

AZTEKAS: a hydrodynamic GPL code

Version1.0

Generated by Doxygen 1.8.13



# Contents

<b>1</b>	<b>AZTEKAS: a hydrodynamic GPL code</b>	<b>1</b>
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Class Documentation</b>	<b>7</b>
4.1	der_gauge_ Struct Reference . . . . .	7
4.1.1	Member Data Documentation . . . . .	7
4.1.1.1	dbeta . . . . .	7
4.1.1.2	dgam . . . . .	7
4.1.1.3	dlapse . . . . .	7
4.2	eos_ Struct Reference . . . . .	8
4.2.1	Member Data Documentation . . . . .	8
4.2.1.1	cs . . . . .	8
4.2.1.2	e . . . . .	8
4.2.1.3	h . . . . .	8
4.3	flx_ Struct Reference . . . . .	8
4.3.1	Member Data Documentation . . . . .	9
4.3.1.1	fm . . . . .	9
4.3.1.2	fp . . . . .	9
4.3.1.3	lm . . . . .	9
4.3.1.4	lp . . . . .	9

4.3.1.5	qm	9
4.3.1.6	qp	9
4.3.1.7	um	9
4.3.1.8	up	10
4.4	gauge_ Struct Reference	10
4.4.1	Member Data Documentation	10
4.4.1.1	beta_con	10
4.4.1.2	dety	10
4.4.1.3	gamma_con	10
4.4.1.4	lapse	11
4.4.1.5	x	11
4.5	grid_ Struct Reference	11
4.5.1	Member Data Documentation	11
4.5.1.1	S1m	11
4.5.1.2	S1p	12
4.5.1.3	S2m	12
4.5.1.4	S2p	12
4.5.1.5	S3m	12
4.5.1.6	S3p	12
4.5.1.7	time	12
4.5.1.8	X1	12
4.5.1.9	X1m	12
4.5.1.10	X1p	13
4.5.1.11	X2	13
4.5.1.12	X2m	13
4.5.1.13	X2p	13
4.5.1.14	X3	13
4.5.1.15	X3m	13
4.5.1.16	X3p	13
4.6	lim_ Struct Reference	14

4.6.1	Detailed Description . . . . .	14
4.6.2	Member Data Documentation . . . . .	14
4.6.2.1	sx1 . . . . .	14
4.6.2.2	sx2 . . . . .	14
4.6.2.3	sx3 . . . . .	14
4.6.2.4	ux . . . . .	15
4.6.2.5	ux1m . . . . .	15
4.6.2.6	ux1p . . . . .	15
4.6.2.7	ux2m . . . . .	15
4.6.2.8	ux2p . . . . .	15
4.6.2.9	ux3m . . . . .	15
4.6.2.10	ux3p . . . . .	15
4.7	vec_ Struct Reference . . . . .	16
4.7.1	Member Data Documentation . . . . .	16
4.7.1.1	A . . . . .	16
4.7.1.2	Fm . . . . .	16
4.7.1.3	Fp . . . . .	16
4.7.1.4	Gm . . . . .	16
4.7.1.5	Gp . . . . .	17
4.7.1.6	Hm . . . . .	17
4.7.1.7	Hp . . . . .	17
4.7.1.8	S . . . . .	17

<b>5</b>	<b>File Documentation</b>	<b>19</b>
5.1	.alloc.c File Reference	19
5.1.1	Function Documentation	19
5.1.1.1	Allocate_Array()	19
5.1.1.2	New_Size()	19
5.2	.initial.c File Reference	19
5.2.1	Function Documentation	20
5.2.1.1	INITIAL()	20
5.2.1.2	RESTART()	20
5.3	.initial_cond.c File Reference	20
5.3.1	Function Documentation	20
5.3.1.1	JET()	20
5.3.1.2	KH()	21
5.3.1.3	RIEMANN()	21
5.3.1.4	SPH_ACC()	21
5.3.1.5	WIND()	21
5.4	.limiters.c File Reference	21
5.4.1	Function Documentation	21
5.4.1.1	Godunov()	22
5.4.1.2	Limiter()	22
5.4.1.3	Maxmod()	22
5.4.1.4	Mc()	22
5.4.1.5	Minmod()	22
5.4.1.6	Reconst1D()	22
5.4.1.7	Reconst2D()	23
5.4.1.8	Reconst3D()	23
5.4.1.9	Superbee()	23
5.4.1.10	Weno5()	23
5.5	.main.c File Reference	23
5.5.1	Function Documentation	24

5.5.1.1	main()	24
5.6	.mesh.c File Reference	24
5.6.1	Function Documentation	24
5.6.1.1	Mesh()	24
5.7	.output.c File Reference	24
5.7.1	Function Documentation	25
5.7.1.1	Output1()	25
5.7.1.2	Output1_bin()	25
5.7.1.3	Output2()	25
5.7.1.4	Output2_bin()	25
5.7.1.5	Output3()	25
5.7.1.6	Output3_bin()	25
5.7.1.7	PrintValues()	26
5.8	.restar.c File Reference	26
5.8.1	Function Documentation	26
5.8.1.1	Restart()	26
5.8.1.2	Restart_Bin()	26
5.9	.runge-kutta.c File Reference	26
5.10	alloc.c File Reference	27
5.10.1	Detailed Description	27
5.10.2	Function Documentation	27
5.10.2.1	Allocate_Array()	27
5.10.2.2	New_Size()	27
5.11	auxfunc.c File Reference	27
5.11.1	Detailed Description	28
5.11.2	Function Documentation	28
5.11.2.1	CheckSimParameters()	28
5.11.2.2	Low_Index_Range1()	28
5.11.2.3	Low_Index_Range2()	28
5.11.2.4	MxV()	29

