HERACLES

Generated by Doxygen 1.6.1

Fri Jul 23 19:02:55 2010

Contents

1 Main Page **1**

2 Directory Hierarchy **3**

2.1 Directories 3

3 Modules Index **5**

3.1 Modules List 5

4 Data Type Index **7**

4.1 Data Types List 7

5 File Index **9**

5.1 File List 9

6 Directory Documentation **13**

6.1 algebra/ Directory Reference 13

6.2 conduction/ Directory Reference 14

6.3 rad_transfer/fld/ Directory Reference 15

6.4 gravity/ Directory Reference 16

6.5 hdf/ Directory Reference 17

6.6 hydro/ Directory Reference 18

6.7 init/ Directory Reference 19

6.8 rad_transfer/m1/ Directory Reference 20

6.9 main/ Directory Reference 22

6.10 mhd/ Directory Reference 24

6.11 rad_transfer/ Directory Reference 25

7 Module Documentation **27**

7.1 cellules Module Reference 27

7.1.1 Detailed Description 28

7.1.2 Variable Documentation . . . 28
7.1.2.1 dt_film . . . 28
7.1.2.2 dx_cell . . . 28
7.1.2.3 fff . . . 28
7.1.2.4 fff_t . . . 28
7.1.2.5 ifich . . . 29
7.1.2.6 ifilm . . . 29
7.1.2.7 Imax . . . 29
7.1.2.8 Imax2 . . . 29
7.1.2.9 ioffset . . . 29
7.1.2.10 N_cell . . . 29
7.1.2.11 Pcell . . . 29
7.1.2.12 Pmax . . . 29
7.1.2.13 Pmax2 . . . 29
7.1.2.14 size_offset . . . 30
7.1.2.15 t_film . . . 30
7.2 cl Module Reference . . . 31
7.2.1 Detailed Description . . . 31
7.2.2 Variable Documentation . . . 32
7.2.2.1 B_l . . . 32
7.2.2.2 E_l . . . 32
7.2.2.3 fx_l . . . 32
7.2.2.4 icl . . . 32
7.2.2.5 icl0 . . . 32
7.2.2.6 icl_glob . . . 32
7.2.2.7 P_l . . . 32
7.2.2.8 rho_l . . . 33
7.2.2.9 rhou_l . . . 33
7.2.2.10 T_l . . . 33
7.2.2.11 u_l . . . 33
7.3 cl_con Module Reference . . . 34
7.3.1 Detailed Description . . . 34
7.3.2 Variable Documentation . . . 34
7.3.2.1 icl_con . . . 34
7.3.2.2 icl_con_glob . . . 34
7.3.2.3 T_l . . . 34

7.4 cl_dif Module Reference . . . 35
7.4.1 Detailed Description . . . 35
7.4.2 Variable Documentation . . . 35
7.4.2.1 Er_dif_l . . . 35
7.4.2.2 icl_dif . . . 35
7.4.2.3 icl_dif_glob . . . 35
7.5 cl_ray Module Reference . . . 36
7.5.1 Detailed Description . . . 36
7.5.2 Variable Documentation . . . 36
7.5.2.1 Er_l . . . 36
7.5.2.2 Fr_l . . . 36
7.5.2.3 icl_ray . . . 36
7.5.2.4 icl_ray_glob . . . 37
7.5.2.5 Tr_l . . . 37
7.6 communication Module Reference . . . 38
7.6.1 Detailed Description . . . 41
7.6.2 Variable Documentation . . . 41
7.6.2.1 ar13_e . . . 41
7.6.2.2 ar13_r . . . 42
7.6.2.3 ar14_e . . . 42
7.6.2.4 ar14_r . . . 42
7.6.2.5 ar15_e . . . 42
7.6.2.6 ar15_r . . . 42
7.6.2.7 ar16_e . . . 42
7.6.2.8 ar16_r . . . 42
7.6.2.9 ar23_e . . . 43
7.6.2.10 ar23_r . . . 43
7.6.2.11 ar24_e . . . 43
7.6.2.12 ar24_r . . . 43
7.6.2.13 ar25_e . . . 43
7.6.2.14 ar25_r . . . 43
7.6.2.15 ar26_e . . . 43
7.6.2.16 ar26_r . . . 44
7.6.2.17 ar35_e . . . 44
7.6.2.18 ar35_r . . . 44
7.6.2.19 ar36_e . . . 44

7.6.2.20 ar36_r . . . 44
7.6.2.21 ar45_e . . . 44
7.6.2.22 ar45_r . . . 44
7.6.2.23 ar46_e . . . 45
7.6.2.24 ar46_r . . . 45
7.6.2.25 coin135_e . . . 45
7.6.2.26 coin135_r . . . 45
7.6.2.27 coin136_e . . . 45
7.6.2.28 coin136_r . . . 45
7.6.2.29 coin13_e . . . 45
7.6.2.30 coin13_r . . . 46
7.6.2.31 coin145_e . . . 46
7.6.2.32 coin145_r . . . 46
7.6.2.33 coin146_e . . . 46
7.6.2.34 coin146_r . . . 46
7.6.2.35 coin14_e . . . 46
7.6.2.36 coin14_r . . . 46
7.6.2.37 coin235_e . . . 47
7.6.2.38 coin235_r . . . 47
7.6.2.39 coin236_e . . . 47
7.6.2.40 coin236_r . . . 47
7.6.2.41 coin23_e . . . 47
7.6.2.42 coin23_r . . . 47
7.6.2.43 coin245_e . . . 47
7.6.2.44 coin245_r . . . 48
7.6.2.45 coin246_e . . . 48
7.6.2.46 coin246_r . . . 48
7.6.2.47 coin24_e . . . 48
7.6.2.48 coin24_r . . . 48
7.6.2.49 face1_e . . . 48
7.6.2.50 face1_r . . . 48
7.6.2.51 face2_e . . . 49
7.6.2.52 face2_r . . . 49
7.6.2.53 face3_e . . . 49
7.6.2.54 face3_r . . . 49
7.6.2.55 face4_e . . . 49

