + lgg * Quad[0] +
00653         udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00654     JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00655         (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00656     JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00657         (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00658     JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00659         udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00660     JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00661         lff * udata[3 + pp] * udata[4 + pp] -
00662         lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00663     JMM[14] = -lf + lgg * Quad[1] +
00664         udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00665     JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00666         (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00667     JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00668         (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00669     JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00670         (-lff + lgg) * udata[2 + pp] * udata[5 + pp];

```

```

00671     JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00672         lff * udata[3 + pp] * udata[5 + pp] -
00673         lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00674     JMM[19] =
00675         lgg * udata[1 + pp] * udata[2 + pp] +
00676         lff * udata[4 + pp] * udata[5 + pp] -
00677         lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]);
00678     JMM[20] = -lf + lgg * Quad[2] +
00679         udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00680
00681     h[0] = 0;
00682     h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00683         dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00684         dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00685     h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00686         dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00687         dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00688     h[3] = 0;
00689     h[4] = dxData[2 + pp];
00690     h[5] = -dxData[1 + pp];
00691     h[0] += -(dyData[pp] * JMM[15]) - dyData[1 + pp] * JMM[16] -
00692         dyData[2 + pp] * JMM[17] - dyData[3 + pp] * JMM[18] -
00693         dyData[4 + pp] * JMM[19] + dyData[5 + pp] * (1 - JMM[20]);
00694     h[1] += 0;
00695     h[2] += dyData[pp] * JMM[6] + dyData[1 + pp] * JMM[7] +
00696         dyData[2 + pp] * JMM[8] + dyData[3 + pp] * (-1 + JMM[9]) +
00697         dyData[4 + pp] * JMM[13] + dyData[5 + pp] * JMM[18];
00698     h[3] += -dyData[2 + pp];
00699     h[4] += 0;
00700     h[5] += dyData[pp];
00701     h[0] += dzData[pp] * JMM[10] + dzData[1 + pp] * JMM[11] +
00702         dzData[2 + pp] * JMM[12] + dzData[3 + pp] * JMM[13] +
00703         dzData[4 + pp] * (-1 + JMM[14]) + dzData[5 + pp] * JMM[19];
00704     h[1] += -(dzData[pp] * JMM[6]) - dzData[1 + pp] * JMM[7] -
00705         dzData[2 + pp] * JMM[8] + dzData[3 + pp] * (1 - JMM[9]) -
00706         dzData[4 + pp] * JMM[13] - dzData[5 + pp] * JMM[18];
00707     h[2] += 0;
00708     h[3] += dzData[1 + pp];
00709     h[4] += -dzData[pp];
00710     h[5] += 0;
00711     h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00712     h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00713     h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00714     udata[pp + 0] =
00715         h[2] * (-JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00716         h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00717         h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00718     udata[pp + 1] =
00719         h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00720         h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00721         h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00722     udata[pp + 2] =
00723         h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00724         h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00725         h[0] * (((1 + JMM[2]) * JMM[3]) + JMM[1] * JMM[4]);
00726     detC =
00727         -((1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00728         (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00729         JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00730         JMM[1] * (-JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00731     udata[pp + 0] /= detC;
00732     udata[pp + 1] /= detC;
00733     udata[pp + 2] /= detC;
00734     udata[pp + 3] = h[3];
00735     udata[pp + 4] = h[4];
00736     udata[pp + 5] = h[5];
00737 }
00738 return;
00739 }

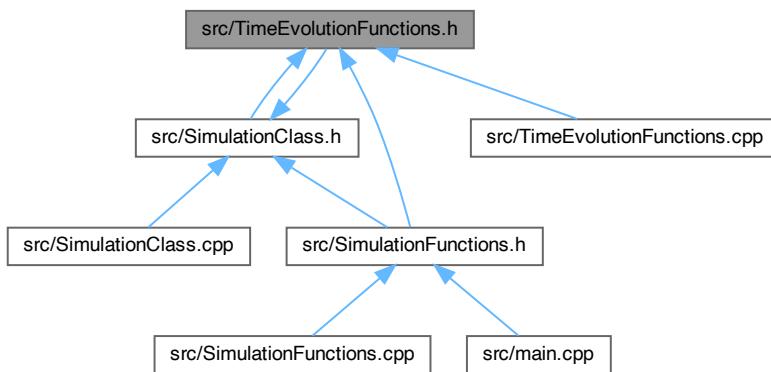
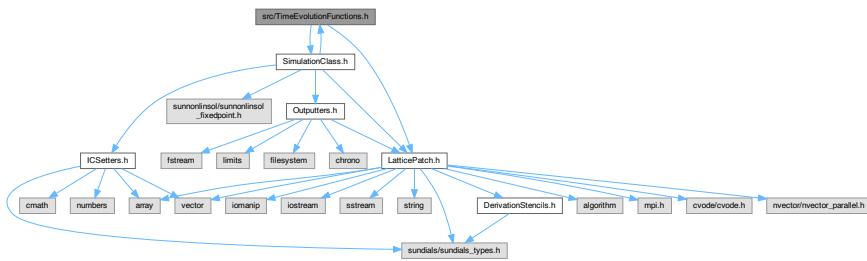
```

6.60 src/TimeEvolutionFunctions.h File Reference

Functions to propagate data vectors in time according to Maxwell's equations, and various orders in the HE weak-field expansion.

```
#include "LatticePatch.h"
#include "SimulationClass.h"
```

Include dependency graph for TimeEvolutionFunctions.h:



Data Structures

- class [TimeEvolution](#)
monostate TimeEvolution class to propagate the field data in time in a given order of the HE weak-field expansion

Functions

- void [linear1DProp](#) ([LatticePatch](#) *data, N_Vector u, N_Vector udot, int *c)
Maxwell propagation function for 1D – only for reference.
- void [nonlinear1DProp](#) ([LatticePatch](#) *data, N_Vector u, N_Vector udot, int *c)
HE propagation function for 1D.
- void [linear2DProp](#) ([LatticePatch](#) *data, N_Vector u, N_Vector udot, int *c)
Maxwell propagation function for 2D – only for reference.
- void [nonlinear2DProp](#) ([LatticePatch](#) *data, N_Vector u, N_Vector udot, int *c)
HE propagation function for 2D.
- void [linear3DProp](#) ([LatticePatch](#) *data, N_Vector u, N_Vector udot, int *c)
Maxwell propagation function for 3D – only for reference.
- void [nonlinear3DProp](#) ([LatticePatch](#) *data, N_Vector u, N_Vector udot, int *c)
HE propagation function for 3D.

6.60.1 Detailed Description

Functions to propagate data vectors in time according to Maxwell's equations, and various orders in the HE weak-field expansion.

Definition in file [TimeEvolutionFunctions.h](#).

6.60.2 Function Documentation

6.60.2.1 linear1DProp()

```
void linear1DProp (
    LatticePatch * data,
    N_Vector u,
    N_Vector udot,
    int * c )
```

Maxwell propagation function for 1D – only for reference.

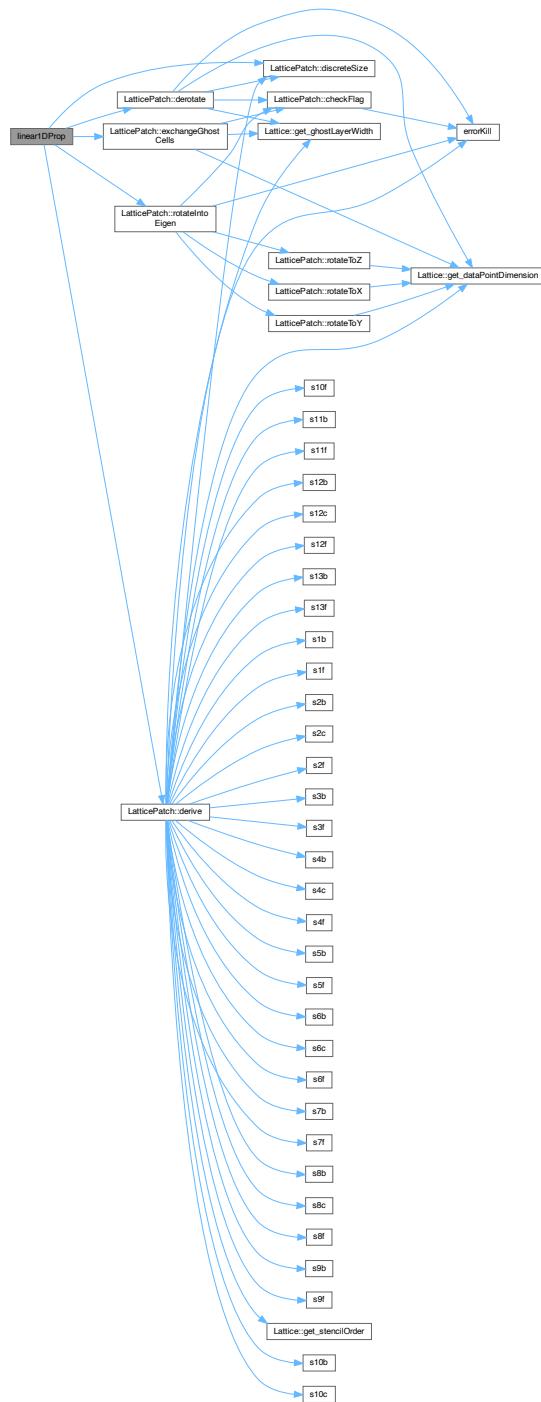
Maxwell propagation function for 1D – only for reference.

Definition at line 48 of file [TimeEvolutionFunctions.cpp](#).

```
00048
00049
00050 // pointers to temporal and spatial derivative data
00051 sunrealtype *duData = data->duData;
00052 sunrealtype *dxData = data->buffData[1 - 1];
00053
00054 // sequence along any dimension according to the scheme:
00055 data->exchangeGhostCells(1); // -> exchange halos
00056 data->rotateIntoEigen(
00057     1); // -> rotate all data to prepare derivative operation
00058 data->derive(1); // -> perform derivative approximation operation on it
00059 data->derotate(
00060     1, dxData); // -> derotate derived data for ensuing time-evolution
00061
00062 const sunindextype totalNP = data->discreteSize();
00063 sunindextype pp = 0;
00064 for (sunindextype i = 0; i < totalNP; i++) {
00065     pp = i * 6;
00066     /*
00067         simple vacuum Maxwell equations for the temporal derivatives using the
00068         spatial derivative only in x-direction without polarization or
00069         magnetization terms
00070     */
00071     duData[pp + 0] = 0;
00072     duData[pp + 1] = -dxData[pp + 5];
00073     duData[pp + 2] = dxData[pp + 4];
00074     duData[pp + 3] = 0;
00075     duData[pp + 4] = dxData[pp + 2];
00076     duData[pp + 5] = -dxData[pp + 1];
00077 }
00078 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [LatticePatch::duData](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Here is the call graph for this function:



6.60.2.2 linear2DProp()

```
void linear2DProp (
    LatticePatch * data,
```

```
N_Vector u,  
N_Vector udot,  
int * c )
```

Maxwell propagation function for 2D – only for reference.

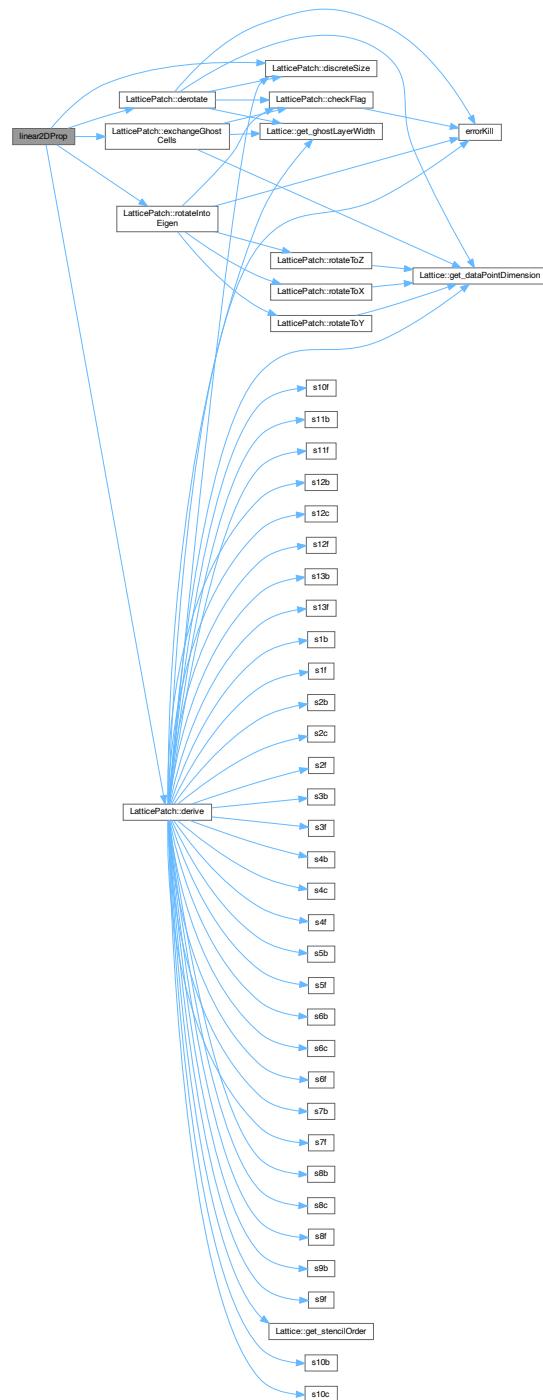
Maxwell propagation function for 2D – only for reference.