5.11.2.5	Raise_Index_Range1()	29
5.11.2.6	RoundGen()	29
5.11.2.7	Scalar_Contraction_Range1()	29
5.12	bound_cond.c File Reference	29
5.12.1	Detailed Description	30
5.12.2	Function Documentation	30
5.12.2.1	Outflow()	30
5.12.2.2	Periodic()	30
5.12.2.3	Reflection()	30
5.13	EOS/eos.c File Reference	30
5.13.1	Detailed Description	31
5.13.2	Function Documentation	31
5.13.2.1	EoS()	31
5.14	flux.c File Reference	31
5.14.1	Detailed Description	31
5.14.2	Function Documentation	32
5.14.2.1	Flux1D()	32
5.14.2.2	Flux2D()	32
5.14.2.3	Flux3D()	32
5.14.2.4	Hll()	32
5.14.2.5	Hllc()	32
5.15	HD/.u2qvector.c File Reference	33
5.15.1	Function Documentation	33
5.15.1.1	funct_U2Q()	33
5.16	RHD/.u2qvector.c File Reference	33
5.16.1	Function Documentation	33
5.16.1.1	funct_U2Q()	33
5.17	HD/.u2qvector2.c File Reference	33
5.17.1	Function Documentation	34
5.17.1.1	funct_U2Q()	34



5.18 RHD/u2qvector2.c File Reference . . . . .	34
5.18.1 Function Documentation . . . . .	34
5.18.1.1 funct_U2Q() . . . . .	34
5.19 HD/amatrix.c File Reference . . . . .	34
5.19.1 Function Documentation . . . . .	35
5.19.1.1 funct_A() . . . . .	35
5.20 RHD/amatrix.c File Reference . . . . .	35
5.20.1 Function Documentation . . . . .	35
5.20.1.1 funct_A() . . . . .	35
5.21 HD/fvector.c File Reference . . . . .	35
5.21.1 Detailed Description . . . . .	36
5.21.2 Function Documentation . . . . .	36
5.21.2.1 Prim2FluxF() . . . . .	36
5.22 RHD/fvector.c File Reference . . . . .	36
5.22.1 Function Documentation . . . . .	36
5.22.1.1 Prim2FluxF() . . . . .	36
5.23 HD/gvector.c File Reference . . . . .	36
5.23.1 Detailed Description . . . . .	37
5.23.2 Function Documentation . . . . .	37
5.23.2.1 Prim2FluxG() . . . . .	37
5.24 RHD/gvector.c File Reference . . . . .	37
5.24.1 Function Documentation . . . . .	37
5.24.1.1 Prim2FluxG() . . . . .	37
5.25 HD/hvector.c File Reference . . . . .	38
5.25.1 Detailed Description . . . . .	38
5.25.2 Function Documentation . . . . .	38
5.25.2.1 Prim2FluxH() . . . . .	38
5.26 RHD/hvector.c File Reference . . . . .	38
5.26.1 Function Documentation . . . . .	38
5.26.1.1 Prim2FluxH() . . . . .	39

5.27	HD/q2uvector.c File Reference	39
5.27.1	Function Documentation	39
5.27.1.1	Cons2Prim()	39
5.28	RHD/q2uvector.c File Reference	39
5.28.1	Detailed Description	39
5.28.2	Function Documentation	40
5.28.2.1	Cons2Prim()	40
5.29	HD/qvector.c File Reference	40
5.29.1	Detailed Description	40
5.29.2	Function Documentation	40
5.29.2.1	Prim2Cons()	40
5.30	RHD/qvector.c File Reference	40
5.30.1	Function Documentation	41
5.30.1.1	Prim2Cons()	41
5.31	HD/surface.c File Reference	41
5.31.1	Function Documentation	41
5.31.1.1	Surface_Volume()	41
5.32	RHD/surface.c File Reference	41
5.32.1	Function Documentation	42
5.32.1.1	Surface_Volume()	42
5.33	HD/svector.c File Reference	42
5.33.1	Function Documentation	42
5.33.1.1	Source_Terms()	42
5.34	RHD/svector.c File Reference	42
5.34.1	Function Documentation	42
5.34.1.1	Source_Terms()	43
5.35	HD/u2qvector.c File Reference	43
5.35.1	Function Documentation	43
5.35.1.1	Prim2Cons_All()	43
5.36	RHD/u2qvector.c File Reference	43