7.6.2.56 face4_r . . . 49
7.6.2.57 face5_e . . . 49
7.6.2.58 face5_r . . . 50
7.6.2.59 face6_e . . . 50
7.6.2.60 face6_r . . . 50
7.7 communication_gen Module Reference . . . 51
7.7.1 Detailed Description . . . 54
7.7.2 Variable Documentation . . . 55
7.7.2.1 ar13_e . . . 55
7.7.2.2 ar13_r . . . 55
7.7.2.3 ar14_e . . . 55
7.7.2.4 ar14_r . . . 55
7.7.2.5 ar15_e . . . 55
7.7.2.6 ar15_r . . . 55
7.7.2.7 ar16_e . . . 55
7.7.2.8 ar16_r . . . 56
7.7.2.9 ar23_e . . . 56
7.7.2.10 ar23_r . . . 56
7.7.2.11 ar24_e . . . 56
7.7.2.12 ar24_r . . . 56
7.7.2.13 ar25_e . . . 56
7.7.2.14 ar25_r . . . 56
7.7.2.15 ar26_e . . . 57
7.7.2.16 ar26_r . . . 57
7.7.2.17 ar35_e . . . 57
7.7.2.18 ar35_r . . . 57
7.7.2.19 ar36_e . . . 57
7.7.2.20 ar36_r . . . 57
7.7.2.21 ar45_e . . . 57
7.7.2.22 ar45_r . . . 58
7.7.2.23 ar46_e . . . 58
7.7.2.24 ar46_r . . . 58
7.7.2.25 coin135_e . . . 58
7.7.2.26 coin135_r . . . 58
7.7.2.27 coin136_e . . . 58
7.7.2.28 coin136_r . . . 58

7.7.2.29 coin13_e . . . 59
7.7.2.30 coin13_r . . . 59
7.7.2.31 coin145_e . . . 59
7.7.2.32 coin145_r . . . 59
7.7.2.33 coin146_e . . . 59
7.7.2.34 coin146_r . . . 59
7.7.2.35 coin14_e . . . 59
7.7.2.36 coin14_r . . . 60
7.7.2.37 coin235_e . . . 60
7.7.2.38 coin235_r . . . 60
7.7.2.39 coin236_e . . . 60
7.7.2.40 coin236_r . . . 60
7.7.2.41 coin23_e . . . 60
7.7.2.42 coin23_r . . . 60
7.7.2.43 coin245_e . . . 61
7.7.2.44 coin245_r . . . 61
7.7.2.45 coin246_e . . . 61
7.7.2.46 coin246_r . . . 61
7.7.2.47 coin24_e . . . 61
7.7.2.48 coin24_r . . . 61
7.7.2.49 face1_e . . . 61
7.7.2.50 face1_r . . . 62
7.7.2.51 face2_e . . . 62
7.7.2.52 face2_r . . . 62
7.7.2.53 face3_e . . . 62
7.7.2.54 face3_r . . . 62
7.7.2.55 face4_e . . . 62
7.7.2.56 face4_r . . . 62
7.7.2.57 face5_e . . . 63
7.7.2.58 face5_r . . . 63
7.7.2.59 face6_e . . . 63
7.7.2.60 face6_r . . . 63
7.7.2.61 Nmax . . . 63
7.8 communication_ray Module Reference . . . 64
7.8.1 Detailed Description . . . 65
7.8.2 Variable Documentation . . . 66

7.8.2.1 ar13_e . . . 66
7.8.2.2 ar13_r . . . 66
7.8.2.3 ar15_e . . . 66
7.8.2.4 ar15_r . . . 66
7.8.2.5 ar16_e . . . 66
7.8.2.6 ar16_r . . . 66
7.8.2.7 ar23_e . . . 66
7.8.2.8 ar23_r . . . 67
7.8.2.9 ar35_e . . . 67
7.8.2.10 ar35_r . . . 67
7.8.2.11 ar36_e . . . 67
7.8.2.12 ar36_r . . . 67
7.8.2.13 coin135_e . . . 67
7.8.2.14 coin135_r . . . 67
7.8.2.15 coin13_e . . . 68
7.8.2.16 coin13_r . . . 68
7.8.2.17 coin145_e . . . 68
7.8.2.18 coin145_r . . . 68
7.8.2.19 coin235_e . . . 68
7.8.2.20 coin235_r . . . 68
7.8.2.21 coin23_e . . . 68
7.8.2.22 coin23_r . . . 69
7.8.2.23 coin245_e . . . 69
7.8.2.24 coin245_r . . . 69
7.8.2.25 face1_e . . . 69
7.8.2.26 face1_r . . . 69
7.8.2.27 face3_e . . . 69
7.8.2.28 face3_r . . . 69
7.8.2.29 face5_e . . . 70
7.8.2.30 face5_r . . . 70
7.9 conduction Module Reference . . . 71
7.9.1 Detailed Description . . . 71
7.9.2 Variable Documentation . . . 71
7.9.2.1 dt_con_imp . . . 71
7.9.2.2 implicit_con . . . 71
7.9.2.3 nitetot_con . . . 72

7.9.2.4 precond_con . . . 72
7.9.2.5 vardt_con . . . 72
7.9.2.6 varrel_con . . . 72
7.10 const Module Reference . . . 73
7.10.1 Detailed Description . . . 73
7.10.2 Variable Documentation . . . 74
7.10.2.1 bigreal . . . 74
7.10.2.2 forth . . . 74
7.10.2.3 four . . . 74
7.10.2.4 half . . . 74
7.10.2.5 one . . . 74
7.10.2.6 sixth . . . 74
7.10.2.7 third . . . 74
7.10.2.8 three . . . 75
7.10.2.9 two . . . 75
7.10.2.10 two3rd . . . 75
7.10.2.11 zero . . . 75
7.11 cputime Module Reference . . . 76
7.11.1 Detailed Description . . . 76
7.11.2 Variable Documentation . . . 76
7.11.2.1 temps1 . . . 76
7.11.2.2 temps_comm . . . 76
7.11.2.3 temps_conduc . . . 77
7.11.2.4 temps_dif . . . 77
7.11.2.5 temps_hydro . . . 77
7.11.2.6 temps_io . . . 77
7.11.2.7 temps_ray . . . 77
7.11.2.8 temps_ref . . . 77
7.12 diffusion Module Reference . . . 78
7.12.1 Detailed Description . . . 78
7.12.2 Variable Documentation . . . 78
7.12.2.1 Dif_imp . . . 78
7.12.2.2 dt_dif . . . 79
7.12.2.3 dt_dif_exp . . . 79
7.12.2.4 dt_dif_imp . . . 79
7.12.2.5 Er_dif . . . 79