Definition at line 296 of file [TimeEvolutionFunctions.cpp](#).

```
00296  
00297  
00298     sunrealtype *duData = data->duData;  
00299     sunrealtype *dxData = data->buffData[1 - 1];  
00300     sunrealtype *dyData = data->buffData[2 - 1];  
00301  
00302     data->exchangeGhostCells(1);  
00303     data->rotateIntoEigen(1);  
00304     data->derive(1);  
00305     data->derotate(1, dxData);  
00306     data->exchangeGhostCells(2);  
00307     data->rotateIntoEigen(2);  
00308     data->derive(2);  
00309     data->derotate(2, dyData);  
00310  
00311     const sunindextype totalNP = data->discreteSize();  
00312     sunindextype pp = 0;  
00313     for (sunindextype i = 0; i < totalNP; i++) {  
00314         pp = i * 6;  
00315         duData[pp + 0] = dyData[pp + 5];  
00316         duData[pp + 1] = -dxData[pp + 5];  
00317         duData[pp + 2] = -dyData[pp + 3] + dxData[pp + 4];  
00318         duData[pp + 3] = -dyData[pp + 2];  
00319         duData[pp + 4] = dxData[pp + 2];  
00320         duData[pp + 5] = dyData[pp + 0] - dxData[pp + 1];  
00321     }  
00322 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [LatticePatch::duData](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Here is the call graph for this function:



6.60.2.3 linear3DProp()

```
void linear3DProp (
    LatticePatch * data,
```

```
N_Vector u,
N_Vector udot,
int * c )
```

Maxwell propagation function for 3D – only for reference.

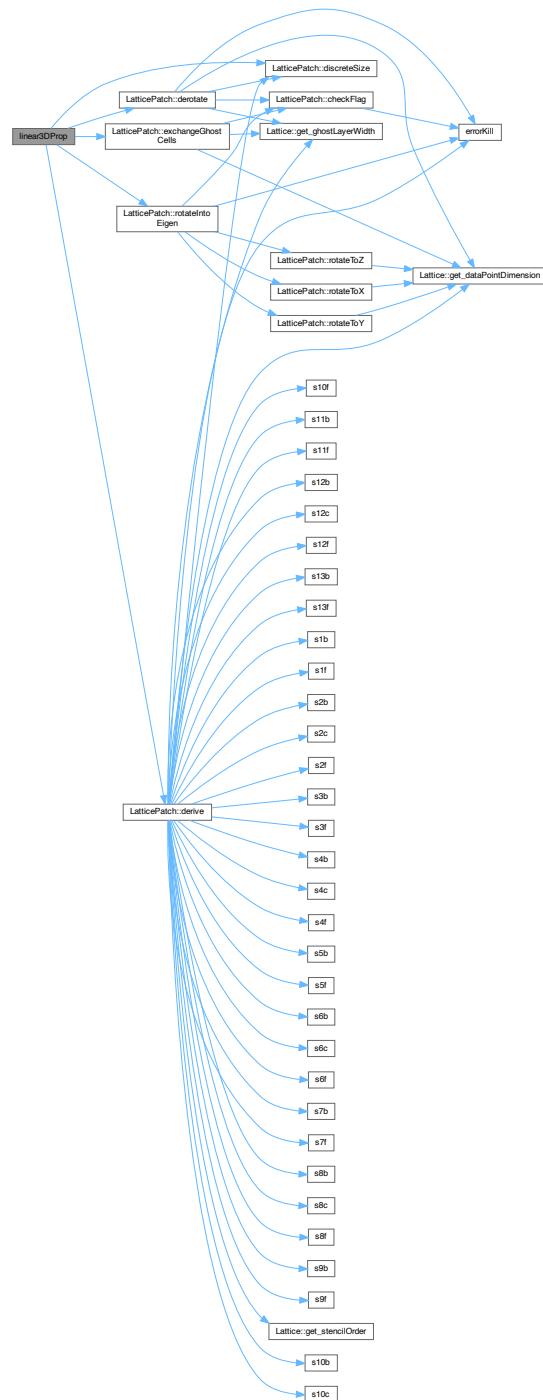
Maxwell propagation function for 3D – only for reference.

Definition at line 509 of file TimeEvolutionFunctions.cpp.

```
00509
00510
00511     sunrealtype *duData = data->duData;
00512     sunrealtype *dxData = data->buffData[1 - 1];
00513     sunrealtype *dyData = data->buffData[2 - 1];
00514     sunrealtype *dzData = data->buffData[3 - 1];
00515
00516     data->exchangeGhostCells(1);
00517     data->rotateIntoEigen(1);
00518     data->derive(1);
00519     data->derotate(1, dxData);
00520     data->exchangeGhostCells(2);
00521     data->rotateIntoEigen(2);
00522     data->derive(2);
00523     data->derotate(2, dyData);
00524     data->exchangeGhostCells(3);
00525     data->rotateIntoEigen(3);
00526     data->derive(3);
00527     data->derotate(3, dzData);
00528
00529     const sunindextype totalNP = data->discreteSize();
00530     sunindextype pp = 0;
00531     for (sunindextype i = 0; i < totalNP; i++) {
00532         pp = i * 6;
00533         duData[pp + 0] = dyData[pp + 5] - dzData[pp + 4];
00534         duData[pp + 1] = dzData[pp + 3] - dxData[pp + 5];
00535         duData[pp + 2] = dxData[pp + 4] - dyData[pp + 3];
00536         duData[pp + 3] = -dyData[pp + 2] + dzData[pp + 1];
00537         duData[pp + 4] = -dzData[pp + 0] + dxData[pp + 2];
00538         duData[pp + 5] = -dxData[pp + 1] + dyData[pp + 0];
00539     }
00540 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [LatticePatch::duData](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Here is the call graph for this function:



6.60.2.4 nonlinear1DProp()

```
void nonlinear1DProp (
    LatticePatch * data,
```

```
N_Vector u,
N_Vector udot,
int * c )
```

HE propagation function for 1D.

HE propagation function for 1D.

Definition at line 81 of file [TimeEvolutionFunctions.cpp](#).

```
00081
00082
00083 // NVector pointers to provided field values and their temp. derivatives
00084 #if defined(_OPENMP)
00085 sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00086             *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00087 #else
00088 sunrealtype *udata = NV_DATA_P(u),
00089             *dudata = NV_DATA_P(udot);
00090 #endif
00091
00092 // pointer to spatial derivatives via pach data
00093 sunrealtype *dxData = data->buffData[1 - 1];
00094
00095 // same sequence as in the linear case
00096 data->exchangeGhostCells(1);
00097 data->rotateIntoEigen(1);
00098 data->derive(1);
00099 data->derotate(1, dxData);
00100
00101 /*
00102 F and G are nonzero in the nonlinear case,
00103 polarization and magnetization derivatives
00104 w.r.t. E- and B-field go into the e.o.m.
00105 */
00106 static sunrealtype f, g; // em field invariants F, G
00107 // derivatives of HE Lagrangian w.r.t. field invariants
00108 static sunrealtype lf, lff, lfg, lg, lgg;
00109 // matrix to hold derivatives of polarization and magnetization
00110 static std::array<sunrealtype, 21> JMM;
00111 // array to hold E^2 and B^2 components
00112 static std::array<sunrealtype, 6> Quad;
00113 // array to hold intermediate temp. derivatives of E and B
00114 static std::array<sunrealtype, 6> h;
00115 // determinant needed for explicit matrix inversion
00116 static sunrealtype detC = nan("0x12345");
00117
00118 // number of points in the patch
00119 const sunindextype totalNP = data->discreteSize();
00120 #pragma omp parallel for default(none) \
00121 private(JMM, Quad, h, detC) \
00122 shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, dxData) \
00123 schedule(static)
00124 for (sunindextype pp = 0; pp < totalNP * 6;
00125     pp += 6) { // loop over all 6dim points in the patch
00126     // em field Lorentz invariants F and G
00127     f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00128                 (Quad[1] = udata[pp + 1] * udata[pp + 1]) +
00129                 (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00130                 (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00131                 (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00132                 (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00133     g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
00134         udata[pp + 2] * udata[pp + 5];
00135     // process/expansion order and corresponding derivative values of L
00136     // w.r.t. F, G
00137     switch (*c) {
00138     case 0: // linear Maxwell vacuum
00139         lf = 0;
00140         lff = 0;
00141         lfg = 0;
00142         lg = 0;
00143         lgg = 0;
00144         break;
00145     case 1: // only 4-photon processes
00146         lf = 0.000206527095658582755255648 * f;
00147         lff = 0.000206527095658582755255648;
00148         lfg = 0;
00149         lg = 0.0003614224174025198216973841 * g;
00150         lgg = 0.0003614224174025198216973841;
00151         break;
00152     case 2: // only 6-photon processes
00153         lf = 0.000354046449700427580438254 * f * f +
00154             0.000191775160254398272737387 * g * g;
```

```

00155     lff = 0.0007080928994008551608765075 * f;
00156     lfg = 0.0003835503205087965454747749 * g;
00157     lg = 0.0003835503205087965454747749 * f * g;
00158     lgg = 0.0003835503205087965454747749 * f;
00159     break;
00160 case 3: // 4- and 6-photon processes
00161     lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00162         f +
00163         0.000191775160254398272737387 * g * g;
00164     lff = 0.000206527095658582755255648 + 0.0007080928994008551608765075 * f;
00165     lfg = 0.0003835503205087965454747749 * g;
00166     lg = (0.0003614224174025198216973841 +
00167         0.0003835503205087965454747749 * f) *
00168         g;
00169     lgg = 0.0003614224174025198216973841 + 0.0003835503205087965454747749 * f;
00170     break;
00171 default:
00172     errorKill(
00173         "You need to specify a correct order in the weak-field expansion.");
00174 }
00175
00176 // derivatives of polarization and magnetization w.r.t. E and B
00177 // Jpx(Ex)
00178 JMM[0] = lf + lff * Quad[0] +
00179     udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00180 // Jpx(Ey)
00181 JMM[1] =
00182     lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00183     lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00184 // Jpy(Ey)
00185 JMM[2] = lf + lff * Quad[1] +
00186     udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00187 // Jpx(Ez) = Jpz(Ex)
00188 JMM[3] =
00189     lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00190     lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00191 // Jpy(Ez) = Jpz(Ey)
00192 JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00193     lfg * udata[2 + pp] * udata[4 + pp] +
00194     lfg * udata[1 + pp] * udata[5 + pp] +
00195     lgg * udata[4 + pp] * udata[5 + pp];
00196 // Jpz(Ez)
00197 JMM[5] = lf + lff * Quad[2] +
00198     udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00199 // Jpx(Bx) = Jmx(Ex)
00200 JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00201     (-lff + lgg) * udata[pp] * udata[3 + pp];
00202 // Jpy(Bx) = Jmx(Ey)
00203 JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00204     udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00205 // Jpz(Bx) = Jmx(Ez)
00206 JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00207     udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00208 // Jmx(Bx)
00209 JMM[9] = -lf + lgg * Quad[0] +
00210     udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00211 // Jpx(By) = Jmy(Ex)
00212 JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00213     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00214 // Jpy(By) = Jmy(Ey)
00215 JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00216     (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00217 // Jpz(By) = Jmy(Ez)
00218 JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00219     udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00220 // Jmx(By) = Jmy(Bx)
00221 JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00222     lff * udata[3 + pp] * udata[4 + pp] -
00223     lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00224 // Jmy(By)
00225 JMM[14] = -lf + lgg * Quad[1] +
00226     udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00227 // Jmz(Ex) = Jpx(Bz)
00228 JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00229     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00230 // Jmz(Ey) = Jpy(Bz)
00231 JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00232     (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00233 // Jpz(Bz) = Jmz(Ez)
00234 JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00235     (-lff + lgg) * udata[2 + pp] * udata[5 + pp];
00236 // Jmz(Bx) = Jmx(Bz)
00237 JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00238     lff * udata[3 + pp] * udata[5 + pp] -
00239     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00240 // Jmy(Bz) = Jmz(By)
00241 JMM[19] =