5.36.1	Function Documentation	43
5.36.1.1	Prim2Cons_All()	43
5.37	Headers/boundaries.h File Reference	44
5.37.1	Detailed Description	44
5.37.2	Function Documentation	44
5.37.2.1	Outflow()	44
5.37.2.2	Periodic()	44
5.37.2.3	Reflection()	44
5.38	Headers/const.h File Reference	45
5.38.1	Detailed Description	45
5.38.2	Macro Definition Documentation	45
5.38.2.1	c_cgs	45
5.38.2.2	G_cgs	45
5.38.2.3	G_cgs_earth	45
5.38.2.4	K_B_cgs	46
5.38.2.5	M_sol_cgs	46
5.38.2.6	mH_cgs	46
5.38.2.7	R_sol_cgs	46
5.38.2.8	yr_cgs	46
5.39	Headers/io.h File Reference	46
5.39.1	Detailed Description	47
5.39.2	Function Documentation	47
5.39.2.1	Output1()	47
5.39.2.2	Output1_bin()	47
5.39.2.3	Output2()	47
5.39.2.4	Output2_bin()	48
5.39.2.5	Output3()	48
5.39.2.6	Output3_bin()	48
5.39.2.7	PrintValues()	48
5.39.2.8	Read_Parameters_File()	48

5.39.2.9 Restart()	48
5.39.2.10 Restart_Bin()	48
5.39.2.11 User_Parameters()	49
5.39.3 Variable Documentation	49
5.39.3.1 binary	49
5.39.3.2 check_param	49
5.39.3.3 outputdirectory	49
5.39.3.4 outputfile	49
5.39.3.5 paramfile_name	49
5.39.3.6 restart_filecount	49
5.39.3.7 restart_simulation	50
5.39.3.8 restartfile	50
5.40 Headers/limiters.h File Reference	50
5.40.1 Detailed Description	50
5.40.2 Function Documentation	50
5.40.2.1 Godunov()	51
5.40.2.2 Limiter()	51
5.40.2.3 Maxmod()	51
5.40.2.4 Mc()	51
5.40.2.5 Minmod()	51
5.40.2.6 Reconst1D()	51
5.40.2.7 Reconst2D()	52
5.40.2.8 Reconst3D()	52
5.40.2.9 Superbee()	52
5.40.2.10 Weno5()	52
5.41 Headers/macros.h File Reference	52
5.41.1 Detailed Description	53
5.41.2 Macro Definition Documentation	53
5.41.2.1 CARTESIAN	53
5.41.2.2 CYLINDRICAL	54

5.41.2.3 DUST	54
5.41.2.4 FALSE	54
5.41.2.5 GODUNOV	54
5.41.2.6 HD	54
5.41.2.7 HLL	54
5.41.2.8 HLLC	54
5.41.2.9 IDEAL	54
5.41.2.10 LOGMESH	55
5.41.2.11 MAX	55
5.41.2.12 MC	55
5.41.2.13 MIN	55
5.41.2.14 MINMOD	55
5.41.2.15 PRE	55
5.41.2.16 PVRs	55
5.41.2.17 RHD	56
5.41.2.18 RHO	56
5.41.2.19 SPHERICAL	56
5.41.2.20 STANDARD	56
5.41.2.21 STIFF	56
5.41.2.22 SUPERBEE	56
5.41.2.23 TRUE	56
5.41.2.24 UNIFORM	56
5.41.2.25 VX1	57
5.41.2.26 VX2	57
5.41.2.27 VX3	57
5.41.2.28 WENO5	57
5.42 Headers/main.h File Reference	57
5.42.1 Detailed Description	58
5.42.2 Function Documentation	59
5.42.2.1 Allocate_Array()	59

5.42.2.2	AMATRIX1D()	59
5.42.2.3	AMATRIX2D()	59
5.42.2.4	AMATRIX3D()	59
5.42.2.5	Boundaries()	59
5.42.2.6	CheckSimParameters()	59
5.42.2.7	Flux1D()	60
5.42.2.8	Flux2D()	60
5.42.2.9	Flux3D()	60
5.42.2.10	Hll()	60
5.42.2.11	Hllc()	60
5.42.2.12	Initial()	60
5.42.2.13	Integration()	61
5.42.2.14	MxV()	61
5.42.2.15	New_Size()	61
5.42.2.16	RK1D()	61
5.42.2.17	RK2D()	61
5.42.2.18	RK3D()	61
5.42.2.19	RoundGen()	62
5.42.2.20	VECTOR()	62
5.42.3	Variable Documentation	62
5.42.3.1	CHECK_NAN	62
5.42.3.2	delta	62
5.42.3.3	K	62
5.42.3.4	Nx1	62
5.42.3.5	Nx2	63
5.42.3.6	Nx3	63
5.42.3.7	Q	63
5.42.3.8	Q1	63
5.42.3.9	Q2	63
5.42.3.10	Q3	63