7.12.2.6 fld_limiter . . . 79
7.12.2.7 nitetot_dif . . . 79
7.12.2.8 precond_dif . . . 79
7.12.2.9 vardt_dif . . . 80
7.12.2.10 varrel_dif . . . 80
7.13 divers Module Reference . . . 81
7.13.1 Detailed Description . . . 84
7.13.2 Variable Documentation . . . 84
7.13.2.1 alpha . . . 84
7.13.2.2 COM . . . 85
7.13.2.3 comments . . . 85
7.13.2.4 CONDUC . . . 85
7.13.2.5 datanames . . . 85
7.13.2.6 dataunits . . . 85
7.13.2.7 DETO . . . 85
7.13.2.8 DIFFU . . . 85
7.13.2.9 dt . . . 86
7.13.2.10 dt_cfl . . . 86
7.13.2.11 dt_ff . . . 86
7.13.2.12 dt_glob . . . 86
7.13.2.13 dt_max . . . 86
7.13.2.14 dt_mhd . . . 86
7.13.2.15 dtold . . . 86
7.13.2.16 fourdim . . . 87
7.13.2.17 GRAV . . . 87
7.13.2.18 HISTOGRAMME . . . 87
7.13.2.19 HYDRO . . . 87
7.13.2.20 i_restart . . . 87
7.13.2.21 iout . . . 87
7.13.2.22 irepr . . . 88
7.13.2.23 ISOTHERME . . . 88
7.13.2.24 limdx . . . 88
7.13.2.25 limfr . . . 88
7.13.2.26 limnite . . . 88
7.13.2.27 mu . . . 88
7.13.2.28 name_of_job . . . 88

7.13.2.29 ndata . . . 88
7.13.2.30 Nimp . . . 89
7.13.2.31 nout . . . 89
7.13.2.32 noutd . . . 89
7.13.2.33 nrepr . . . 89
7.13.2.34 nreprd . . . 89
7.13.2.35 nsupp . . . 89
7.13.2.36 nsx . . . 89
7.13.2.37 nsy . . . 90
7.13.2.38 nsz . . . 90
7.13.2.39 rankine . . . 90
7.13.2.40 RAY . . . 90
7.13.2.41 REF . . . 90
7.13.2.42 reprise . . . 90
7.13.2.43 Solver . . . 91
7.13.2.44 sortie . . . 91
7.13.2.45 STIR . . . 91
7.13.2.46 subcycle . . . 91
7.13.2.47 suppdim . . . 91
7.13.2.48 suppnames . . . 91
7.13.2.49 suppunits . . . 91
7.13.2.50 swip . . . 91
7.13.2.51 tcpu_ini . . . 92
7.13.2.52 tcpu_last . . . 92
7.13.2.53 tcpu_max . . . 92
7.13.2.54 tcpu_min . . . 92
7.13.2.55 temps . . . 92
7.13.2.56 tend . . . 92
7.13.2.57 tout . . . 92
7.13.2.58 trepr . . . 93
7.13.2.59 verbose . . . 93
7.14 divers_ray Module Reference . . . 94
7.14.1 Detailed Description . . . 95
7.14.2 Variable Documentation . . . 95
7.14.2.1 alpha . . . 95
7.14.2.2 alpha_var . . . 95

7.14.2.3 cal_valp . . . 95
7.14.2.4 deja_converge . . . 96
7.14.2.5 dt_ray . . . 96
7.14.2.6 dt_ray_exp . . . 96
7.14.2.7 dt_ray_glob . . . 96
7.14.2.8 dt_ray_old . . . 96
7.14.2.9 EXP_RAY . . . 96
7.14.2.10 IMP_RAY . . . 96
7.14.2.11 nitemax_ray . . . 97
7.14.2.12 nitetot_ray . . . 97
7.14.2.13 nsy . . . 97
7.14.2.14 ordre_des_faces_pour_le_ray . . . 97
7.14.2.15 rad_trans_model . . . 97
7.14.2.16 rankine . . . 97
7.14.2.17 rho1 . . . 97
7.14.2.18 slope_type_ray . . . 97
7.14.2.19 T1 . . . 98
7.14.2.20 varrel_gaz . . . 98
7.14.2.21 varrel_ray . . . 98
7.15 Etat_GDS Module Reference . . . 99
7.15.1 Detailed Description . . . 99
7.15.2 Variable Documentation . . . 99
7.15.2.1 eS . . . 99
7.15.2.2 FxS . . . 99
7.15.2.3 pS . . . 99
7.15.2.4 rhoS . . . 99
7.15.2.5 uS . . . 100
7.16 gammas Module Reference . . . 101
7.16.1 Detailed Description . . . 102
7.16.2 Variable Documentation . . . 102
7.16.2.1 Cs_iso . . . 102
7.16.2.2 Cs_iso2 . . . 102
7.16.2.3 g1 . . . 102
7.16.2.4 g2 . . . 102
7.16.2.5 g3 . . . 102
7.16.2.6 g4 . . . 102

7.16.2.7 g5 . . . 103
7.16.2.8 g6 . . . 103
7.16.2.9 g7 . . . 103
7.16.2.10 g8 . . . 103
7.16.2.11 gamma . . . 103
7.16.2.12 gamma1 . . . 104
7.16.2.13 gamma2 . . . 104
7.16.2.14 mu_reac . . . 104
7.17 geom Module Reference . . . 105
7.17.1 Detailed Description . . . 106
7.17.2 Variable Documentation . . . 106
7.17.2.1 box_max . . . 106
7.17.2.2 box_min . . . 106
7.17.2.3 Cartesien . . . 106
7.17.2.4 Cylindrique . . . 106
7.17.2.5 ds . . . 106
7.17.2.6 dv . . . 107
7.17.2.7 dx . . . 107
7.17.2.8 dxc . . . 107
7.17.2.9 geom_dir . . . 107
7.17.2.10 geometrie . . . 107
7.17.2.11 Lbox . . . 108
7.17.2.12 nshift_gr . . . 108
7.17.2.13 shift_gr . . . 108
7.17.2.14 Spherique . . . 108
7.17.2.15 x . . . 108
7.17.2.16 x_glob . . . 108
7.17.2.17 xcloc . . . 109
7.18 gravity Module Reference . . . 110
7.18.1 Detailed Description . . . 110
7.18.2 Variable Documentation . . . 110
7.18.2.1 gravity_params . . . 110
7.18.2.2 gravity_type . . . 110
7.18.2.3 ISOLE . . . 110
7.18.2.4 phi . . . 110
7.18.2.5 phiold . . . 111