```

```

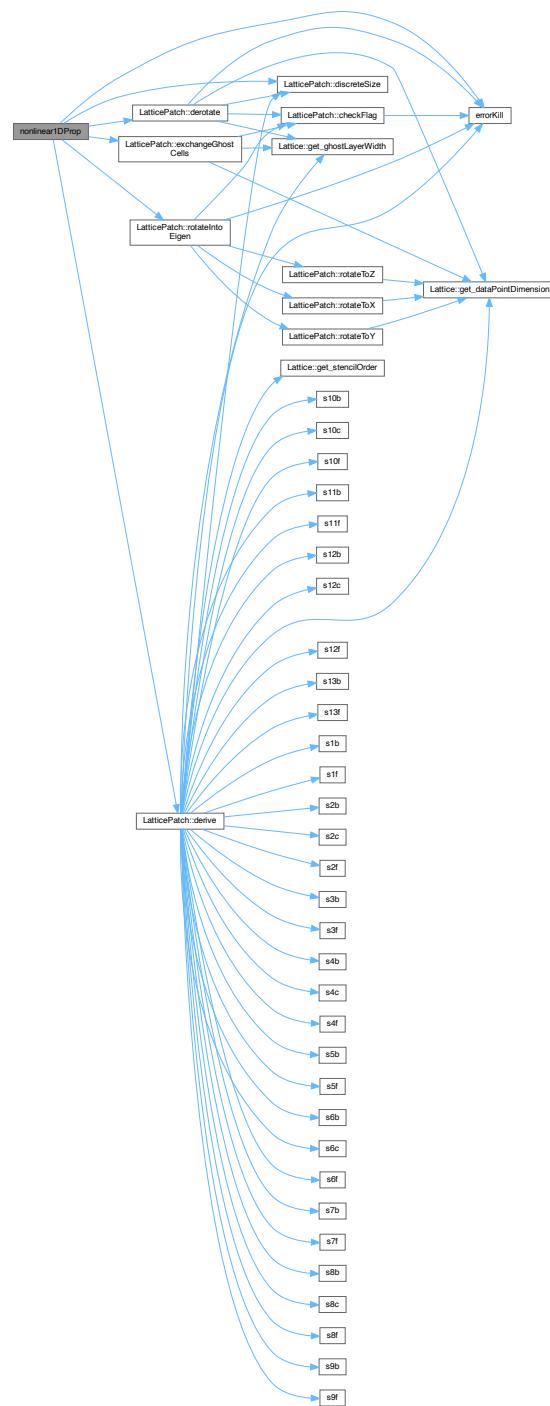
00242     lgg * udata[1 + pp] * udata[2 + pp] +
00243     lff * udata[4 + pp] * udata[5 + pp] -
00244     lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]));
00245 // Jmz(Bz)
00246 JMM[20] = -lf + lgg * Quad[2] +
00247     udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00248
00249 // apply Z
00250 // top block: -QJm(E)*E, Q-QJm(B)*B
00251 h[0] = 0;
00252 h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00253     dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00254     dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00255 h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00256     dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00257     dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00258 // bottom blocks: -Q*E
00259 h[3] = 0;
00260 h[4] = dxData[2 + pp];
00261 h[5] = -dxData[1 + pp];
00262 // (1+A)^-1 applies only to E components
00263 // -Jp(B)*B
00264 h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00265 h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00266 h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00267 // apply C^-1 explicitly, with C=1+Jp(E)
00268 dudata[pp + 0] =
00269     h[2] * (-(JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00270     h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00271     h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00272 dudata[pp + 1] =
00273     h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00274     h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00275     h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00276 dudata[pp + 2] =
00277     h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00278     h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00279     h[0] * (-(1 + JMM[2]) * JMM[3]) + JMM[1] * JMM[4]);
00280 detC = // determinant of C
00281     -((1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00282     (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00283     JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00284     JMM[1] * (-(JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00285 dudata[pp + 0] /= detC;
00286 dudata[pp + 1] /= detC;
00287 dudata[pp + 2] /= detC;
00288 dudata[pp + 3] = h[3];
00289 dudata[pp + 4] = h[4];
00290 dudata[pp + 5] = h[5];
00291 }
00292 return;
00293 }

```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [errorKill\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Referenced by [Sim1D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.60.2.5 nonlinear2DProp()

```
void nonlinear2DProp (
    LatticePatch * data,
    N_Vector u,
    N_Vector udot,
    int * c )
```

HE propagation function for 2D.

HE propagation function for 2D.

Definition at line 325 of file [TimeEvolutionFunctions.cpp](#).

```
00325
00326
00327 #if defined(_OPENMP)
00328     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00329                 *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00330 #else
00331     sunrealtype *udata = NV_DATA_P(u),
00332                 *dudata = NV_DATA_P(udot);
00333 #endif
00334
00335     sunrealtype *dxData = data->buffData[1 - 1];
00336     sunrealtype *dyData = data->buffData[2 - 1];
00337
00338     data->exchangeGhostCells(1);
00339     data->rotateIntoEigen(1);
00340     data->derive(1);
00341     data->derotate(1, dxData);
00342     data->exchangeGhostCells(2);
00343     data->rotateIntoEigen(2);
00344     data->derive(2);
00345     data->derotate(2, dyData);
00346
00347     static sunrealtype f, g;
00348     static sunrealtype lf, lff, lfg, lg, lgg;
00349     static std::array<sunrealtype, 21> JMM;
00350     static std::array<sunrealtype, 6> Quad;
00351     static std::array<sunrealtype, 6> h;
00352     static sunrealtype detC;
00353
00354     const sunindextype totalNP = data->discreteSize();
00355     #pragma omp parallel for default(none) \
00356     private(JMM, Quad, h, detC) \
00357     shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, \
00358             dxData, dyData) \
00359     schedule(static)
00360     for (sunindextype pp = 0; pp < totalNP * 6; pp += 6) {
00361         f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00362                     (Quad[1] = udata[pp + 1] * udata[pp + 1])) +
00363                     (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00364                     (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00365                     (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00366                     (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00367         g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
```

```

00368     udata[pp + 2] * udata[pp + 5];
00369     switch (*c) {
00370     case 0:
00371         lf = 0;
00372         lff = 0;
00373         lfg = 0;
00374         lg = 0;
00375         lgg = 0;
00376         break;
00377     case 1:
00378         lf = 0.000206527095658582755255648 * f;
00379         lff = 0.000206527095658582755255648;
00380         lfg = 0;
00381         lg = 0.0003614224174025198216973841 * g;
00382         lgg = 0.0003614224174025198216973841;
00383         break;
00384     case 2:
00385         lf = 0.000354046449700427580438254 * f * f +
00386             0.000191775160254398272737387 * g * g;
00387         lff = 0.0007080928994008551608765075 * f;
00388         lfg = 0.0003835503205087965454747749 * g;
00389         lg = 0.0003835503205087965454747749 * f * g;
00390         lgg = 0.0003835503205087965454747749 * f;
00391         break;
00392     case 3:
00393         lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00394             f +
00395             0.000191775160254398272737387 * g * g;
00396         lff = 0.000206527095658582755255648 + 0.000708092899400855160876508 * f;
00397         lfg = 0.0003835503205087965454747749 * g;
00398         lg = (0.000361422417402519821697384 + 0.000383550320508796545474775 * f) *
00399             g;
00400         lgg = 0.000361422417402519821697384 + 0.000383550320508796545474775 * f;
00401         break;
00402     default:
00403         errorKill(
00404             "You need to specify a correct order in the weak-field expansion.");
00405     }
00406
00407 JMM[0] = lf + lff * Quad[0] +
00408     udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00409 JMM[1] =
00410     lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00411     lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00412 JMM[2] = lf + lff * Quad[1] +
00413     udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00414 JMM[3] =
00415     lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00416     lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00417 JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00418     lfg * udata[2 + pp] * udata[4 + pp] +
00419     lfg * udata[1 + pp] * udata[5 + pp] +
00420     lgg * udata[4 + pp] * udata[5 + pp];
00421 JMM[5] = lf + lff * Quad[2] +
00422     udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00423 JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00424     (-lff + lgg) * udata[pp] * udata[3 + pp];
00425 JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00426     udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00427 JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00428     udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00429 JMM[9] = -lf + lgg * Quad[0] +
00430     udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00431 JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00432     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00433 JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00434     (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00435 JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00436     udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00437 JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00438     lff * udata[3 + pp] * udata[4 + pp] -
00439     lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00440 JMM[14] = -lf + lgg * Quad[1] +
00441     udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00442 JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00443     (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00444 JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00445     (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00446 JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00447     (-lff + lgg) * udata[2 + pp] * udata[5 + pp];
00448 JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00449     lff * udata[3 + pp] * udata[5 + pp] -
00450     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00451 JMM[19] =
00452     lgg * udata[1 + pp] * udata[2 + pp] +
00453     lff * udata[4 + pp] * udata[5 + pp] -
00454     lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]);

```

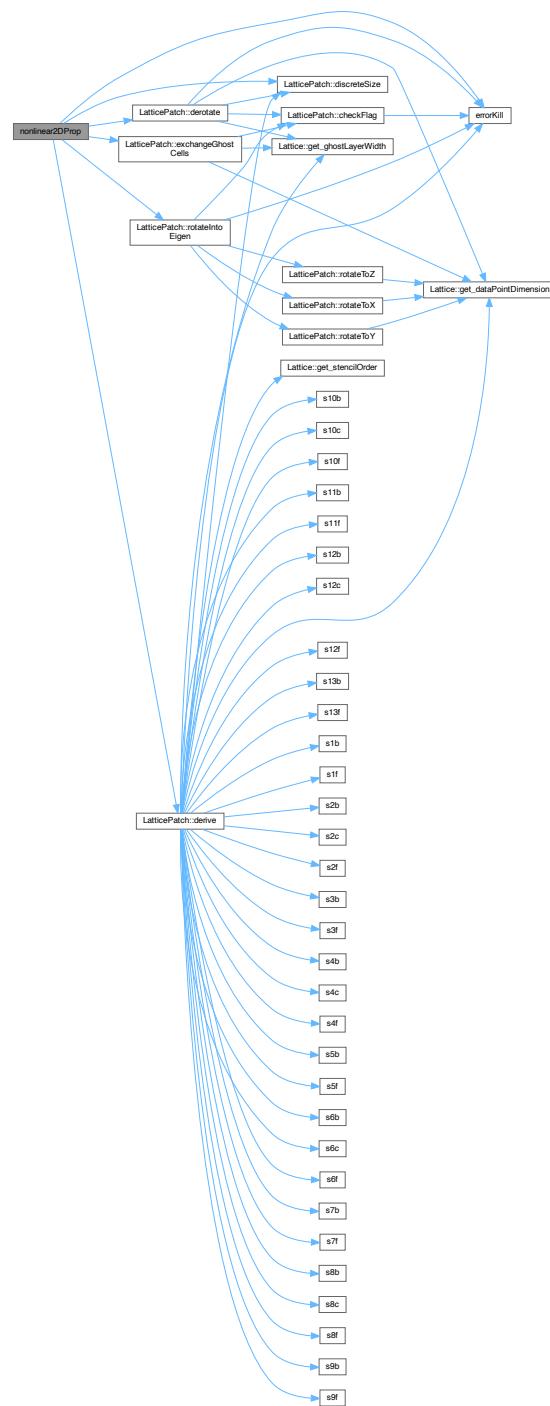
```

00455     JMM[20] = -lf + lgg * Quad[2] +
00456         udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00457
00458     h[0] = 0;
00459     h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00460         dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00461         dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00462     h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00463         dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00464         dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00465     h[3] = 0;
00466     h[4] = dxData[2 + pp];
00467     h[5] = -dxData[1 + pp];
00468     h[0] += -(dyData[pp] * JMM[15]) - dyData[1 + pp] * JMM[16] -
00469         dyData[2 + pp] * JMM[17] - dyData[3 + pp] * JMM[18] -
00470         dyData[4 + pp] * JMM[19] + dyData[5 + pp] * (1 - JMM[20]);
00471     h[1] += 0;
00472     h[2] += dyData[pp] * JMM[6] + dyData[1 + pp] * JMM[7] +
00473         dyData[2 + pp] * JMM[8] + dyData[3 + pp] * (-1 + JMM[9]) +
00474         dyData[4 + pp] * JMM[13] + dyData[5 + pp] * JMM[18];
00475     h[3] += -dyData[2 + pp];
00476     h[4] += 0;
00477     h[5] += dyData[pp];
00478     h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00479     h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00480     h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00481     dudata[pp + 0] =
00482         h[2] * (- (JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00483         h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00484         h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00485     dudata[pp + 1] =
00486         h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00487         h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00488         h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00489     dudata[pp + 2] =
00490         h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00491         h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00492         h[0] * (-((1 + JMM[2]) * JMM[3]) + JMM[1] * JMM[4]);
00493     detC =
00494         -((1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00495         (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00496         JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00497         JMM[1] * (- (JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00498     dudata[pp + 0] /= detC;
00499     dudata[pp + 1] /= detC;
00500     dudata[pp + 2] /= detC;
00501     dudata[pp + 3] = h[3];
00502     dudata[pp + 4] = h[4];
00503     dudata[pp + 5] = h[5];
00504 }
00505 return;
00506 }
```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [errorKill\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Referenced by [Sim2D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.60.2.6 nonlinear3DProp()

```
void nonlinear3DProp (
    LatticePatch * data,
    N_Vector u,
    N_Vector udot,
    int * c )
```

HE propagation function for 3D.

HE propagation function for 3D.