5.42.3.11 start	63
5.42.3.12 U	63
5.42.3.13 U0	64
5.42.3.14 U1	64
5.42.3.15 U2	64
5.42.3.16 U3	64
5.42.3.17 x1max	64
5.42.3.18 x1min	64
5.42.3.19 x2max	64
5.42.3.20 x2min	64
5.42.3.21 x3max	65
5.42.3.22 x3min	65
5.43 Headers/matrix.h File Reference	65
5.43.1 Function Documentation	65
5.43.1.1 funct_A()	65
5.43.1.2 funct_Dm()	65
5.43.1.3 funct_Dn()	65
5.43.1.4 funct_Do()	66
5.44 Headers/mesh.h File Reference	66
5.44.1 Function Documentation	66
5.44.1.1 Mesh()	66
5.44.1.2 Surface_Volume()	66
5.44.1.3 TimeStep()	67
5.44.2 Variable Documentation	67
5.44.2.1 cou	67
5.44.2.2 dt	67
5.44.2.3 dx1	67
5.44.2.4 dx2	67
5.44.2.5 dx3	67
5.44.2.6 grid	67

5.44.2.7	timefile	68
5.44.2.8	tmax	68
5.45	Headers/mod.h File Reference	68
5.45.1	Function Documentation	68
5.45.1.1	funct_Mm()	68
5.45.1.2	funct_Mn()	68
5.45.1.3	funct_Mo()	69
5.46	Headers/physics.h File Reference	69
5.46.1	Function Documentation	69
5.46.1.1	Cons2Prim()	70
5.46.1.2	EoS()	70
5.46.1.3	funct_A()	70
5.46.1.4	Gauge_Derivatives()	70
5.46.1.5	Get_Metric_Components()	70
5.46.1.6	Low_Index_Range1()	70
5.46.1.7	Low_Index_Range2()	71
5.46.1.8	Prim2Cons()	71
5.46.1.9	Prim2Cons_All()	71
5.46.1.10	Prim2FluxF()	71
5.46.1.11	Prim2FluxG()	71
5.46.1.12	Prim2FluxH()	72
5.46.1.13	Raise_Index_Range1()	72
5.46.1.14	Scalar_Contraction_Range1()	72
5.46.1.15	Source_Terms()	72
5.46.1.16	Sources()	72
5.46.1.17	User_Source_Terms()	72
5.47	input.c File Reference	73
5.47.1	Detailed Description	73
5.47.2	Function Documentation	73
5.47.2.1	Read_Parameters_File()	73



5.47.3	Variable Documentation	73
5.47.3.1	paramfile	73
5.48	integration-pvrs.c File Reference	74
5.48.1	Function Documentation	74
5.48.1.1	INTEGRATION()	74
5.49	integration.c File Reference	74
5.49.1	Detailed Description	74
5.49.2	Function Documentation	74
5.49.2.1	Integration()	75
5.50	limiters.c File Reference	75
5.50.1	Function Documentation	75
5.50.1.1	Godunov()	75
5.50.1.2	Limiter()	75
5.50.1.3	Maxmod()	75
5.50.1.4	Mc()	76
5.50.1.5	Minmod()	76
5.50.1.6	Superbee()	76
5.50.1.7	Weno5()	76
5.51	main.c File Reference	76
5.51.1	Detailed Description	77
5.51.2	Function Documentation	77
5.51.2.1	main()	77
5.52	mesh.c File Reference	77
5.52.1	Function Documentation	77
5.52.1.1	Mesh()	77
5.53	output.c File Reference	77
5.53.1	Detailed Description	78
5.53.2	Function Documentation	78
5.53.2.1	PrintValues()	78
5.54	restart.c File Reference	78

5.54.1 Detailed Description . . . . .	78
5.54.2 Function Documentation . . . . .	78
5.54.2.1 Restart() . . . . .	79
5.54.2.2 Restart_Bin() . . . . .	79
5.55 RHD/.fvector.c File Reference . . . . .	79
5.55.1 Function Documentation . . . . .	79
5.55.1.1 Prim2FluxF() . . . . .	79
5.56 RHD/q2uvector.c File Reference . . . . .	79
5.56.1 Function Documentation . . . . .	79
5.56.1.1 Cons2Prim() . . . . .	80
5.57 RHD/qvector.c File Reference . . . . .	80
5.57.1 Function Documentation . . . . .	80
5.57.1.1 Prim2Cons() . . . . .	80
5.58 RHD/Boyer-Lindquist/metric.c File Reference . . . . .	80
5.58.1 Function Documentation . . . . .	80
5.58.1.1 Gauge_Derivatives() . . . . .	81
5.58.1.2 Get_Metric_Components() . . . . .	81
5.59 RHD/Eddington-Finkelstein/metric.c File Reference . . . . .	81
5.59.1 Function Documentation . . . . .	81
5.59.1.1 Gauge_Derivatives() . . . . .	81
5.59.1.2 Get_Metric_Components() . . . . .	81
5.60 RHD/Kerr-Schild/metric.c File Reference . . . . .	81
5.60.1 Function Documentation . . . . .	82
5.60.1.1 Gauge_Derivatives() . . . . .	82
5.60.1.2 Get_Metric_Components() . . . . .	82
5.61 RHD/Minkowski/metric.c File Reference . . . . .	82
5.61.1 Function Documentation . . . . .	82
5.61.1.1 Gauge_Derivatives() . . . . .	82
5.61.1.2 Get_Metric_Components() . . . . .	83
5.62 RHD/Schwarzschild/metric.c File Reference . . . . .	83