7.19 histo Module Reference . . . 112
7.19.1 Detailed Description . . . 113
7.19.2 Variable Documentation . . . 113
7.19.2.1 hN . . . 113
7.19.2.2 hNg . . . 113
7.19.2.3 hP . . . 113
7.19.2.4 hPg . . . 114
7.19.2.5 hT . . . 114
7.19.2.6 hTg . . . 114
7.19.2.7 lr_N . . . 114
7.19.2.8 lr_P . . . 114
7.19.2.9 lr_T . . . 114
7.19.2.10 N . . . 114
7.19.2.11 Nmax . . . 115
7.19.2.12 Nmin . . . 115
7.19.2.13 Npoint . . . 115
7.19.2.14 P . . . 115
7.19.2.15 Pmax . . . 115
7.19.2.16 Pmin . . . 115
7.19.2.17 r_N . . . 115
7.19.2.18 r_P . . . 116
7.19.2.19 r_T . . . 116
7.19.2.20 T . . . 116
7.19.2.21 Tmax . . . 116
7.19.2.22 Tmin . . . 116
7.20 mfilm Module Reference . . . 117
7.20.1 Detailed Description . . . 117
7.20.2 Variable Documentation . . . 117
7.20.2.1 dt_film . . . 117
7.20.2.2 FILM . . . 118
7.20.2.3 ifilm . . . 118
7.20.2.4 nl1 . . . 118
7.20.2.5 nl2 . . . 118
7.20.2.6 nl3 . . . 118
7.20.2.7 t_end . . . 118
7.20.2.8 t_film . . . 118

7.20.2.9 t_start . . . 119
7.21 para Module Reference . . . 120
7.21.1 Detailed Description . . . 121
7.21.2 Variable Documentation . . . 121
7.21.2.1 grid_cpu . . . 121
7.21.2.2 grid_pos . . . 121
7.21.2.3 lim_x . . . 121
7.21.2.4 lim_y . . . 121
7.21.2.5 lim_z . . . 121
7.21.2.6 mype . . . 121
7.21.2.7 ncpu . . . 122
7.21.2.8 ncpu_x . . . 122
7.21.2.9 ncpu_y . . . 122
7.21.2.10 ncpu_z . . . 122
7.21.2.11 voisin . . . 122
7.21.2.12 x_end . . . 122
7.21.2.13 x_end_cpu . . . 123
7.21.2.14 x_start . . . 123
7.21.2.15 x_start_cpu . . . 123
7.22 param_ini Module Reference . . . 124
7.22.1 Detailed Description . . . 126
7.22.2 Variable Documentation . . . 126
7.22.2.1 B0 . . . 126
7.22.2.2 cs1 . . . 126
7.22.2.3 d1 . . . 126
7.22.2.4 d2 . . . 126
7.22.2.5 E1 . . . 126
7.22.2.6 E2 . . . 126
7.22.2.7 E3 . . . 126
7.22.2.8 E4 . . . 127
7.22.2.9 p1 . . . 127
7.22.2.10 p2 . . . 127
7.22.2.11 p3 . . . 127
7.22.2.12 p4 . . . 127
7.22.2.13 rho1 . . . 127
7.22.2.14 rho2 . . . 127

7.22.2.15 rho3 . . . 127
7.22.2.16 rho4 . . . 127
7.22.2.17 T1 . . . 128
7.22.2.18 T2 . . . 128
7.22.2.19 T3 . . . 128
7.22.2.20 T4 . . . 128
7.22.2.21 Temperature_isotherme . . . 128
7.22.2.22 u1 . . . 128
7.22.2.23 u2 . . . 128
7.22.2.24 u3 . . . 128
7.22.2.25 u4 . . . 129
7.22.2.26 v1 . . . 129
7.22.2.27 v2 . . . 129
7.22.2.28 v3 . . . 129
7.22.2.29 v4 . . . 129
7.22.2.30 ww . . . 129
7.22.2.31 x1 . . . 129
7.22.2.32 x2 . . . 129
7.23 parameters Module Reference . . . 130
7.23.1 Detailed Description . . . 131
7.23.2 Variable Documentation . . . 131
7.23.2.1 N_vit . . . 131
7.23.2.2 Nbuf . . . 131
7.23.2.3 ndim . . . 131
7.23.2.4 nFx . . . 132
7.23.2.5 nvar . . . 132
7.23.2.6 nvar_ray . . . 132
7.23.2.7 nx . . . 132
7.23.2.8 nx_cpu . . . 133
7.23.2.9 nx_glob . . . 133
7.23.2.10 nx_glob_max . . . 133
7.23.2.11 nx_max . . . 133
7.23.2.12 nxmax . . . 133
7.23.2.13 nxmin . . . 133
7.23.2.14 Pi . . . 134
7.23.2.15 quatre_Pi . . . 134

7.23.2.16 riemann . . . 134
7.23.2.17 riemann2d . . . 134
7.23.2.18 slope_type . . . 134
7.23.2.19 smallc . . . 134
7.23.2.20 smallr . . . 134
7.24 prime Module Reference . . . 135
7.24.1 Detailed Description . . . 135
7.24.2 Variable Documentation . . . 135
7.24.2.1 Nprime . . . 135
7.24.2.2 Number . . . 135
7.24.2.3 Pdmax . . . 135
7.24.2.4 Pmax . . . 135
7.24.2.5 PrimeNumber . . . 135
7.25 rdwrt_h5 Module Reference . . . 136
7.25.1 Detailed Description . . . 136
7.26 solvers_hydro Module Reference . . . 137
7.26.1 Function/Subroutine Documentation . . . 138
7.26.1.1 cal_gamma . . . 138
7.26.1.2 guessp . . . 138
7.26.1.3 guessp_iso . . . 138
7.26.1.4 prefun . . . 139
7.26.1.5 prefun_iso . . . 139
7.26.1.6 sample . . . 139
7.26.1.7 sample_iso . . . 140
7.26.1.8 solver_acoustic . . . 140
7.26.1.9 solver_cc . . . 141
7.26.1.10 solver_eos . . . 141
7.26.1.11 solver_exact . . . 142
7.26.1.12 solver_g . . . 142
7.26.1.13 solver_iso . . . 143
7.26.1.14 solver_relax . . . 143
7.26.1.15 solver_relaxation . . . 144
7.26.1.16 starpu . . . 144
7.26.1.17 starpu_iso . . . 145
7.27 TabGmres Module Reference . . . 146
7.27.1 Detailed Description . . . 146