Definition at line 543 of file [TimeEvolutionFunctions.cpp](#).

```
00543
00544
00545 #if defined(_OPENMP)
00546     sunrealtype *udata = N_VGetArrayPointer_MPIPlusX(u),
00547             *dudata = N_VGetArrayPointer_MPIPlusX(udot);
00548 #else
00549     sunrealtype *udata = NV_DATA_P(u),
00550         *dudata = NV_DATA_P(udot);
00551 #endif
00552
00553     sunrealtype *dxData = data->buffData[1 - 1];
00554     sunrealtype *dyData = data->buffData[2 - 1];
00555     sunrealtype *dzData = data->buffData[3 - 1];
00556
00557     data->exchangeGhostCells(1);
00558     data->rotateIntoEigen(1);
00559     data->derive(1);
00560     data->derotate(1,dxData);
00561     data->exchangeGhostCells(2);
00562     data->rotateIntoEigen(2);
00563     data->derive(2);
00564     data->derotate(2,dyData);
00565     data->exchangeGhostCells(3);
00566     data->rotateIntoEigen(3);
00567     data->derive(3);
00568     data->derotate(3,dzData);
00569
00570     static sunrealtype f, g;
00571     static sunrealtype lf, lff, lfg, lg, lgg;
00572     static std::array<sunrealtype, 21> JMM;
00573     static std::array<sunrealtype, 6> Quad;
00574     static std::array<sunrealtype, 6> h;
00575     static sunrealtype detC = nan("0x12345");
00576
00577     const sunindextype totalNP = data->discreteSize();
00578     #pragma omp parallel for default(none) \
00579     private(JMM, Quad, h, detC) \
00580     shared(totalNP, c, f, g, lf, lff, lfg, lg, lgg, udata, dudata, \
00581           dxData, dyData, dzData) \
00582     schedule(static)
00583     for (sunindextype pp = 0; pp < totalNP * 6; pp += 6) {
00584         f = 0.5 * ((Quad[0] = udata[pp] * udata[pp]) +
00585                     (Quad[1] = udata[pp + 1] * udata[pp + 1]) +
00586                     (Quad[2] = udata[pp + 2] * udata[pp + 2]) +
00587                     (Quad[3] = udata[pp + 3] * udata[pp + 3]) +
00588                     (Quad[4] = udata[pp + 4] * udata[pp + 4]) +
00589                     (Quad[5] = udata[pp + 5] * udata[pp + 5])) / 6.0;
00590
00591         lf = (lff = (lg = (lgg = (f * f) * f) * f) * f) * f;
00592
00593         lfg = lf * lff;
00594         lg = lf * lg;
00595         lgg = lf * lgg;
00596
00597         f = lf * f;
00598
00599         lf = lf * lf;
00600         lff = lf * lff;
00601         lfg = lf * lfg;
00602         lg = lf * lg;
00603         lgg = lf * lgg;
00604
00605         lf = lf * lf;
00606         lff = lf * lff;
00607         lfg = lf * lfg;
00608         lg = lf * lg;
00609         lgg = lf * lgg;
00610
00611         lf = lf * lf;
00612         lff = lf * lff;
00613         lfg = lf * lfg;
00614         lg = lf * lg;
00615         lgg = lf * lgg;
00616
00617         lf = lf * lf;
00618         lff = lf * lff;
00619         lfg = lf * lfg;
00620         lg = lf * lg;
00621         lgg = lf * lgg;
00622
00623         lf = lf * lf;
00624         lff = lf * lff;
00625         lfg = lf * lfg;
00626         lg = lf * lg;
00627         lgg = lf * lgg;
00628
00629         lf = lf * lf;
00630         lff = lf * lff;
00631         lfg = lf * lfg;
00632         lg = lf * lg;
00633         lgg = lf * lgg;
00634
00635         lf = lf * lf;
00636         lff = lf * lff;
00637         lfg = lf * lfg;
00638         lg = lf * lg;
00639         lgg = lf * lgg;
00640
00641         lf = lf * lf;
00642         lff = lf * lff;
00643         lfg = lf * lfg;
00644         lg = lf * lg;
00645         lgg = lf * lgg;
00646
00647         lf = lf * lf;
00648         lff = lf * lff;
00649         lfg = lf * lfg;
00650         lg = lf * lg;
00651         lgg = lf * lgg;
00652
00653         lf = lf * lf;
00654         lff = lf * lff;
00655         lfg = lf * lfg;
00656         lg = lf * lg;
00657         lgg = lf * lgg;
00658
00659         lf = lf * lf;
00660         lff = lf * lff;
00661         lfg = lf * lfg;
00662         lg = lf * lg;
00663         lgg = lf * lgg;
00664
00665         lf = lf * lf;
00666         lff = lf * lff;
00667         lfg = lf * lfg;
00668         lg = lf * lg;
00669         lgg = lf * lgg;
00670
00671         lf = lf * lf;
00672         lff = lf * lff;
00673         lfg = lf * lfg;
00674         lg = lf * lg;
00675         lgg = lf * lgg;
00676
00677         lf = lf * lf;
00678         lff = lf * lff;
00679         lfg = lf * lfg;
00680         lg = lf * lg;
00681         lgg = lf * lgg;
00682
00683         lf = lf * lf;
00684         lff = lf * lff;
00685         lfg = lf * lfg;
00686         lg = lf * lg;
00687         lgg = lf * lgg;
00688
00689         lf = lf * lf;
00690         lff = lf * lff;
00691         lfg = lf * lfg;
00692         lg = lf * lg;
00693         lgg = lf * lgg;
00694
00695         lf = lf * lf;
00696         lff = lf * lff;
00697         lfg = lf * lfg;
00698         lg = lf * lg;
00699         lgg = lf * lgg;
00700
00701         lf = lf * lf;
00702         lff = lf * lff;
00703         lfg = lf * lfg;
00704         lg = lf * lg;
00705         lgg = lf * lgg;
00706
00707         lf = lf * lf;
00708         lff = lf * lff;
00709         lfg = lf * lfg;
00710         lg = lf * lg;
00711         lgg = lf * lgg;
00712
00713         lf = lf * lf;
00714         lff = lf * lff;
00715         lfg = lf * lfg;
00716         lg = lf * lg;
00717         lgg = lf * lgg;
00718
00719         lf = lf * lf;
00720         lff = lf * lff;
00721         lfg = lf * lfg;
00722         lg = lf * lg;
00723         lgg = lf * lgg;
00724
00725         lf = lf * lf;
00726         lff = lf * lff;
00727         lfg = lf * lfg;
00728         lg = lf * lg;
00729         lgg = lf * lgg;
00730
00731         lf = lf * lf;
00732         lff = lf * lff;
00733         lfg = lf * lfg;
00734         lg = lf * lg;
00735         lgg = lf * lgg;
00736
00737         lf = lf * lf;
00738         lff = lf * lff;
00739         lfg = lf * lfg;
00740         lg = lf * lg;
00741         lgg = lf * lgg;
00742
00743         lf = lf * lf;
00744         lff = lf * lff;
00745         lfg = lf * lfg;
00746         lg = lf * lg;
00747         lgg = lf * lgg;
00748
00749         lf = lf * lf;
00750         lff = lf * lff;
00751         lfg = lf * lfg;
00752         lg = lf * lg;
00753         lgg = lf * lgg;
00754
00755         lf = lf * lf;
00756         lff = lf * lff;
00757         lfg = lf * lfg;
00758         lg = lf * lg;
00759         lgg = lf * lgg;
00760
00761         lf = lf * lf;
00762         lff = lf * lff;
00763         lfg = lf * lfg;
00764         lg = lf * lg;
00765         lgg = lf * lgg;
00766
00767         lf = lf * lf;
00768         lff = lf * lff;
00769         lfg = lf * lfg;
00770         lg = lf * lg;
00771         lgg = lf * lgg;
00772
00773         lf = lf * lf;
00774         lff = lf * lff;
00775         lfg = lf * lfg;
00776         lg = lf * lg;
00777         lgg = lf * lgg;
00778
00779         lf = lf * lf;
00780         lff = lf * lff;
00781         lfg = lf * lfg;
00782         lg = lf * lg;
00783         lgg = lf * lgg;
00784
00785         lf = lf * lf;
00786         lff = lf * lff;
00787         lfg = lf * lfg;
00788         lg = lf * lg;
00789         lgg = lf * lgg;
00790
00791         lf = lf * lf;
00792         lff = lf * lff;
00793         lfg = lf * lfg;
00794         lg = lf * lg;
00795         lgg = lf * lgg;
00796
00797         lf = lf * lf;
00798         lff = lf * lff;
00799         lfg = lf * lfg;
00800         lg = lf * lg;
00801         lgg = lf * lgg;
00802
00803         lf = lf * lf;
00804         lff = lf * lff;
00805         lfg = lf * lfg;
00806         lg = lf * lg;
00807         lgg = lf * lgg;
00808
00809         lf = lf * lf;
00810         lff = lf * lff;
00811         lfg = lf * lfg;
00812         lg = lf * lg;
00813         lgg = lf * lgg;
00814
00815         lf = lf * lf;
00816         lff = lf * lff;
00817         lfg = lf * lfg;
00818         lg = lf * lg;
00819         lgg = lf * lgg;
00820
00821         lf = lf * lf;
00822         lff = lf * lff;
00823         lfg = lf * lfg;
00824         lg = lf * lg;
00825         lgg = lf * lgg;
00826
00827         lf = lf * lf;
00828         lff = lf * lff;
00829         lfg = lf * lfg;
00830         lg = lf * lg;
00831         lgg = lf * lgg;
00832
00833         lf = lf * lf;
00834         lff = lf * lff;
00835         lfg = lf * lfg;
00836         lg = lf * lg;
00837         lgg = lf * lgg;
00838
00839         lf = lf * lf;
00840         lff = lf * lff;
00841         lfg = lf * lfg;
00842         lg = lf * lg;
00843         lgg = lf * lgg;
00844
00845         lf = lf * lf;
00846         lff = lf * lff;
00847         lfg = lf * lfg;
00848         lg = lf * lg;
00849         lgg = lf * lgg;
00850
00851         lf = lf * lf;
00852         lff = lf * lff;
00853         lfg = lf * lfg;
00854         lg = lf * lg;
00855         lgg = lf * lgg;
00856
00857         lf = lf * lf;
00858         lff = lf * lff;
00859         lfg = lf * lfg;
00860         lg = lf * lg;
00861         lgg = lf * lgg;
00862
00863         lf = lf * lf;
00864         lff = lf * lff;
00865         lfg = lf * lfg;
00866         lg = lf * lg;
00867         lgg = lf * lgg;
00868
00869         lf = lf * lf;
00870         lff = lf * lff;
00871         lfg = lf * lfg;
00872         lg = lf * lg;
00873         lgg = lf * lgg;
00874
00875         lf = lf * lf;
00876         lff = lf * lff;
00877         lfg = lf * lfg;
00878         lg = lf * lg;
00879         lgg = lf * lgg;
00880
00881         lf = lf * lf;
00882         lff = lf * lff;
00883         lfg = lf * lfg;
00884         lg = lf * lg;
00885         lgg = lf * lgg;
00886
00887         lf = lf * lf;
00888         lff = lf * lff;
00889         lfg = lf * lfg;
00890         lg = lf * lg;
00891         lgg = lf * lgg;
00892
00893         lf = lf * lf;
00894         lff = lf * lff;
00895         lfg = lf * lfg;
00896         lg = lf * lg;
00897         lgg = lf * lgg;
00898
00899         lf = lf * lf;
00900         lff = lf * lff;
00901         lfg = lf * lfg;
00902         lg = lf * lg;
00903         lgg = lf * lgg;
00904
00905         lf = lf * lf;
00906         lff = lf * lff;
00907         lfg = lf * lfg;
00908         lg = lf * lg;
00909         lgg = lf * lgg;
00910
00911         lf = lf * lf;
00912         lff = lf * lff;
00913         lfg = lf * lfg;
00914         lg = lf * lg;
00915         lgg = lf * lgg;
00916
00917         lf = lf * lf;
00918         lff = lf * lff;
00919         lfg = lf * lfg;
00920         lg = lf * lg;
00921         lgg = lf * lgg;
00922
00923         lf = lf * lf;
00924         lff = lf * lff;
00925         lfg = lf * lfg;
00926         lg = lf * lg;
00927         lgg = lf * lgg;
00928
00929         lf = lf * lf;
00930         lff = lf * lff;
00931         lfg = lf * lfg;
00932         lg = lf * lg;
00933         lgg = lf * lgg;
00934
00935         lf = lf * lf;
00936         lff = lf * lff;
00937         lfg = lf * lfg;
00938         lg = lf * lg;
00939         lgg = lf * lgg;
00940
00941         lf = lf * lf;
00942         lff = lf * lff;
00943         lfg = lf * lfg;
00944         lg = lf * lg;
00945         lgg = lf * lgg;
00946
00947         lf = lf * lf;
00948         lff = lf * lff;
00949         lfg = lf * lfg;
00950         lg = lf * lg;
00951         lgg = lf * lgg;
00952
00953         lf = lf * lf;
00954         lff = lf * lff;
00955         lfg = lf * lfg;
00956         lg = lf * lg;
00957         lgg = lf * lgg;
00958
00959         lf = lf * lf;
00960         lff = lf * lff;
00961         lfg = lf * lfg;
00962         lg = lf * lg;
00963         lgg = lf * lgg;
00964
00965         lf = lf * lf;
00966         lff = lf * lff;
00967         lfg = lf * lfg;
00968         lg = lf * lg;
00969         lgg = lf * lgg;
00970
00971         lf = lf * lf;
00972         lff = lf * lff;
00973         lfg = lf * lfg;
00974         lg = lf * lg;
00975         lgg = lf * lgg;
00976
00977         lf = lf * lf;
00978         lff = lf * lff;
00979         lfg = lf * lfg;
00980         lg = lf * lg;
00981         lgg = lf * lgg;
00982
00983         lf = lf * lf;
00984         lff = lf * lff;
00985         lfg = lf * lfg;
00986         lg = lf * lg;
00987         lgg = lf * lgg;
00988
00989         lf = lf * lf;
00990         lff = lf * lff;
00991         lfg = lf * lfg;
00992         lg = lf * lg;
00993         lgg = lf * lgg;
00994
00995         lf = lf * lf;
00996         lff = lf * lff;
00997         lfg = lf * lfg;
00998         lg = lf * lg;
00999         lgg = lf * lgg;
01000
01001         lf = lf * lf;
01002         lff = lf * lff;
01003         lfg = lf * lfg;
01004         lg = lf * lg;
01005         lgg = lf * lgg;
01006
01007         lf = lf * lf;
01008         lff = lf * lff;
01009         lfg = lf * lfg;
01010         lg = lf * lg;
01011         lgg = lf * lgg;
01012
01013         lf = lf * lf;
01014         lff = lf * lff;
01015         lfg = lf * lfg;
01016         lg = lf * lg;
01017         lgg = lf * lgg;
01018
01019         lf = lf * lf;
01020         lff = lf * lff;
01021         lfg = lf * lfg;
01022         lg = lf * lg;
01023         lgg = lf * lgg;
01024
01025         lf = lf * lf;
01026         lff = lf * lff;
01027         lfg = lf * lfg;
01028         lg = lf * lg;
01029         lgg = lf * lgg;
01030
01031         lf = lf * lf;
01032         lff = lf * lff;
01033         lfg = lf * lfg;
01034         lg = lf * lg;
01035         lgg = lf * lgg;
01036
01037         lf = lf * lf;
01038         lff = lf * lff;
01039         lfg = lf * lfg;
01040         lg = lf * lg;
01041         lgg = lf * lgg;
01042
01043         lf = lf * lf;
01044         lff = lf * lff;
01045         lfg = lf * lfg;
01046         lg = lf * lg;
01047         lgg = lf * lgg;
01048
01049         lf = lf * lf;
01050         lff = lf * lff;
01051         lfg = lf * lfg;
01052         lg = lf * lg;
01053         lgg = lf * lgg;
01054
01055         lf = lf * lf;
01056         lff = lf * lff;
01057         lfg = lf * lfg;
01058         lg = lf * lg;
01059         lgg = lf * lgg;
01060
01061         lf = lf * lf;
01062         lff = lf * lff;
01063         lfg = lf * lfg;
01064         lg = lf * lg;
01065         lgg = lf * lgg;
01066
01067         lf = lf * lf;
01068         lff = lf * lff;
01069         lfg = lf * lfg;
01070         lg = lf * lg;
01071         lgg = lf * lgg;
01072
01073         lf = lf * lf;
01074         lff = lf * lff;
01075         lfg = lf * lfg;
01076         lg = lf * lg;
01077         lgg = lf * lgg;
01078
01079         lf = lf * lf;
01080         lff = lf * lff;
01081         lfg = lf * lfg;
01082         lg = lf * lg;
01083         lgg = lf * lgg;
01084
01085         lf = lf * lf;
01086         lff = lf * lff;
01087         lfg = lf * lfg;
01088         lg = lf * lg;
01089         lgg = lf * lgg;
01090
01091         lf = lf * lf;
01092         lff = lf * lff;
01093         lfg = lf * lfg;
01094         lg = lf * lg;
01095         lgg = lf * lgg;
01096
01097         lf = lf * lf;
01098         lff = lf * lff;
01099         lfg = lf * lfg;
01100         lg = lf * lg;
01101         lgg = lf * lgg;
01102
01103         lf = lf * lf;
01104         lff = lf * lff;
01105         lfg = lf * lfg;
01106         lg = lf * lg;
01107         lgg = lf * lgg;
01108
01109         lf = lf * lf;
01110         lff = lf * lff;
01111         lfg = lf * lfg;
01112         lg = lf * lg;
01113         lgg = lf * lgg;
01114
01115         lf = lf * lf;
01116         lff = lf * lff;
01117         lfg = lf * lfg;
01118         lg = lf * lg;
01119         lgg = lf * lgg;
01120
01121         lf = lf * lf;
01122         lff = lf * lff;
01123         lfg = lf * lfg;
01124         lg = lf * lg;
01125         lgg = lf * lgg;
01126
01127         lf = lf * lf;
01128         lff = lf * lff;
01129         lfg = lf * lfg;
01130         lg = lf * lg;
01131         lgg = lf * lgg;
01132
01133         lf = lf * lf;
01134         lff = lf * lff;
01135         lfg = lf * lfg;
01136         lg = lf * lg;
01137         lgg = lf * lgg;
01138
01139         lf = lf * lf;
01140         lff = lf * lff;
01141         lfg = lf * lfg;
01142         lg = lf * lg;
01143         lgg = lf * lgg;
01144
01145         lf = lf * lf;
01146         lff = lf * lff;
01147         lfg = lf * lfg;
01148         lg = lf * lg;
01149         lgg = lf * lgg;
01150
01151         lf = lf * lf;
01152         lff = lf * lff;
01153         lfg = lf * lfg;
01154         lg = lf * lg;
01155         lgg = lf * lgg;
01156
01157         lf = lf * lf;
01158         lff = lf * lff;
01159         lfg = lf * lfg;
01160         lg = lf * lg;
01161         lgg = lf * lgg;
01162
01163         lf = lf * lf;
01164         lff = lf * lff;
01165         lfg = lf * lfg;
01166         lg = lf * lg;
01167         lgg = lf * lgg;
01168
01169         lf = lf * lf;
01170         lff = lf * lff;
01171         lfg = lf * lfg;
01172         lg = lf * lg;
01173         lgg = lf * lgg;
01174
01175         lf = lf * lf;
01176         lff = lf * lff;
01177         lfg = lf * lfg;
01178         lg = lf * lg;
01179         lgg = lf * lgg;
01180
01181         lf = lf * lf;
01182         lff = lf * lff;
01183         lfg = lf * lfg;
01184         lg = lf * lg;
01185         lgg = lf * lgg;
01186
01187         lf = lf * lf;
01188         lff = lf * lff;
01189         lfg = lf * lfg;
01190         lg = lf * lg;
01191         lgg = lf * lgg;
01192
01193         lf = lf * lf;
01194         lff = lf * lff;
01195         lfg = lf * lfg;
01196         lg = lf * lg;
01197         lgg = lf * lgg;
01198
01199         lf = lf * lf;
01200         lff = lf * lff;
01201         lfg = lf * lfg;
01202         lg = lf * lg;
01203         lgg = lf * lgg;
01204
01205         lf = lf * lf;
01206         lff = lf * lff;
01207         lfg = lf * lfg;
01208         lg = lf * lg;
01209         lgg = lf * lgg;
01210
01211         lf = lf * lf;
01212         lff = lf * lff;
01213         lfg = lf * lfg;
01214         lg = lf * lg;
01215         lgg = lf * lgg;
01216
01217         lf = lf * lf;
01218         lff = lf * lff;
01219         lfg = lf * lfg;
01220         lg = lf * lg;
01221         lgg = lf * lgg;
01222
01223         lf = lf * lf;
01224         lff = lf * lff;
01225         lfg = lf * lfg;
01226         lg = lf * lg;
01227         lgg = lf * lgg;
01228
01229         lf = lf * lf;
01230         lff = lf * lff;
01231         lfg = lf * lfg;
01232         lg = lf * lg;
01233         lgg = lf * lgg;
01234
01235         lf = lf * lf;
01236         lff = lf * lff;
01237         lfg = lf * lfg;
01238         lg = lf * lg;
01239         lgg = lf * lgg;
01240
01241         lf = lf * lf;
01242         lff = lf * lff;
01243         lfg = lf * lfg;
01244         lg = lf * lg;
01245         lgg = lf * lgg;
01246
01247         lf = lf * lf;
01248         lff = lf * lff;
01249         lfg = lf * lfg;
01250         lg = lf * lg;
01251         lgg = lf * lgg;
01252
01253         lf = lf * lf;
01254         lff = lf * lff;
01255         lfg = lf * lfg;
01256         lg = lf * lg;
01257         lgg = lf * lgg;
01258
01259         lf = lf * lf;
01260         lff = lf * lff;
01261         lfg = lf * lfg;
01262         lg = lf * lg;
01263         lgg = lf * lgg;
01264
01265         lf = lf * lf;
01266         lff = lf * lff;
01267         lfg = lf * lfg;
01268         lg = lf * lg;
01269         lgg = lf * lgg;
01270
01271         lf = lf * lf;
01272         lff = lf * lff;
01273         lfg = lf * lfg;
01274         lg = lf * lg;
01275         lgg = lf * lgg;
01276
01277         lf = lf * lf;
01278         lff = lf * lff;
01279         lfg = lf * lfg;
01280         lg = lf * lg;
01281         lgg = lf * lgg;
01282
01283         lf = lf * lf;
01284         lff = lf * lff;
01285         lfg = lf * lfg;
01286         lg = lf * lg;
01287         lgg = lf * lgg;
01288
01289         lf = lf * lf;
01290         lff = lf * lff;
01291         lfg = lf * lfg;
01292         lg = lf * lg;
01293         lgg = lf * lgg;
01294
01295         lf = lf * lf;
01296         lff = lf * lff;
01297         lfg = lf * lfg;
01298         lg = lf * lg;
01299         lgg = lf * lgg;
01300
01301         lf = lf * lf;
01302         lff = lf * lff;
01303         lfg = lf * lfg;
01304         lg = lf * lg;
01305         lgg = lf * lgg;
01306
01307         lf = lf * lf;
01308         lff = lf * lff;
01309         lfg = lf * lfg;
01310         lg = lf * lg;
01311         lgg = lf * lgg;
01312
01313         lf = lf * lf;
01314         lff = lf * lff;
01315         lfg = lf * lfg;
01316         lg = lf * lg;
01317         lgg = lf * lgg;
01318
01319         lf = lf * lf;
01320         lff = lf * lff;
01321         lfg = lf * lfg;
01322         lg = lf * lg;
01323         lgg = lf * lgg;
01324
01325         lf = lf * lf;
01326         lff = lf * lff;
01327         lfg = lf * lfg;
01328         lg = lf * lg;
01329         lgg = lf * lgg;
01330
01331         lf = lf * lf;
01332         lff = lf * lff;
01333         lfg = lf * lfg;
01334         lg = lf * lg;
01335         lgg = lf * lgg;
01336
01337         lf = lf * lf;
01338         lff = lf * lff;
01339         lfg = lf * lfg;
01340         lg = lf * lg;
01341         lgg = lf * lgg;
01342
01343         lf = lf * lf;
01344         lff = lf * lff;
01345         lfg = lf * lfg;
01346         lg = lf * lg;
01347         lgg = lf * lgg;
01348
01349         lf = lf * lf;
01350         lff = lf * lff;
01351         lfg = lf * lfg;
01352         lg = lf * lg;
01353         lgg = lf * lgg;
01354
01355         lf = lf * lf;
01356         lff = lf * lff;
01357         lfg = lf * lfg;
01358         lg = lf * lg;
01359         lgg = lf * lgg;
01360
01361         lf = lf * lf;
01362         lff = lf * lff;
01363         lfg = lf * lfg;
01364         lg = lf * lg;
01365         lgg = lf * lgg;
01366
01367         lf = lf * lf;
01368         lff = lf * lff;
01369         lfg = lf * lfg;
01370         lg = lf * lg;
01371         lgg = lf * lgg;
01372
01373         lf = lf * lf;
01374         lff = lf * lff;
01375         lfg = lf * lfg;
01376         lg = lf * lg;
01377         lgg = lf * lgg;
01378
01379         lf = lf * lf;
01380         lff = lf * lff;
01381         lfg = lf * lfg;
01382         lg = lf * lg;
01383         lgg = lf * lgg;
01384
01385         lf = lf * lf;
01386         lff = lf * lff;
01387         lfg = lf * lfg;
01388         lg = lf * lg;
01389         lgg = lf * lgg;
01390
01391         lf = lf * lf;
01392         lff = lf * lff;
01393         lfg = lf * lfg;
01394         lg = lf * lg;
01395         lgg = lf * lgg;
01396
01397         lf = lf * lf;
01398         lff = lf * lff;
01399         lfg = lf * lfg;
01400         lg = lf * lg;
01401         lgg = lf * lgg;
01402
01403         lf = lf * lf;
01404         lff = lf * lff;
01405         lfg = lf * lfg;
01406         lg = lf * lg;
01407         lgg = lf * lgg;
01408
01409         lf = lf * lf;