5.62.1	Function Documentation	83
5.62.1.1	Gauge_Derivatives()	83
5.62.1.2	Get_Metric_Components()	83
5.63	RHD/Kerr-Schild/.metric.c File Reference	83
5.63.1	Function Documentation	84
5.63.1.1	Gauge_Derivatives()	84
5.63.1.2	Get_Metric_Components()	84
5.64	RHD/Kerr-Schild/.metric2.c File Reference	84
5.64.1	Function Documentation	84
5.64.1.1	Gauge_Derivatives()	84
5.64.1.2	Get_Metric_Components()	84
5.65	RHD/Kerr-Schild/.metric3.c File Reference	85
5.65.1	Function Documentation	85
5.65.1.1	Gauge_Derivatives()	85
5.65.1.2	Get_Metric_Components()	85
5.66	RHD/Kerr-Schild/.metric4.c File Reference	85
5.66.1	Function Documentation	85
5.66.1.1	Gauge_Derivatives()	86
5.66.1.2	Get_Metric_Components()	86
5.67	RHD/Kerr-Schild/.metric5.c File Reference	86
5.67.1	Function Documentation	86
5.67.1.1	Gauge_Derivatives()	86
5.67.1.2	Get_Metric_Components()	86
5.68	RHD/new_q2uvector.c File Reference	86
5.68.1	Function Documentation	87
5.68.1.1	Cons2Prim()	87
5.69	RHD/stiff_q2uvector.c File Reference	87
5.69.1	Function Documentation	87
5.69.1.1	Cons2Prim()	87
5.70	runge-kutta.c File Reference	87
5.71	timestep.c File Reference	87
5.71.1	Detailed Description	88
5.71.2	Function Documentation	88
5.71.2.1	TimeStep()	88
5.72	vectors.c File Reference	88
5.72.1	Function Documentation	88
5.72.1.1	Sources()	88
5.72.1.2	VECTOR()	88



## Chapter 1

# AZTEKAS: a hydrodynamic GPL code

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">der_gauge_</a>	7
<a href="#">eos_</a>	8
<a href="#">flx_</a>	8
<a href="#">gauge_</a>	10
<a href="#">grid_</a>	11
<a href="#">lim_</a>	14
<a href="#">vec_</a>	16





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">alloc.c</a>	Essential allocation functions for <i>aztekas</i> . . . . .	27
<a href="#">auxfunc.c</a>	Helpful functions for <i>aztekas</i> . . . . .	27
<a href="#">bound_cond.c</a>	Standard boundary conditions. Outflow, Periodic and Reflection . . . . .	29
<a href="#">flux.c</a>	Numerical flux computing and implementation . . . . .	31
<a href="#">input.c</a>	Important input parameters for <i>aztekas</i> . . . . .	73
<a href="#">integration.c</a>	Main function for the time integration in the conservative variables $Q$ . . . . .	74
<a href="#">main.c</a>	Main file of <i>aztekas</i> . . . . .	76
<a href="#">output.c</a>	Output functions: ASCII and Binary . . . . .	77
<a href="#">restart.c</a>	Functions to restart from a given file . . . . .	78
<a href="#">timestep.c</a>	Time-step calculation . . . . .	87
EOS/ <a href="#">eos.c</a>	Equation of state . . . . .	30
HD/ <a href="#">fvector.c</a>	. . . . .	35
HD/ <a href="#">gvector.c</a>	. . . . .	36
HD/ <a href="#">hvector.c</a>	. . . . .	38
HD/ <a href="#">qvector.c</a>	Function that converts Primitives to Conservative variables (HD) . . . . .	40
Headers/ <a href="#">boundaries.h</a>	Boundary condition functions definitions . . . . .	44
Headers/ <a href="#">const.h</a>	Physical and numerical constants . . . . .	45
Headers/ <a href="#">io.h</a>	Input and output function and variable definitions . . . . .	46
Headers/ <a href="#">limiters.h</a>	Reconstruction variables and functions definitions . . . . .	50

Headers/ <a href="#">macros.h</a>	
Macros definitios for <i>aztekas</i> . . . . .	52
Headers/ <a href="#">main.h</a>	
Main function, headers and variable declaration . . . . .	57
Headers/ <a href="#">matrix.h</a> . . . . .	65
Headers/ <a href="#">mesh.h</a> . . . . .	66
Headers/ <a href="#">mod.h</a> . . . . .	68
Headers/ <a href="#">physics.h</a> . . . . .	69
RHD/ <a href="#">q2uvector.c</a>	
Function that convert from Conservative to Primitives (RHD) . . . . .	39

## Chapter 4

# Class Documentation

### 4.1 der\_gauge\_ Struct Reference

#### Public Attributes

- double **dlapse** [3]
- double **dbeta** [3][3]
- double **dgam** [3][3][3]

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

- Headers/mesh.h

### 4.2 eos\_ Struct Reference

#### Public Attributes

- double **e**
- double **cs**
- double **h**

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

- Headers/physics.h

### 4.3 flx\_ Struct Reference

#### Public Attributes

- double **up** [eq+1]
- double **um** [eq+1]
- double **qp** [eq+1]
- double **qm** [eq+1]
- double **fp** [eq+1]
- double **fm** [eq+1]
- double **lp**
- double **lm**