7.27.2 Variable Documentation . . . 147
7.27.2.1 b . . . 147
7.27.2.2 cntl . . . 147
7.27.2.3 icntl . . . 147
7.27.2.4 imin . . . 147
7.27.2.5 itermax . . . 147
7.27.2.6 jmin . . . 147
7.27.2.7 kmin . . . 147
7.27.2.8 work . . . 147
7.27.2.9 xin . . . 148
7.28 unites Module Reference . . . 149
7.28.1 Detailed Description . . . 152
7.28.2 Variable Documentation . . . 152
7.28.2.1 a_R . . . 152
7.28.2.2 an . . . 152
7.28.2.3 bar . . . 152
7.28.2.4 c . . . 152
7.28.2.5 c2 . . . 152
7.28.2.6 centimetre . . . 152
7.28.2.7 cm2 . . . 153
7.28.2.8 cm3 . . . 153
7.28.2.9 degre . . . 153
7.28.2.10 dyne . . . 153
7.28.2.11 erg . . . 153
7.28.2.12 eV . . . 153
7.28.2.13 G . . . 153
7.28.2.14 Gans . . . 154
7.28.2.15 Gauss . . . 154
7.28.2.16 gramme . . . 154
7.28.2.17 H0 . . . 154
7.28.2.18 hplanck . . . 154
7.28.2.19 joule . . . 154
7.28.2.20 kB . . . 154
7.28.2.21 Kelvin . . . 155
7.28.2.22 kg . . . 155
7.28.2.23 kms . . . 155

7.28.2.24 kpc . . . 155
7.28.2.25 Lsol . . . 155
7.28.2.26 m2 . . . 155
7.28.2.27 m3 . . . 155
7.28.2.28 Mans . . . 155
7.28.2.29 metre . . . 156
7.28.2.30 micron . . . 156
7.28.2.31 Mpc . . . 156
7.28.2.32 ms . . . 156
7.28.2.33 Msol . . . 156
7.28.2.34 mu0 . . . 156
7.28.2.35 ns . . . 156
7.28.2.36 Pascal . . . 157
7.28.2.37 pc . . . 157
7.28.2.38 Rsol . . . 157
7.28.2.39 seconde . . . 157
7.28.2.40 uma . . . 157
7.28.2.41 unitd . . . 157
7.28.2.42 unite . . . 157
7.28.2.43 unitf . . . 158
7.28.2.44 unitl . . . 158
7.28.2.45 unitm . . . 158
7.28.2.46 unitmom . . . 158
7.28.2.47 unitt . . . 158
7.29 unites_sortie Module Reference . . . 159
7.29.1 Detailed Description . . . 159
7.29.2 Variable Documentation . . . 159
7.29.2.1 u_dens . . . 159
7.29.2.2 u_E . . . 160
7.29.2.3 u_evol . . . 160
7.29.2.4 u_Fr . . . 160
7.29.2.5 u_L . . . 160
7.29.2.6 u_M . . . 160
7.29.2.7 u_Mom . . . 160
7.29.2.8 u_T . . . 160
7.29.2.9 u_Vit . . . 161

7.29.2.10 u_Vol . . . 161
7.30 valeurs_propres Module Reference . . . 162
7.30.1 Detailed Description . . . 162
7.30.2 Variable Documentation . . . 162
7.30.2.1 n_points . . . 162
7.30.2.2 valp . . . 162
7.31 var Module Reference . . . 163
7.31.1 Detailed Description . . . 163
7.31.2 Variable Documentation . . . 164
7.31.2.1 B . . . 164
7.31.2.2 E . . . 164
7.31.2.3 Fx . . . 164
7.31.2.4 rho . . . 164
7.31.2.5 rhou . . . 164
7.31.2.6 supp1 . . . 165
7.31.2.7 supp2 . . . 165
7.31.2.8 supp3 . . . 165
7.31.2.9 supp4 . . . 165
7.31.2.10 supp5 . . . 165
7.31.2.11 Tgaz . . . 165
7.32 var_loc Module Reference . . . 166
7.32.1 Detailed Description . . . 166
7.32.2 Variable Documentation . . . 166
7.32.2.1 Einloc . . . 166
7.32.2.2 Eloc . . . 167
7.32.2.3 Fxloc . . . 167
7.32.2.4 philoc . . . 167
7.32.2.5 Ploc . . . 167
7.32.2.6 rholoc . . . 167
7.32.2.7 rhouloc . . . 167
7.32.2.8 Tloc . . . 167
7.32.2.9 uloc . . . 168
7.32.2.10 xloc . . . 168
7.33 varold Module Reference . . . 169
7.33.1 Detailed Description . . . 169
7.33.2 Variable Documentation . . . 169

7.33.2.1 Bold . . . 169
7.33.2.2 Eold . . . 169
7.33.2.3 Fxold . . . 170
7.33.2.4 rhoold . . . 170
7.33.2.5 rhouold . . . 170
7.33.2.6 Tgazold . . . 170
7.34 varray Module Reference . . . 171
7.34.1 Detailed Description . . . 171
7.34.2 Variable Documentation . . . 171
7.34.2.1 Eray . . . 171
7.34.2.2 Eray_t . . . 172
7.34.2.3 Erayold . . . 172
7.34.2.4 ff_t . . . 172
7.34.2.5 Fray . . . 172
7.34.2.6 Fray_t . . . 172
7.34.2.7 Frayold . . . 172
7.34.2.8 kappa_abs_t . . . 172
7.34.2.9 sigma_diff . . . 173