```

```

00586             (Quad[2] = udata[pp + 2] * udata[pp + 2]) -
00587             (Quad[3] = udata[pp + 3] * udata[pp + 3]) -
00588             (Quad[4] = udata[pp + 4] * udata[pp + 4]) -
00589             (Quad[5] = udata[pp + 5] * udata[pp + 5]));
00590     g = udata[pp] * udata[pp + 3] + udata[pp + 1] * udata[pp + 4] +
00591         udata[pp + 2] * udata[pp + 5];
00592     switch ((*c) {
00593     case 0:
00594         lf = 0;
00595         lff = 0;
00596         lfg = 0;
00597         lg = 0;
00598         lgg = 0;
00599         break;
00600     case 1:
00601         lf = 0.000206527095658582755255648 * f;
00602         lff = 0.000206527095658582755255648;
00603         lfg = 0;
00604         lg = 0.0003614224174025198216973841 * g;
00605         lgg = 0.0003614224174025198216973841;
00606         break;
00607     case 2:
00608         lf = 0.000354046449700427580438254 * f * f +
00609             0.000191775160254398272737387 * g * g;
00610         lff = 0.0007080928994008551608765075 * f;
00611         lfg = 0.0003835503205087965454747749 * g;
00612         lg = 0.0003835503205087965454747749 * f * g;
00613         lgg = 0.0003835503205087965454747749 * f;
00614         break;
00615     case 3:
00616         lf = (0.000206527095658582755255648 + 0.000354046449700427580438254 * f) *
00617             f +
00618             0.000191775160254398272737387 * g * g;
00619         lff = 0.000206527095658582755255648 + 0.000708092899400855160876508 * f;
00620         lfg = 0.0003835503205087965454747749 * g;
00621         lg = (0.000361422417402519821697384 + 0.000383550320508796545474775 * f) *
00622             g;
00623         lgg = 0.000361422417402519821697384 + 0.000383550320508796545474775 * f;
00624         break;
00625     default:
00626         errorKill(
00627             "You need to specify a correct order in the weak-field expansion.");
00628     }
00629
00630     JMM[0] = lf + lff * Quad[0] +
00631         udata[3 + pp] * (2 * lfg * udata[pp] + lgg * udata[3 + pp]);
00632     JMM[1] =
00633         lff * udata[pp] * udata[1 + pp] + lfg * udata[1 + pp] * udata[3 + pp] +
00634         lfg * udata[pp] * udata[4 + pp] + lgg * udata[3 + pp] * udata[4 + pp];
00635     JMM[2] = lf + lff * Quad[1] +
00636         udata[4 + pp] * (2 * lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00637     JMM[3] =
00638         lff * udata[pp] * udata[2 + pp] + lfg * udata[2 + pp] * udata[3 + pp] +
00639         lfg * udata[pp] * udata[5 + pp] + lgg * udata[3 + pp] * udata[5 + pp];
00640     JMM[4] = lff * udata[1 + pp] * udata[2 + pp] +
00641             lfg * udata[2 + pp] * udata[4 + pp] +
00642             lfg * udata[1 + pp] * udata[5 + pp] +
00643             lgg * udata[4 + pp] * udata[5 + pp];
00644     JMM[5] = lf + lff * Quad[2] +
00645         udata[5 + pp] * (2 * lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00646     JMM[6] = lg + lfg * (Quad[0] - Quad[3 + 0]) +
00647             (-lff + lgg) * udata[pp] * udata[3 + pp];
00648     JMM[7] = -(udata[3 + pp] * (lff * udata[1 + pp] + lfg * udata[4 + pp])) +
00649             udata[pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]);
00650     JMM[8] = -(udata[3 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00651             udata[pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00652     JMM[9] = -lf + lgg * Quad[0] +
00653         udata[3 + pp] * (-2 * lfg * udata[pp] + lff * udata[3 + pp]);
00654     JMM[10] = udata[1 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00655             (lff * udata[pp] + lfg * udata[3 + pp]) * udata[4 + pp];
00656     JMM[11] = lg + lfg * (Quad[1] - Quad[4 + 0]) +
00657             (-lff + lgg) * udata[1 + pp] * udata[4 + pp];
00658     JMM[12] = -(udata[4 + pp] * (lff * udata[2 + pp] + lfg * udata[5 + pp])) +
00659             udata[1 + pp] * (lfg * udata[2 + pp] + lgg * udata[5 + pp]);
00660     JMM[13] = lgg * udata[pp] * udata[1 + pp] +
00661         lff * udata[3 + pp] * udata[4 + pp] -
00662         lfg * (udata[1 + pp] * udata[3 + pp] + udata[pp] * udata[4 + pp]);
00663     JMM[14] = -lf + lgg * Quad[1] +
00664         udata[4 + pp] * (-2 * lfg * udata[1 + pp] + lff * udata[4 + pp]);
00665     JMM[15] = udata[2 + pp] * (lfg * udata[pp] + lgg * udata[3 + pp]) -
00666             (lff * udata[pp] + lfg * udata[3 + pp]) * udata[5 + pp];
00667     JMM[16] = udata[2 + pp] * (lfg * udata[1 + pp] + lgg * udata[4 + pp]) -
00668             (lff * udata[1 + pp] + lfg * udata[4 + pp]) * udata[5 + pp];
00669     JMM[17] = lg + lfg * (Quad[2] - Quad[5 + 0]) +
00670             (-lff + lgg) * udata[2 + pp] * udata[5 + pp];
00671     JMM[18] = lgg * udata[pp] * udata[2 + pp] +
00672             lff * udata[3 + pp] * udata[5 + pp] -