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

- Headers/physics.h

## 4.4 gauge\_ Struct Reference

### Public Attributes

- double **x** [4]
- double **lapse**
- double **beta\_con** [3]
- double **gamma\_con** [3][3]
- double **dety**

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

- Headers/mesh.h

## 4.5 grid\_ Struct Reference

### Public Attributes

- double **time**
- double \* **X1**
- double \* **X1p**
- double \* **X1m**
- double \* **X2**
- double \* **X2p**
- double \* **X2m**
- double \* **X3**
- double \* **X3p**
- double \* **X3m**
- double \* **S1p**
- double \* **S1m**
- double \* **S2p**
- double \* **S2m**
- double \* **S3p**
- double \* **S3m**

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

- Headers/mesh.h

## 4.6 lim\_ Struct Reference

```
#include <limiters.h>
```

### Public Attributes

- double **ux1p** [2 \*eq]
- double **ux1m** [2 \*eq]
- double **sx1** [2 \*eq]
- double **ux2p** [2 \*eq]
- double **ux2m** [2 \*eq]
- double **sx2** [2 \*eq]
- double **ux3p** [2 \*eq]
- double **ux3m** [2 \*eq]
- double **sx3** [2 \*eq]
- double **ux** [2 \*eq]

#### 4.6.1 Detailed Description

The structure **lim\_** contains vectors in which the reconstructed variables of  $U$  in each cell are stored.

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

- Headers/[limiters.h](#)

## 4.7 vec\_ Struct Reference

### Public Attributes

- double **A** [(eq+1) \*(eq+1)]
- double **S** [eq+1]
- double **Fp** [eq+1]
- double **Fm** [eq+1]
- double **Gp** [eq+1]
- double **Gm** [eq+1]
- double **Hp** [eq+1]
- double **Hm** [eq+1]

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

- Headers/physics.h



## Chapter 5

# File Documentation

### 5.1 alloc.c File Reference

Essential allocation functions for *aztekas*.

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

#### Functions

- void [Allocate\\_Array](#) ()
- void **New\_Size** ()

#### 5.1.1 Detailed Description

##### Author

Alejandro Aguayo-Ortiz

#### 5.1.2 Function Documentation

##### 5.1.2.1 Allocate\_Array()

```
void Allocate_Array ( )
```

This function allocates the space in memory for all the vectors used in *aztekas*.

## 5.2 auxfunc.c File Reference

Helpful functions for *aztekas*.

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

### Functions

- int **MxV** (double \*M, double \*V, double \*L)
- void **RoundGen** (double \*num)
- void **Scalar\_Contraction\_Range1** (double \*scalar, double \*cov, double \*con)
- void **Raise\_Index\_Range1** (double \*con, double \*cov, [gauge\\_](#) \*local\_grid)
- void **Low\_Index\_Range1** (double \*cov, double \*con, [gauge\\_](#) \*local\_grid)
- void **Low\_Index\_Range2** (double \*\*diag, double \*\*con, [gauge\\_](#) \*local\_grid)
- void **CheckSimParameters** ()

#### 5.2.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

## 5.3 bound\_cond.c File Reference

Standard boundary conditions. Outflow, Periodic and Reflection.

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

### Functions

- void [Outflow](#) (double \*B)
- void [Reflection](#) (double \*B)
- void [Periodic](#) (double \*B)

#### 5.3.1 Detailed Description

Author

Alejandro Aguayo-Ortiz

#### 5.3.2 Function Documentation



### 5.3.2.1 Outflow()

```
void Outflow (
    double * B )
```

The function **Outflow()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the last computed cell of the domain.

### 5.3.2.2 Periodic()

```
void Periodic (
    double * B )
```

The function **Periodic()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells with the values of the correspondent other side of the domain.

### 5.3.2.3 Reflection()

```
void Reflection (
    double * B )
```

The function **Reflection()**, receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the mirrored cells, and for the velocity it changes sign.

## 5.4 EOS/eos.c File Reference

Equation of state.

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

### Functions

- void **EoS** ([eos\\_](#) \*eos, double \*u, [gauge\\_](#) local\_grid)

### 5.4.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.5 flux.c File Reference

Numerical flux computing and implementation.

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

## Functions

- int **Flux1D** ([vec\\_](#) \*v, [lim\\_](#) \*l, int \*l)
- int **Flux2D** ([vec\\_](#) \*v, [lim\\_](#) \*l, int \*l)
- int **Flux3D** ([vec\\_](#) \*v, [lim\\_](#) \*l, int \*l)
- int **HII** (double \*F, [flx\\_](#) \*f, int x)
- int **HIIc** (double \*F, [flx\\_](#) \*f, int x)

### 5.5.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

The functions receives as an argument the structures [vec\\_](#) and [lim\\_](#) and the integer vector **l**. The structures carries the values of the Numerical Fluxes, e.g.  $\mathbf{Fp} = \mathcal{F}_{i+1/2}$ .