8 Data Type Documentation **175**
8.1 cg__interface Interface Reference . . . 175
8.1.1 Detailed Description . . . 175
8.1.2 Member Function/Subroutine Documentation . . . 175
8.1.2.1 cmp_precond . . . 175
8.1.2.2 mat_prod . . . 175
8.2 conduction_imp_gc__interface Interface Reference . . . 177
8.2.1 Detailed Description . . . 177
8.2.2 Member Function/Subroutine Documentation . . . 177
8.2.2.1 cmp_precond_con . . . 177
8.2.2.2 mat_prod_con . . . 177
8.3 diffusion_imp__interface Interface Reference . . . 178
8.3.1 Detailed Description . . . 178
8.3.2 Member Function/Subroutine Documentation . . . 178
8.3.2.1 cmp_precond_dif . . . 178
8.3.2.2 mat_prod_dif . . . 178
8.4 gc__interface Interface Reference . . . 179
8.4.1 Detailed Description . . . 179

8.4.2 Member Function/Subroutine Documentation . . . 179
8.4.2.1 cmp_precond_gra . . . 179
8.4.2.2 mat_prod_gra . . . 179
8.5 TabGmres::interface Interface Reference . . . 180
8.5.1 Detailed Description . . . 180
8.5.2 Member Function/Subroutine Documentation . . . 180
8.5.2.1 gmres . . . 180

9 File Documentation **181**

9.1 algebra/cg.f90 File Reference . . . 181
9.1.1 Detailed Description . . . 181
9.1.2 Function Documentation . . . 181
9.1.2.1 cal_dif . . . 181
9.1.2.2 cg . . . 182
9.1.2.3 vec_prod . . . 182
9.2 conduction/conduction.f90 File Reference . . . 183
9.2.1 Detailed Description . . . 183
9.2.2 Function Documentation . . . 183
9.2.2.1 conduction_ex . . . 183
9.3 conduction/conduction_imp_gc.f90 File Reference . . . 184
9.3.1 Detailed Description . . . 184
9.3.2 Function Documentation . . . 184
9.3.2.1 cmp_precond_con . . . 184
9.3.2.2 conduction_imp_gc . . . 185
9.3.2.3 Mat_prod_con . . . 185
9.4 conduction/conduction_step.f90 File Reference . . . 186
9.4.1 Detailed Description . . . 186
9.4.2 Function Documentation . . . 186
9.4.2.1 conduction_step . . . 186
9.5 conduction/init_conduction.f90 File Reference . . . 187
9.5.1 Detailed Description . . . 187
9.5.2 Function Documentation . . . 187
9.5.2.1 init_conduction . . . 187
9.6 conduction/kappa_con.f90 File Reference . . . 188
9.6.1 Detailed Description . . . 188
9.6.2 Function Documentation . . . 188
9.6.2.1 cal_kappa_con . . . 188

9.7 conduction/modules_con.f90 File Reference . . . 189
9.7.1 Detailed Description . . . 189
9.8 conduction/pasdt_con.f90 File Reference . . . 190
9.8.1 Detailed Description . . . 190
9.8.2 Function Documentation . . . 190
9.8.2.1 pasdt_con . . . 190
9.9 gravity/gc.f90 File Reference . . . 191
9.9.1 Detailed Description . . . 191
9.9.2 Function Documentation . . . 191
9.9.2.1 cmp_precond_gra . . . 191
9.9.2.2 gc . . . 191
9.9.2.3 mat_prod_gra . . . 192
9.10 gravity/grav_predictor.f90 File Reference . . . 193
9.10.1 Detailed Description . . . 193
9.10.2 Function Documentation . . . 193
9.10.2.1 grav_predictor . . . 193
9.11 gravity/init_gravity.f90 File Reference . . . 194
9.11.1 Detailed Description . . . 194
9.11.2 Function Documentation . . . 194
9.11.2.1 init_gravity . . . 194
9.12 gravity/limites_gra.f90 File Reference . . . 195
9.12.1 Detailed Description . . . 195
9.12.2 Function Documentation . . . 195
9.12.2.1 cal_barycentre . . . 195
9.12.2.2 limites_gra . . . 195
9.13 gravity/modules_gra.f90 File Reference . . . 197
9.13.1 Detailed Description . . . 197
9.13.2 Function Documentation . . . 197
9.13.2.1 allocate_array_gra . . . 197
9.14 gravity/pasdt_grav.f90 File Reference . . . 199
9.14.1 Detailed Description . . . 199
9.14.2 Function Documentation . . . 199
9.14.2.1 pasdt_grav . . . 199
9.15 gravity/poisson.f90 File Reference . . . 200
9.15.1 Detailed Description . . . 200
9.15.2 Function Documentation . . . 200

9.15.2.1 cst_gravity . . . 200
9.15.2.2 poisson . . . 200
9.15.2.3 pt_mass . . . 201
9.16 gravity/update_gravity.f90 File Reference . . . 202
9.16.1 Detailed Description . . . 202
9.16.2 Function Documentation . . . 202
9.16.2.1 update_gravity . . . 202
9.17 hdf/film_hdf.f90 File Reference . . . 203
9.17.1 Detailed Description . . . 203
9.17.2 Function Documentation . . . 203
9.17.2.1 film_hdf . . . 203
9.18 hdf/rd_restart_h5.f90 File Reference . . . 204
9.18.1 Detailed Description . . . 204
9.18.2 Function Documentation . . . 204
9.18.2.1 rd_restart_h5 . . . 204
9.19 hdf/rdwrt_h5.f90 File Reference . . . 205
9.19.1 Detailed Description . . . 205
9.20 hdf/utils_h5.f90 File Reference . . . 206
9.20.1 Detailed Description . . . 206
9.20.2 Function Documentation . . . 206
9.20.2.1 output_data . . . 206
9.21 hdf/wrt_film_h5.f90 File Reference . . . 207
9.21.1 Detailed Description . . . 207
9.21.2 Function Documentation . . . 207
9.21.2.1 wrt_film_h5 . . . 207
9.22 hdf/wrt_main_h5.f90 File Reference . . . 208
9.22.1 Detailed Description . . . 208
9.22.2 Function Documentation . . . 208
9.22.2.1 wrt_main_h5 . . . 208
9.23 hdf/wrt_output_h5.f90 File Reference . . . 209
9.23.1 Detailed Description . . . 209
9.23.2 Function Documentation . . . 209
9.23.2.1 wrt_output_h5 . . . 209
9.24 hdf/wrt_restart_h5.f90 File Reference . . . 210
9.24.1 Detailed Description . . . 210
9.24.2 Function Documentation . . . 210