```

```

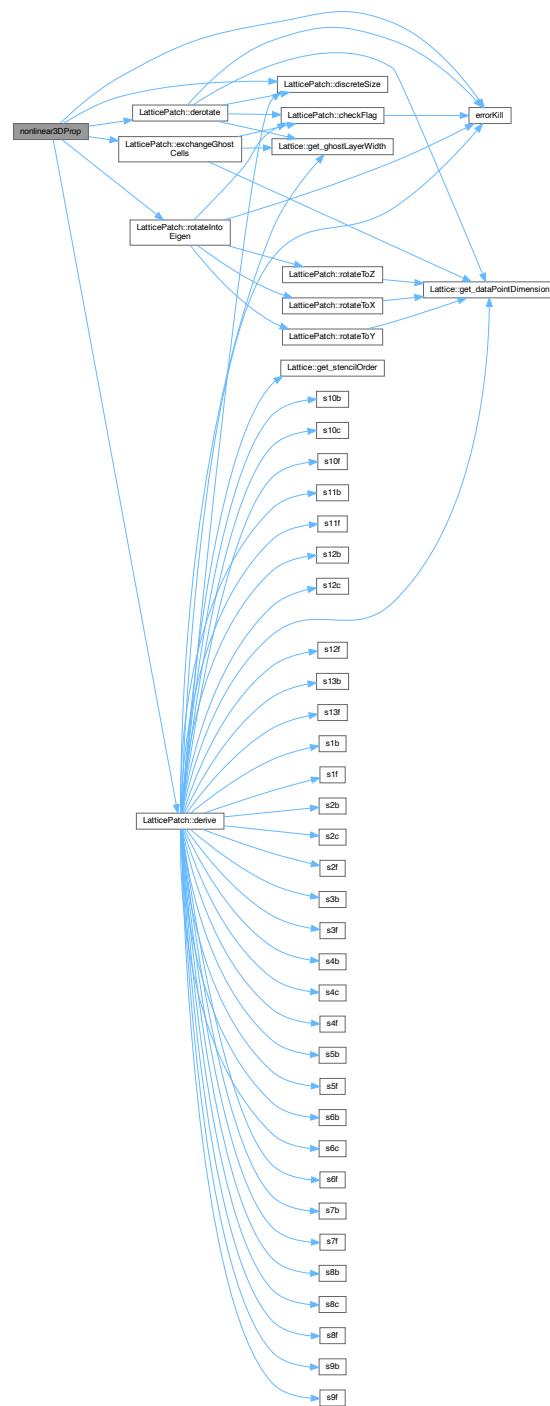
00673     lfg * (udata[2 + pp] * udata[3 + pp] + udata[pp] * udata[5 + pp]);
00674     JMM[19] =
00675         lgg * udata[1 + pp] * udata[2 + pp] +
00676         lff * udata[4 + pp] * udata[5 + pp] -
00677         lfg * (udata[2 + pp] * udata[4 + pp] + udata[1 + pp] * udata[5 + pp]);
00678     JMM[20] = -lf + lgg * Quad[2] +
00679         udata[5 + pp] * (-2 * lfg * udata[2 + pp] + lff * udata[5 + pp]);
00680
00681     h[0] = 0;
00682     h[1] = dxData[pp] * JMM[15] + dxData[1 + pp] * JMM[16] +
00683         dxData[2 + pp] * JMM[17] + dxData[3 + pp] * JMM[18] +
00684         dxData[4 + pp] * JMM[19] + dxData[5 + pp] * (-1 + JMM[20]);
00685     h[2] = -(dxData[pp] * JMM[10]) - dxData[1 + pp] * JMM[11] -
00686         dxData[2 + pp] * JMM[12] - dxData[3 + pp] * JMM[13] +
00687         dxData[4 + pp] * (1 - JMM[14]) - dxData[5 + pp] * JMM[19];
00688     h[3] = 0;
00689     h[4] = dxData[2 + pp];
00690     h[5] = -dxData[1 + pp];
00691     h[0] += -(dyData[pp] * JMM[15]) - dyData[1 + pp] * JMM[16] -
00692         dyData[2 + pp] * JMM[17] - dyData[3 + pp] * JMM[18] -
00693         dyData[4 + pp] * JMM[19] + dyData[5 + pp] * (1 - JMM[20]);
00694     h[1] += 0;
00695     h[2] += dyData[pp] * JMM[6] + dyData[1 + pp] * JMM[7] +
00696         dyData[2 + pp] * JMM[8] + dyData[3 + pp] * (-1 + JMM[9]) +
00697         dyData[4 + pp] * JMM[13] + dyData[5 + pp] * JMM[18];
00698     h[3] += -dyData[2 + pp];
00699     h[4] += 0;
00700     h[5] += dyData[pp];
00701     h[0] += dzData[pp] * JMM[10] + dzData[1 + pp] * JMM[11] +
00702         dzData[2 + pp] * JMM[12] + dzData[3 + pp] * JMM[13] +
00703         dzData[4 + pp] * (-1 + JMM[14]) + dzData[5 + pp] * JMM[19];
00704     h[1] += -(dzData[pp] * JMM[6]) - dzData[1 + pp] * JMM[7] -
00705         dzData[2 + pp] * JMM[8] + dzData[3 + pp] * (1 - JMM[9]) -
00706         dzData[4 + pp] * JMM[13] - dzData[5 + pp] * JMM[18];
00707     h[2] += 0;
00708     h[3] += dzData[1 + pp];
00709     h[4] += -dzData[pp];
00710     h[5] += 0;
00711     h[0] -= h[3] * JMM[6] + h[4] * JMM[10] + h[5] * JMM[15];
00712     h[1] -= h[3] * JMM[7] + h[4] * JMM[11] + h[5] * JMM[16];
00713     h[2] -= h[3] * JMM[8] + h[4] * JMM[12] + h[5] * JMM[17];
00714     dudata[pp + 0] =
00715         h[2] * (-JMM[3] * (1 + JMM[2])) + JMM[1] * JMM[4]) +
00716         h[1] * (JMM[3] * JMM[4] - JMM[1] * (1 + JMM[5])) +
00717         h[0] * (1 - JMM[4] * JMM[4] + JMM[5] + JMM[2] * (1 + JMM[5]));
00718     dudata[pp + 1] =
00719         h[2] * (JMM[3] * JMM[1] - (1 + JMM[0]) * JMM[4]) +
00720         h[1] * (1 - JMM[3] * JMM[3] + JMM[5] + JMM[0] * (1 + JMM[5])) +
00721         h[0] * (JMM[4] * JMM[3] - JMM[1] * (1 + JMM[5]));
00722     dudata[pp + 2] =
00723         h[2] * (1 - JMM[1] * JMM[1] + JMM[2] + JMM[0] * (1 + JMM[2])) +
00724         h[1] * (JMM[1] * JMM[3] - (1 + JMM[0]) * JMM[4]) +
00725         h[0] * (-(1 + JMM[2]) * JMM[3] + JMM[1] * JMM[4]);
00726     detC =
00727         -(1 + JMM[2]) * (-1 + JMM[3] * JMM[3])) +
00728         (JMM[3] * JMM[1] - JMM[4]) * JMM[4] + JMM[5] + JMM[2] * JMM[5] +
00729         JMM[0] * (1 + JMM[2] - JMM[4] * JMM[4] + (1 + JMM[2]) * JMM[5]) -
00730         JMM[1] * (-(JMM[4] * JMM[3]) + JMM[1] * (1 + JMM[5]));
00731     dudata[pp + 0] /= detC;
00732     dudata[pp + 1] /= detC;
00733     dudata[pp + 2] /= detC;
00734     dudata[pp + 3] = h[3];
00735     dudata[pp + 4] = h[4];
00736     dudata[pp + 5] = h[5];
00737 }
00738 return;
00739 }

```

References [LatticePatch::buffData](#), [LatticePatch::derive\(\)](#), [LatticePatch::derotate\(\)](#), [LatticePatch::discreteSize\(\)](#), [errorKill\(\)](#), [LatticePatch::exchangeGhostCells\(\)](#), and [LatticePatch::rotateIntoEigen\(\)](#).