## 5.6 HD/fvector.c File Reference

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

## Functions

- void **Prim2FluxF** (double \*f, double \*v, double \*u, [gauge\\_](#) local\_grid)

### 5.6.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.7 HD/gvector.c File Reference

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

## Functions

- void **Prim2FluxG** (double \*f, double \*v, double \*u, [gauge\\_](#) local\_grid)

### 5.7.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.8 HD/hvector.c File Reference

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

### Functions

- void **Prim2FluxH** (double \*f, double \*v, double \*u, [gauge\\_](#) local\_grid)

### 5.8.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.9 RHD/q2uvector.c File Reference

Function that convert from Conservative to Primitives (RHD).

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

### Functions

- int **Cons2Prim** (double \*u, double \*q)

### 5.9.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.10 HD/qvector.c File Reference

Function that converts Primitives to Conservative variables (HD).

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

### Functions

- void **Prim2Cons** (double \*q, double \*u, [gauge\\_](#) local\_grid)

### 5.10.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.11 Headers/boundaries.h File Reference

Boundary condition functions definitions.

### Functions

- void [Outflow](#) (double \*B)
- void [Periodic](#) (double \*B)
- void [Reflection](#) (double \*B)

### 5.11.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

### 5.11.2 Function Documentation

#### 5.11.2.1 Outflow()

```
void Outflow (  
    double * B )
```

The function [Outflow\(\)](#), receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the last computed cell of the domain.

#### 5.11.2.2 Periodic()

```
void Periodic (  
    double * B )
```

The function [Periodic\(\)](#), receives the vector solution as an parameter **B**. It fills the value of the ghost cells with the values of the correspondent other side of the domain.

#### 5.11.2.3 Reflection()

```
void Reflection (  
    double * B )
```

The function [Reflection\(\)](#), receives the vector solution as an parameter **B**. It fills the value of the ghost cells in the specified direction using the value of the mirrored cells, and for the velocity it changes sign.

## 5.12 Headers/const.h File Reference

Physical and numerical constants.

### Macros

- `#define G_cgs` 6.67408e-08
- `#define M_sol_cgs` 1.98855e-33
- `#define R_sol_cgs` 6.957e-10
- `#define K_B_cgs` 1.3806488e-16
- `#define mH_cgs` 1.66e-24
- `#define c_cgs` 3e+10
- `#define G_cgs_earth` 4\*(3.14159265358979323846)\*(3.14159265358979323846)
- `#define yr_cgs` 3.14e+07

### 5.12.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.13 Headers/io.h File Reference

Input and output function and variable definitions.

### Functions

- int **PrintValues** (double \*tprint, double \*dtprint, int \*itprint)
- int **Output1** (int \*itprint)
- int **Output2** (int \*itprint)
- int **Output3** (int \*itprint)
- int **Output1\_bin** (int \*itprint)
- int **Output2\_bin** (int \*itprint)
- int **Output3\_bin** (int \*itprint)
- void **Restart** ()
- void **Restart\_Bin** ()
- int **Read\_Parameters\_File** (char const \*paramfile\_name)
- int **User\_Parameters** (char const \*paramfile\_name)

### Variables

- int **binary**
- int **check\_param**
- int **restart\_simulation**
- int **restart\_filecount**
- char **paramfile\_name** [50]
- char **outputdirectory** [50]
- char **outputfile** [50]
- char **restartfile** [50]

### 5.13.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.14 Headers/limiters.h File Reference

Reconstruction variables and functions definitions.

### Classes

- struct [lim\\_](#)

### Functions

- double **Limiter** (double A, double B, int r)
- double **Godunov** (double A, double B)
- double **Maxmod** (double A, double B)
- double **Minmod** (double A, double B)
- double **Mc** (double A, double B)
- double **Superbee** (double A, double B)
- double **Weno5** (double v1, double v2, double v3, double v4, double v5)
- int **Reconst1D** (double \*u, [lim\\_](#) \*l, int \*l)
- int **Reconst2D** (double \*u, [lim\\_](#) \*l, int \*l)
- int **Reconst3D** (double \*u, [lim\\_](#) \*l, int \*l)

### 5.14.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.15 Headers/macros.h File Reference

Macros definitios for *aztekas*.

## Macros

- `#define MIN(a, b) (((a)<(b))?(a):(b))`
- `#define MAX(a, b) (((a)>(b))?(a):(b))`
- `#define TRUE 1`
- `#define FALSE 0`
- `#define HD 0`
- `#define RHD 1`
- `#define IDEAL 0`
- `#define DUST 1`
- `#define STIFF 2`
- `#define CARTESIAN 0`
- `#define CYLINDRICAL 1`
- `#define SPHERICAL 2`
- `#define UNIFORM 0`
- `#define LOGMESH 1`
- `#define HLL 0`
- `#define HLLC 1`
- `#define STANDARD 0`
- `#define PVR5 1`
- `#define GODUNOV 0`
- `#define MINMOD 1`
- `#define MC 2`
- `#define SUPERBEE 3`
- `#define WENO5 4`
- `#define RHO 0`
- `#define PRE 1`
- `#define VX1 2`
- `#define VX2 3`
- `#define VX3 4`