9.24.2.1 wrt_restart_h5 . . . 210
9.25 hydro/eos.f90 File Reference . . . 211
9.25.1 Detailed Description . . . 211
9.25.2 Function Documentation . . . 211
9.25.2.1 Energy . . . 211
9.25.2.2 heat_capacity . . . 212
9.25.2.3 init_eos . . . 212
9.25.2.4 Pressure . . . 213
9.25.2.5 Sound_speed . . . 213
9.25.2.6 Temperature . . . 213
9.26 hydro/evol_hydro.f90 File Reference . . . 214
9.26.1 Detailed Description . . . 214
9.26.2 Function Documentation . . . 214
9.26.2.1 evol_hydro . . . 214
9.27 hydro/fill_new_arrays.f90 File Reference . . . 216
9.27.1 Detailed Description . . . 216
9.27.2 Function Documentation . . . 216
9.27.2.1 fill_new_arrays . . . 216
9.28 hydro/histogramme.f90 File Reference . . . 217
9.28.1 Detailed Description . . . 218
9.28.2 Function Documentation . . . 219
9.28.2.1 cal_histogramme . . . 219
9.28.2.2 init_histo . . . 219
9.29 hydro/hydro_step.f90 File Reference . . . 220
9.29.1 Detailed Description . . . 220
9.29.2 Function Documentation . . . 220
9.29.2.1 hydro_step . . . 220
9.30 hydro/monitoring.f90 File Reference . . . 223
9.30.1 Detailed Description . . . 223
9.30.2 Function Documentation . . . 223
9.30.2.1 monitoring . . . 223
9.31 hydro/pasdt_hydro.f90 File Reference . . . 224
9.31.1 Detailed Description . . . 224
9.31.2 Function Documentation . . . 224
9.31.2.1 pasdt_hydro . . . 224
9.32 hydro/rh.f90 File Reference . . . 225

9.32.1 Detailed Description . . . 225
9.32.2 Function Documentation . . . 225
9.32.2.1 f . . . 225
9.32.2.2 rh . . . 225
9.33 hydro/solvers.f90 File Reference . . . 226
9.33.1 Detailed Description . . . 227
9.34 init/init.f90 File Reference . . . 228
9.34.1 Detailed Description . . . 228
9.34.2 Function Documentation . . . 228
9.34.2.1 dimensions . . . 228
9.34.2.2 init . . . 228
9.35 init/init_out.f90 File Reference . . . 229
9.35.1 Detailed Description . . . 229
9.35.2 Function Documentation . . . 229
9.35.2.1 init_out . . . 229
9.36 main/aff.f90 File Reference . . . 230
9.36.1 Detailed Description . . . 230
9.36.2 Function Documentation . . . 230
9.36.2.1 aff_new . . . 230
9.36.2.2 convtoasc . . . 231
9.36.2.3 make_name . . . 231
9.36.2.4 mk_dir_out . . . 232
9.36.2.5 print_configuration . . . 232
9.36.2.6 sortie_ecran . . . 232
9.37 main/allocate_array.f90 File Reference . . . 233
9.37.1 Detailed Description . . . 233
9.37.2 Function Documentation . . . 233
9.37.2.1 allocate_array . . . 233
9.38 main/check_flags.f90 File Reference . . . 234
9.38.1 Detailed Description . . . 234
9.38.2 Function Documentation . . . 234
9.38.2.1 check_flags . . . 234
9.39 main/check_size.f90 File Reference . . . 235
9.39.1 Detailed Description . . . 235
9.39.2 Function Documentation . . . 235
9.39.2.1 check_size . . . 235

9.40 main/cl_ana.f90 File Reference . . . 236
9.40.1 Detailed Description . . . 236
9.40.2 Function Documentation . . . 236
9.40.2.1 cl_ana . . . 236
9.40.2.2 cl_ana_gen . . . 236
9.41 main/cpu.f90 File Reference . . . 238
9.41.1 Detailed Description . . . 238
9.41.2 Function Documentation . . . 238
9.41.2.1 cpu_ . . . 238
9.42 main/film.f90 File Reference . . . 239
9.42.1 Detailed Description . . . 240
9.42.2 Function Documentation . . . 240
9.42.2.1 film_ . . . 240
9.42.2.2 init_film . . . 240
9.43 main/geom.f90 File Reference . . . 241
9.43.1 Detailed Description . . . 241
9.43.2 Function Documentation . . . 241
9.43.2.1 cal_geom . . . 241
9.43.2.2 def_geom . . . 242
9.44 main/init_phys.f90 File Reference . . . 243
9.44.1 Detailed Description . . . 243
9.44.2 Function Documentation . . . 243
9.44.2.1 init_phys . . . 243
9.45 main/limites.f90 File Reference . . . 245
9.45.1 Detailed Description . . . 245
9.45.2 Function Documentation . . . 245
9.45.2.1 limites . . . 245
9.46 main/limites_generique.f90 File Reference . . . 246
9.46.1 Detailed Description . . . 246
9.46.2 Function Documentation . . . 246
9.46.2.1 limites_gen . . . 246
9.47 main/main.f90 File Reference . . . 247
9.47.1 Detailed Description . . . 247
9.47.2 Function Documentation . . . 247
9.47.2.1 Arret . . . 247
9.47.2.2 close_program . . . 248