Referenced by [Sim3D\(\)](#).

Here is the call graph for this function:



Here is the caller graph for this function:



6.61 TimeEvolutionFunctions.h

[Go to the documentation of this file.](#)

```

00001 ///////////////////////////////////////////////////////////////////
00002 /// @file TimeEvolutionFunctions.h
00003 /// @brief Functions to propagate data vectors in time
00004 /// according to Maxwell's equations, and various
00005 /// orders in the HE weak-field expansion
00006 ///////////////////////////////////////////////////////////////////
00007
00008 #pragma once
00009
00010 #include "LatticePatch.h"
00011 #include "SimulationClass.h"
00012
00013 /** @brief monostate TimeEvolution class to propagate the field data in time in
00014 * a given order of the HE weak-field expansion */
00015 class TimeEvolution {
00016 public:
00017     /// choice which processes of the weak field expansion are included
00018     static int *c;
00019
00020     /// Pointer to functions for differentiation and time evolution
00021     static void (*TimeEvolver)(LatticePatch *, N_Vector, N_Vector, int *);
00022
00023     /// CVODE right hand side function (CVRhsFn) to provide IVP of the ODE
00024     static int f(sunrealtype t, N_Vector u, N_Vector udot, void *data_loc);
00025 };
00026
00027 /// Maxwell propagation function for 1D -- only for reference
00028 void linear1DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c);
00029 /// HE propagation function for 1D
00030 void nonlinear1DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c);
00031 /// Maxwell propagation function for 2D -- only for reference
00032 void linear2DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c);
00033 /// HE propagation function for 2D
00034 void nonlinear2DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c);
00035 /// Maxwell propagation function for 3D -- only for reference
00036 void linear3DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c);
00037 /// HE propagation function for 3D
00038 void nonlinear3DProp(LatticePatch *data, N_Vector u, N_Vector udot, int *c);
00039

```

Index

__has_include
 CMakeCCompilerId.c, 141
 CMakeCXXCompilerId.cpp, 156

~LatticePatch
 LatticePatch, 62

~Simulation
 Simulation, 120

A1
 Gauss2D, 20
 Gauss3D, 25

A2
 Gauss2D, 20
 Gauss3D, 26

add
 ICSetter, 35

addGauss1D
 ICSetter, 36

addGauss2D
 ICSetter, 37

addGauss3D
 ICSetter, 37

addInitialConditions
 Simulation, 121

addPeriodicCLayerInX
 Simulation, 122

addPeriodicCLayerInXY
 Simulation, 122

addPlaneWave1D
 ICSetter, 38

addPlaneWave2D
 ICSetter, 39

addPlaneWave3D
 ICSetter, 39

addToSpace
 Gauss1D, 13
 Gauss2D, 19
 Gauss3D, 25
 PlaneWave1D, 111
 PlaneWave2D, 114
 PlaneWave3D, 117

advanceToTime
 Simulation, 123

Amp
 Gauss2D, 20
 Gauss3D, 26

amp
 gaussian2D, 30
 gaussian3D, 32

ARCHITECTURE_ID

CMakeCCompilerId.c
 __has_include, 141
 ARCHITECTURE_ID, 142
 C_VERSION, 142
 COMPILER_ID, 142
 DEC, 142
 HEX, 142
 info_arch, 144
 info_compiler, 144
 info_language_extensions_default, 144
 info_language_standard_default, 145
 info_platform, 145
 main, 143
 PLATFORM_ID, 143
 STRINGIFY, 143
 STRINGIFY_HELPER, 143

CMakeCXXCompilerId.cpp
 __has_include, 156
 ARCHITECTURE_ID, 156
 COMPILER_ID, 156
 CXX_STD, 156
 DEC, 156
 HEX, 156
 info_arch, 158
 info_compiler, 158
 info_language_extensions_default, 158
 info_language_standard_default, 159
 info_platform, 159
 main, 157
 PLATFORM_ID, 157
 STRINGIFY, 157
 STRINGIFY_HELPER, 157

comm
 Lattice, 53

COMPILER_ID
 CMakeCCompilerId.c, 142
 CMakeCXXCompilerId.cpp, 156

cvoce_mem
 Simulation, 135

CvodeObjectSetUp
 SimulationClass.h, 320

CXX_STD
 CMakeCXXCompilerId.cpp, 156

dataPointDimension
 Lattice, 53

DEC
 CMakeCCompilerId.c, 142
 CMakeCXXCompilerId.cpp, 156

DerivationStencils.h
 s10b, 231
 s10c, 232, 233
 s10f, 233, 234
 s11b, 234, 235
 s11f, 236
 s12b, 237
 s12c, 238, 239
 s12f, 239, 240
 s13b, 240, 241
 s13f, 242
 s1b, 243
 s1f, 244
 s2b, 245
 s2c, 246
 s2f, 247
 s3b, 248
 s3f, 249
 s4b, 250
 s4c, 251
 s4f, 252
 s5b, 253
 s5f, 254
 s6b, 255
 s6c, 256
 s6f, 257

s7b, 258
 s7f, 259
 s8b, 260
 s8c, 261
 s8f, 262
 s9b, 263
 s9f, 264

derive
 LatticePatch, 64

derotate
 LatticePatch, 69

dis
 Gauss2D, 21
 Gauss3D, 26

discreteSize
 LatticePatch, 71

du
 LatticePatch, 86

duData
 LatticePatch, 86

duLocal
 LatticePatch, 86

dx
 Lattice, 53
 LatticePatch, 87

dy
 Lattice, 54
 LatticePatch, 87

dz
 Lattice, 54
 LatticePatch, 87

envelopeLattice
 LatticePatch, 87

errorKill
 LatticePatch.cpp, 281
 LatticePatch.h, 298

eval
 ICSsetter, 40

exchangeGhostCells
 LatticePatch, 72

f
 TimeEvolution, 138

FLatticeDimensionSet
 LatticePatch.h, 300

FLatticePatchSetUp
 LatticePatch.h, 300

Gauss1D, 11
 addToSpace, 13
 Gauss1D, 12
 kx, 14
 ky, 14
 kz, 14
 phig, 14
 phix, 14
 phiy, 15
 phiz, 15

px, 15
py, 15
pz, 16
x0x, 16
x0y, 16
x0z, 16
gauss1Ds
 ICSetter, 41
Gauss2D, 17
 A1, 20
 A2, 20
 addToSpace, 19
 Amp, 20
 axis, 20
 dis, 21
 Gauss2D, 18
 lambda, 21
 Ph0, 21
 PhA, 21
 phip, 22
 w0, 22
 zr, 22
gauss2Ds
 ICSetter, 41
Gauss3D, 23
 A1, 25
 A2, 26
 addToSpace, 25
 Amp, 26
 axis, 26
 dis, 26
 Gauss3D, 24
 lambda, 27
 Ph0, 27
 PhA, 27
 phip, 27
 w0, 28
 zr, 28
gauss3Ds
 ICSetter, 41
gaussian1D, 28
 k, 29
 p, 29
 phi, 29
 phig, 29
 x0, 29
gaussian2D, 30
 amp, 30
 axis, 30
 ph0, 31
 phA, 31
 phip, 31
 w0, 31
 x0, 31
 zr, 32
gaussian3D, 32
 amp, 32
 axis, 33
 ph0, 33
 phA, 33
 phip, 33
 w0, 33
 x0, 34
 zr, 34
gCLData
 LatticePatch, 88
gCRData
 LatticePatch, 88
generateOutputFolder
 OutputManager, 99
generatePatchwork
 LatticePatch, 84
 LatticePatch.cpp, 282
generateTranslocationLookup
 LatticePatch, 74
get_cart_comm
 Simulation, 127
get_dataPointDimension
 Lattice, 45
get_dx
 Lattice, 45
get_dy
 Lattice, 46
get_dz
 Lattice, 46
get_ghostLayerWidth
 Lattice, 46
getStencilOrder
 Lattice, 47
get_tot_lx
 Lattice, 48
get_tot_ly
 Lattice, 48
get_tot_lz
 Lattice, 49
get_tot_noDP
 Lattice, 49
get_tot_noP
 Lattice, 50
get_tot_nx
 Lattice, 50
get_tot_ny
 Lattice, 50
get_tot_nz
 Lattice, 50
getDelta
 LatticePatch, 76
getSimCode
 OutputManager, 100
ghostCellLeft
 LatticePatch, 88
ghostCellLeftToSend
 LatticePatch, 88
ghostCellRight
 LatticePatch, 89
ghostCellRightToSend

LatticePatch, 89
 ghostCells
 LatticePatch, 89
 ghostCellsToSend
 LatticePatch, 89
 GhostLayersInitialized
 LatticePatch.h, 300
 ghostLayerWidth
 Lattice, 54

 HEX
 CMakeCCompilerId.c, 142
 CMakeCXXCompilerId.cpp, 156

 ICSetter, 34
 add, 35
 addGauss1D, 36
 addGauss2D, 37
 addGauss3D, 37
 addPlaneWave1D, 38
 addPlaneWave2D, 39
 addPlaneWave3D, 39
 eval, 40
 gauss1Ds, 41
 gauss2Ds, 41
 gauss3Ds, 41
 planeWaves1D, 42
 planeWaves2D, 42
 planeWaves3D, 42

 icsettings
 Simulation, 135

 ID
 LatticePatch, 89

 info_arch
 CMakeCCompilerId.c, 144
 CMakeCXXCompilerId.cpp, 158

 info_compiler
 CMakeCCompilerId.c, 144
 CMakeCXXCompilerId.cpp, 158

 info_language_extensions_default
 CMakeCCompilerId.c, 144
 CMakeCXXCompilerId.cpp, 158

 info_language_standard_default
 CMakeCCompilerId.c, 145
 CMakeCXXCompilerId.cpp, 159

 info_platform
 CMakeCCompilerId.c, 145
 CMakeCXXCompilerId.cpp, 159

 initializeBuffers
 LatticePatch, 77

 initializeCommunicator
 Lattice, 51

 initializeCVODEobject
 Simulation, 127

 initializePatchwork
 Simulation, 129

 k
 gaussian1D, 29

 planewave, 109

 kx
 Gauss1D, 14
 PlaneWave, 106

 ky
 Gauss1D, 14
 PlaneWave, 106

 kz
 Gauss1D, 14
 PlaneWave, 106

 lambda
 Gauss2D, 21
 Gauss3D, 27

 Lattice, 43
 comm, 53
 dataPointDimension, 53
 dx, 53
 dy, 54
 dz, 54
 get_dataPointDimension, 45
 get_dx, 45
 get_dy, 46
 get_dz, 46
 get_ghostLayerWidth, 46
 get_stencilOrder, 47
 get_tot_lx, 48
 get_tot_ly, 48
 get_tot_lz, 49
 get_tot_noDP, 49
 get_tot_noP, 50
 get_tot_nx, 50
 get_tot_ny, 50
 get_tot_nz, 50
 ghostLayerWidth, 54
 initializeCommunicator, 51
 Lattice, 44
 my_prc, 54
 n_prc, 55
 setDiscreteDimensions, 51
 setPhysicalDimensions, 52
 statusFlags, 55
 stencilOrder, 55
 sunctx, 55
 tot_lx, 56
 tot_ly, 56
 tot_lz, 56
 tot_noDP, 56
 tot_noP, 57
 tot_nx, 57
 tot_ny, 57
 tot_nz, 57

 lattice
 Simulation, 136

 LatticeDiscreteSetUp
 SimulationClass.h, 321

 LatticePatch, 58
 ~LatticePatch, 62
 buffData, 85

buffX, 85
buffY, 85
buffZ, 86
checkFlag, 62
derive, 64
derotate, 69
discreteSize, 71
du, 86
duData, 86
duLocal, 86
dx, 87
dy, 87
dz, 87
envelopeLattice, 87
exchangeGhostCells, 72
gCLData, 88
gCRData, 88
generatePatchwork, 84
generateTranslocationLookup, 74
getDelta, 76
ghostCellLeft, 88
ghostCellLeftToSend, 88
ghostCellRight, 89
ghostCellRightToSend, 89
ghostCells, 89
ghostCellsToSend, 89
ID, 89
initializeBuffers, 77
LatticePatch, 61
lgcTox, 90
lgcToy, 90
lgcToz, 90
lx, 90
ly, 91
lz, 91
ix, 91
iy, 91
iz, 92
nx, 92
ny, 92
nz, 92
origin, 78
rgcTox, 93
rgcToy, 93
rgcToz, 93
rotateIntoEigen, 79
rotateToX, 80
rotateToY, 81
rotateToZ, 82
statusFlags, 93
u, 94
uAux, 94
uAuxData, 94
uData, 94
uLocal, 95
uTox, 95
uToy, 95
uToz, 95
x0, 96
xTou, 96
y0, 96
yTou, 96
z0, 97
zTou, 97
latticePatch
 Simulation, 136
LatticePatch.cpp
 check_error, 279
 check_retval, 280
 errorKill, 281
 generatePatchwork, 282
LatticePatch.h
 BuffersInitialized, 300
 check_error, 297
 check_retval, 297
 errorKill, 298
 FLatticeDimensionSet, 300
 FLatticePatchSetUp, 300
 GhostLayersInitialized, 300
 TranslocationLookupSetUp, 300
LatticePatchworkSetUp
 SimulationClass.h, 321
LatticePhysicalSetUp
 SimulationClass.h, 321
lgcTox
 LatticePatch, 90
lgcToy
 LatticePatch, 90
lgcToz
 LatticePatch, 90
linear1DProp
 TimeEvolutionFunctions.cpp, 347
 TimeEvolutionFunctions.h, 376
linear2DProp
 TimeEvolutionFunctions.cpp, 349
 TimeEvolutionFunctions.h, 377
linear3DProp
 TimeEvolutionFunctions.cpp, 351
 TimeEvolutionFunctions.h, 379
lx
 LatticePatch, 90
ly
 LatticePatch, 91
lz
 LatticePatch, 91
ix
 LatticePatch, 91
iy
 LatticePatch, 91
iz
 LatticePatch, 92
main
 CMakeCCompilerId.c, 143
 CMakeCXXCompilerId.cpp, 157
 main.cpp, 304
 OpenMPCheckVersion.c, 171