### 5.15.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.16 Headers/main.h File Reference

Main function, headers and variable declaration.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <omp.h>
#include "physics.h"
#include "boundaries.h"
#include "limiters.h"
#include "const.h"
#include "macros.h"
#include "io.h"
#include "user_param.h"
```

## Functions

- void [Allocate\\_Array](#) ()
- void **New\_Size** ()
- void **Initial** ()
- int **Boundaries** (double \*B)
- void **Integration** ()
- int **RK1D** (double \*u, double \*q, double \*q1, double \*q2, int order)
- int **RK2D** (double \*u, double \*q, double \*q1, double \*q2, int order)
- int **RK3D** (double \*u, double \*q, double \*q1, double \*q2, int order)
- int **Flux1D** ([vec\\_](#) \*v, [lim\\_](#) \*l, int \*l)
- int **Flux2D** ([vec\\_](#) \*v, [lim\\_](#) \*l, int \*l)
- int **Flux3D** ([vec\\_](#) \*v, [lim\\_](#) \*l, int \*l)
- int **Hll** (double \*F, [flx\\_](#) \*f, int x)
- int **Hllc** (double \*F, [flx\\_](#) \*f, int x)
- int **AMATRIX1D** (double \*u, [vec\\_](#) \*v, int \*l)
- int **AMATRIX2D** (double \*u, [vec\\_](#) \*v, int \*l)
- int **AMATRIX3D** (double \*u, [vec\\_](#) \*v, int \*l)
- int **VECTOR** (int pm, char flux, [lim\\_](#) \*l, [flx\\_](#) \*f, int \*l)
- int **MxV** (double \*M, double \*V, double \*L)
- void **RoundGen** (double \*num)
- void **CheckSimParameters** ()

## Variables

- int **CHECK\_NAN**
- double \* **U**
- double \* **U0**
- double \* **U1**
- double \* **U2**
- double \* **U3**
- double \* **Q**
- double \* **Q1**
- double \* **Q2**
- double \* **Q3**
- double **start**
- double **delta**
- double **K**
- int **Nx1**
- int **Nx2**
- int **Nx3**
- double **x1max**
- double **x2max**
- double **x3max**
- double **x1min**
- double **x2min**
- double **x3min**

### 5.16.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz



## 5.16.2 Function Documentation

### 5.16.2.1 Allocate\_Array()

```
void Allocate_Array ( )
```

This function allocates the space in memory for all the vectors used in *aztekas*.

## 5.17 input.c File Reference

Important input parameters for *aztekas*.

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

### Functions

- int **Read\_Parameters\_File** (char const \*paramfile\_name)

### Variables

- FILE \* **paramfile**

### 5.17.1 Detailed Description

#### Author

Emilio Tejeda

## 5.18 integration.c File Reference

Main function for the time integration in the conservative variables **Q**.

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

### Functions

- void **Integration** ()

### 5.18.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz

## 5.19 main.c File Reference

Main file of aztekas.

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

### Functions

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

### 5.19.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz.

## 5.20 output.c File Reference

Output functions: ASCII and Binary.

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

### Functions

- int **PrintValues** (double \*tprint, double \*dtprint, int \*itprint)

### 5.20.1 Detailed Description

#### Authors

Alejandro Aguayo-Ortiz and Emilio Tejeda

## 5.21 restart.c File Reference

Functions to restart from a given file.

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

## Functions

- void **Restart** ()
- void **Restart\_Bin** ()

### 5.21.1 Detailed Description

#### Author

Emilio Tejeda

## 5.22 timestep.c File Reference

Time-step calculation.

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

## Functions

- double **TimeStep** ()

### 5.22.1 Detailed Description

#### Author

Alejandro Aguayo-Ortiz



# Index

alloc.c, [11](#)  
    Allocate\_Array, [11](#)

Allocate\_Array  
    alloc.c, [11](#)  
    main.h, [21](#)

auxfunc.c, [12](#)

bound\_cond.c, [12](#)  
    Outflow, [13](#)  
    Periodic, [13](#)  
    Reflection, [13](#)

boundaries.h  
    Outflow, [16](#)  
    Periodic, [17](#)  
    Reflection, [17](#)

der\_gauge\_, [7](#)

EOS/eos.c, [13](#)

eos\_, [7](#)

flux.c, [14](#)

flx\_, [7](#)

gauge\_, [8](#)

grid\_, [8](#)

HD/fvector.c, [14](#)  
HD/gvector.c, [14](#)  
HD/hvector.c, [15](#)  
HD/qvector.c, [16](#)  
Headers/boundaries.h, [16](#)  
Headers/const.h, [17](#)  
Headers/io.h, [18](#)  
Headers/limiters.h, [18](#)  
Headers/macros.h, [19](#)  
Headers/main.h, [20](#)

input.c, [21](#)

integration.c, [22](#)

lim\_, [8](#)

main.c, [22](#)

main.h  
    Allocate\_Array, [21](#)

Outflow  
    bound\_cond.c, [13](#)  
    boundaries.h, [16](#)  
output.c, [23](#)

Periodic

    bound\_cond.c, [13](#)  
    boundaries.h, [17](#)

RHD/q2uvector.c, [15](#)

Reflection  
    bound\_cond.c, [13](#)  
    boundaries.h, [17](#)

restart.c, [23](#)

timestep.c, [23](#)

vec\_, [9](#)