OpenMPCheckVersion.cpp, 173
 OpenMPTryFlag.c, 174
 OpenMPTryFlag.cpp, 176
 test_mpi.cpp, 169
 main.cpp
 main, 304
 my_prc
 Lattice, 54
 n_prc
 Lattice, 55
 NLS
 Simulation, 136
 nonlinear1DProp
 TimeEvolutionFunctions.cpp, 353
 TimeEvolutionFunctions.h, 381
 nonlinear2DProp
 TimeEvolutionFunctions.cpp, 358
 TimeEvolutionFunctions.h, 386
 nonlinear3DProp
 TimeEvolutionFunctions.cpp, 362
 TimeEvolutionFunctions.h, 390
 nx
 LatticePatch, 92
 ny
 LatticePatch, 92
 nz
 LatticePatch, 92
 ompver_str
 OpenMPCheckVersion.c, 171
 OpenMPCheckVersion.cpp, 173
 OpenMPCheckVersion.c
 main, 171
 ompver_str, 171
 OpenMPCheckVersion.cpp
 main, 173
 ompver_str, 173
 OpenMPTryFlag.c
 main, 174
 OpenMPTryFlag.cpp
 main, 176
 origin
 LatticePatch, 78
 outAllFieldData
 Simulation, 130
 OutputManager, 97
 generateOutputFolder, 99
 getSimCode, 100
 OutputManager, 98
 outputStyle, 104
 outUState, 101
 Path, 104
 set_outputStyle, 102
 simCode, 104
 SimCodeGenerator, 103
 outputManager
 Simulation, 136
 outputStyle
 OutputManager, 104
 outUState
 OutputManager, 101
 p
 gaussian1D, 29
 planewave, 109
 Path
 OutputManager, 104
 Ph0
 Gauss2D, 21
 Gauss3D, 27
 ph0
 gaussian2D, 31
 gaussian3D, 33
 PhA
 Gauss2D, 21
 Gauss3D, 27
 phA
 gaussian2D, 31
 gaussian3D, 33
 phi
 gaussian1D, 29
 planewave, 109
 phig
 Gauss1D, 14
 gaussian1D, 29
 phip
 Gauss2D, 22
 Gauss3D, 27
 gaussian2D, 31
 gaussian3D, 33
 phix
 Gauss1D, 14
 PlaneWave, 106
 phiy
 Gauss1D, 15
 PlaneWave, 107
 phiz
 Gauss1D, 15
 PlaneWave, 107
 PlaneWave, 105
 kx, 106
 ky, 106
 kz, 106
 phix, 106
 phiy, 107
 phiz, 107
 px, 107
 py, 107
 pz, 108
 planewave, 108
 k, 109
 p, 109
 phi, 109
 PlaneWave1D, 110
 addToSpace, 111
 PlaneWave1D, 111
 PlaneWave2D, 112

addToSpace, 114
PlaneWave2D, 113
PlaneWave3D, 115
addToSpace, 117
PlaneWave3D, 116
planeWaves1D
ICSetter, 42
planeWaves2D
ICSetter, 42
planeWaves3D
ICSetter, 42
PLATFORM_ID
CMakeCCompilerId.c, 143
CMakeCXXCompilerId.cpp, 157
px
Gauss1D, 15
PlaneWave, 107
py
Gauss1D, 15
PlaneWave, 107
pz
Gauss1D, 16
PlaneWave, 108

README.md, 141
rgcTox
LatticePatch, 93
rgcToy
LatticePatch, 93
rgcToz
LatticePatch, 93
rotateIntoEigen
LatticePatch, 79
rotateToX
LatticePatch, 80
rotateToY
LatticePatch, 81
rotateToZ
LatticePatch, 82

s10b
DerivationStencils.h, 231
s10c
DerivationStencils.h, 232, 233
s10f
DerivationStencils.h, 233, 234
s11b
DerivationStencils.h, 234, 235
s11f
DerivationStencils.h, 236
s12b
DerivationStencils.h, 237
s12c
DerivationStencils.h, 238, 239
s12f
DerivationStencils.h, 239, 240
s13b
DerivationStencils.h, 240, 241
s13f
DerivationStencils.h, 242
s1b
DerivationStencils.h, 243
s1f
DerivationStencils.h, 244
s2b
DerivationStencils.h, 245
s2c
DerivationStencils.h, 246
s2f
DerivationStencils.h, 247
s3b
DerivationStencils.h, 248
s3f
DerivationStencils.h, 249
s4b
DerivationStencils.h, 250
s4c
DerivationStencils.h, 251
s4f
DerivationStencils.h, 252
s5b
DerivationStencils.h, 253
s5f
DerivationStencils.h, 254
s6b
DerivationStencils.h, 255
s6c
DerivationStencils.h, 256
s6f
DerivationStencils.h, 257
s7b
DerivationStencils.h, 258
s7f
DerivationStencils.h, 259
s8b
DerivationStencils.h, 260
s8c
DerivationStencils.h, 261
s8f
DerivationStencils.h, 262
s9b
DerivationStencils.h, 263
s9f
DerivationStencils.h, 264
set_outputStyle
OutputManager, 102
setDiscreteDimensions
Lattice, 51
setDiscreteDimensionsOfLattice
Simulation, 131
setInitialConditions
Simulation, 132
setPhysicalDimensions
Lattice, 52
setPhysicalDimensionsOfLattice
Simulation, 133
Sim1D

SimulationFunctions.cpp, 324
 SimulationFunctions.h, 337
Sim2D
 SimulationFunctions.cpp, 326
 SimulationFunctions.h, 339
Sim3D
 SimulationFunctions.cpp, 329
 SimulationFunctions.h, 342
simCode
 OutputManager, 104
SimCodeGenerator
 OutputManager, 103
Simulation, 117
 ~Simulation, 120
 addInitialConditions, 121
 addPeriodicCLayerInX, 122
 addPeriodicCLayerInXY, 122
 advanceToTime, 123
 checkFlag, 124
 checkNoFlag, 126
 cvode_mem, 135
 get_cart_comm, 127
 icsettings, 135
 initializeCVODEobject, 127
 initializePatchwork, 129
 lattice, 136
 latticePatch, 136
 NLS, 136
 outAllFieldData, 130
 outputManager, 136
 setDiscreteDimensionsOfLattice, 131
 setInitialConditions, 132
 setPhysicalDimensionsOfLattice, 133
 Simulation, 119
 start, 134
 statusFlags, 137
 t, 137
SimulationClass.h
 CvodeObjectSetUp, 320
 LatticeDiscreteSetUp, 321
 LatticePatchworkSetUp, 321
 LatticePhysicalSetUp, 321
 SimulationStarted, 321
SimulationFunctions.cpp
 Sim1D, 324
 Sim2D, 326
 Sim3D, 329
 timer, 331
SimulationFunctions.h
 Sim1D, 337
 Sim2D, 339
 Sim3D, 342
 timer, 344
SimulationStarted
 SimulationClass.h, 321
src/build/CMakeFiles/3.24.2/CompilerIdC/CMakeCCompilerId.c, 141, 145
 src/build/CMakeFiles/3.24.2/CompilerIdCXX/CMakeCXXCompilerId.cpp, 155, 159
src/build/CMakeFiles/FindMPI/test_mpi.cpp, 169, 170
src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.c, 170, 172
src/build/CMakeFiles/FindOpenMP/OpenMPCheckVersion.cpp, 172, 174
src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.c, 174, 175
src/build/CMakeFiles/FindOpenMP/OpenMPTryFlag.cpp, 175, 176
src/build/CMakeFiles/hewes.dir/DerivationStencils.cpp.o.d, 176
src/build/CMakeFiles/hewes.dir/ICSetters.cpp.o.d, 177
src/build/CMakeFiles/hewes.dir/LatticePatch.cpp.o.d, 183
src/build/CMakeFiles/hewes.dir/main.cpp.o.d, 191
src/build/CMakeFiles/hewes.dir/Outputters.cpp.o.d, 198
src/build/CMakeFiles/hewes.dir/SimulationClass.cpp.o.d, 205
src/build/CMakeFiles/hewes.dir/SimulationFunctions.cpp.o.d, 213
src/build/CMakeFiles/hewes.dir/TimeEvolutionFunctions.cpp.o.d, 220
src/DerivationStencils.cpp, 228, 229
src/DerivationStencils.h, 229, 265
src/ICSetters.cpp, 269
src/ICSetters.h, 274, 275
src/LatticePatch.cpp, 279, 284
src/LatticePatch.h, 295, 301
src/main.cpp, 304, 308
src/Outputters.cpp, 311, 312
src/Outputters.h, 313, 315
src/SimulationClass.cpp, 315, 316
src/SimulationClass.h, 319, 322
src/SimulationFunctions.cpp, 323, 332
src/SimulationFunctions.h, 335, 345
src/TimeEvolutionFunctions.cpp, 346, 366
src/TimeEvolutionFunctions.h, 374, 394
start
 Simulation, 134
statusFlags
 Lattice, 55
 LatticePatch, 93
 Simulation, 137
stencilOrder
 Lattice, 55
STRINGIFY
 CMakeCCompilerId.c, 143
 CMakeCXXCompilerId.cpp, 157
STRINGIFY_HELPER
 CMakeCCompilerId.c, 143
 CMakeCXXCompilerId.cpp, 157
sunctx
 Lattice, 55
test
 Simulation, 137
test_mpi.cpp

main, 169
TimeEvolution, 137
 c, 139
 f, 138
 TimeEvolver, 139
TimeEvolutionFunctions.cpp
 linear1DProp, 347
 linear2DProp, 349
 linear3DProp, 351
 nonlinear1DProp, 353
 nonlinear2DProp, 358
 nonlinear3DProp, 362
TimeEvolutionFunctions.h
 linear1DProp, 376
 linear2DProp, 377
 linear3DProp, 379
 nonlinear1DProp, 381
 nonlinear2DProp, 386
 nonlinear3DProp, 390
TimeEvolver
 TimeEvolution, 139
timer
 SimulationFunctions.cpp, 331
 SimulationFunctions.h, 344
tot_lx
 Lattice, 56
tot_ly
 Lattice, 56
tot_lz
 Lattice, 56
tot_noDP
 Lattice, 56
tot_noP
 Lattice, 57
tot_nx
 Lattice, 57
tot_ny
 Lattice, 57
tot_nz
 Lattice, 57
TranslocationLookupSetUp
 LatticePatch.h, 300

u
 LatticePatch, 94
uAux
 LatticePatch, 94
uAuxData
 LatticePatch, 94
uData
 LatticePatch, 94
uLocal
 LatticePatch, 95
uTox
 LatticePatch, 95
uToy
 LatticePatch, 95
uToz
 LatticePatch, 95
w0
 Gauss2D, 22
 Gauss3D, 28
 gaussian2D, 31
 gaussian3D, 33

x0
 gaussian1D, 29
 gaussian2D, 31
 gaussian3D, 34
 LatticePatch, 96
x0x
 Gauss1D, 16
x0y
 Gauss1D, 16
x0z
 Gauss1D, 16
xTou
 LatticePatch, 96

y0
 LatticePatch, 96
yTou
 LatticePatch, 96

z0
 LatticePatch, 97
zr
 Gauss2D, 22
 Gauss3D, 28
 gaussian2D, 32
 gaussian3D, 34
zTou
 LatticePatch, 97