9.47.2.3 compute_cputime . . . 250
9.47.2.4 heracles . . . 250
9.47.2.5 MPI_Wtime . . . 252
9.47.2.6 output . . . 252
9.47.2.7 tremain . . . 253
9.48 main/modules.f90 File Reference . . . 254
9.48.1 Detailed Description . . . 271
9.49 main/para.f90 File Reference . . . 272
9.49.1 Detailed Description . . . 277
9.49.2 Function Documentation . . . 277
9.49.2.1 allocate_array_com . . . 277
9.49.2.2 communications . . . 277
9.49.2.3 init_para . . . 278
9.50 main/para_generique.f90 File Reference . . . 280
9.50.1 Detailed Description . . . 284
9.50.2 Function Documentation . . . 284
9.50.2.1 allocate_array_com_gen . . . 284
9.50.2.2 communications_gen . . . 285
9.51 main/pasdt.f90 File Reference . . . 286
9.51.1 Detailed Description . . . 286
9.51.2 Function Documentation . . . 286
9.51.2.1 pasdt . . . 286
9.52 main/prime.f90 File Reference . . . 287
9.52.1 Detailed Description . . . 287
9.52.2 Function Documentation . . . 287
9.52.2.1 CpuDec . . . 287
9.52.2.2 find_prime . . . 288
9.52.2.3 PrimeDec . . . 288
9.53 main/read_params.f90 File Reference . . . 289
9.53.1 Detailed Description . . . 289
9.53.2 Function Documentation . . . 289
9.53.2.1 default_params . . . 289
9.53.2.2 read_params . . . 289
9.54 main/restart.f90 File Reference . . . 291
9.54.1 Detailed Description . . . 291
9.54.2 Function Documentation . . . 291

9.54.2.1 restart_new . . . 291
9.55 main/set_units_out.f90 File Reference . . . 292
9.55.1 Detailed Description . . . 292
9.55.2 Function Documentation . . . 292
9.55.2.1 set_units_out . . . 292
9.56 main/slopes.f90 File Reference . . . 293
9.56.1 Detailed Description . . . 293
9.56.2 Function Documentation . . . 293
9.56.2.1 minmod . . . 293
9.56.2.2 moncen . . . 293
9.56.2.3 moyharm . . . 294
9.56.2.4 slopes . . . 294
9.56.2.5 vavl . . . 295
9.57 main/user_init.f90 File Reference . . . 296
9.57.1 Detailed Description . . . 296
9.57.2 Function Documentation . . . 296
9.57.2.1 user_init . . . 296
9.58 main/user_output.f90 File Reference . . . 297
9.58.1 Detailed Description . . . 297
9.58.2 Function Documentation . . . 297
9.58.2.1 user_output . . . 297
9.59 main/user_step.f90 File Reference . . . 298
9.59.1 Detailed Description . . . 298
9.59.2 Function Documentation . . . 298
9.59.2.1 user_step . . . 298
9.60 mhd/ctoprim.f90 File Reference . . . 299
9.60.1 Detailed Description . . . 299
9.60.2 Function Documentation . . . 299
9.60.2.1 ctoprim . . . 299
9.61 mhd/evol_mhd.f90 File Reference . . . 300
9.61.1 Detailed Description . . . 300
9.61.2 Function Documentation . . . 300
9.61.2.1 evol_mhd . . . 300
9.62 mhd/godunov_utils.f90 File Reference . . . 302
9.62.1 Detailed Description . . . 303
9.62.2 Function Documentation . . . 303

9.62.2.1 athena_roe . . . 303
9.62.2.2 eigen_cons . . . 303
9.62.2.3 eigenvalues . . . 304
9.62.2.4 find_mhd_flux . . . 304
9.62.2.5 find_mhd_flux2 . . . 305
9.62.2.6 find_speed_alfven . . . 305
9.62.2.7 find_speed_fast . . . 305
9.62.2.8 find_speed_info . . . 306
9.62.2.9 hll . . . 306
9.62.2.10 hlld . . . 306
9.62.2.11 hydro_acoustic . . . 307
9.62.2.12 lax_friedrich . . . 307
9.62.2.13 upwind . . . 308
9.63 mhd/pasdt_mhd.f90 File Reference . . . 309
9.63.1 Detailed Description . . . 309
9.63.2 Function Documentation . . . 309
9.63.2.1 pasdt_mhd . . . 309
9.64 mhd/trace.f90 File Reference . . . 310
9.64.1 Detailed Description . . . 310
9.64.2 Function Documentation . . . 310
9.64.2.1 trace1d . . . 310
9.64.2.2 trace2d . . . 311
9.64.2.3 trace3d . . . 311
9.65 mhd/umuscl.f90 File Reference . . . 312
9.65.1 Detailed Description . . . 312
9.65.2 Function Documentation . . . 312
9.65.2.1 cmp_mag_flx . . . 312
9.65.2.2 cmpflxm . . . 313
9.65.2.3 fast_mhd_speed . . . 314
9.65.2.4 uslope . . . 314
9.66 mhd/update.f90 File Reference . . . 316
9.66.1 Detailed Description . . . 316
9.66.2 Function Documentation . . . 316
9.66.2.1 update . . . 316
9.67 rad_transfer/exp_comobile_ray.f90 File Reference . . . 317
9.67.1 Detailed Description . . . 317

9.67.2 Function Documentation . . . 317
9.67.2.1 exp_comobile_ray . . . 317
9.68 rad_transfer/exp_source_ray.f90 File Reference . . . 318
9.68.1 Detailed Description . . . 318
9.68.2 Function Documentation . . . 318
9.68.2.1 exp_source_ray . . . 318
9.69 rad_transfer/fld/diffusion.f90 File Reference . . . 320
9.69.1 Detailed Description . . . 320
9.69.2 Function Documentation . . . 320
9.69.2.1 diffusion_ex . . . 320
9.70 rad_transfer/fld/diffusion_imp_gc.f90 File Reference . . . 321
9.70.1 Detailed Description . . . 321
9.70.2 Function Documentation . . . 321
9.70.2.1 cmp_precond_dif . . . 321
9.70.2.2 diffusion_imp . . . 322
9.70.2.3 mat_prod_dif . . . 322
9.71 rad_transfer/fld/diffusion_step.f90 File Reference . . . 323
9.71.1 Detailed Description . . . 323
9.71.2 Function Documentation . . . 323
9.71.2.1 diffusion_step . . . 323
9.72 rad_transfer/fld/init_diffusion.f90 File Reference . . . 324
9.72.1 Detailed Description . . . 324
9.72.2 Function Documentation . . . 324
9.72.2.1 allocate_array_dif . . . 324
9.72.2.2 derive_source_dif . . . 324
9.72.2.3 init_diffusion . . . 325
9.72.2.4 source_dif . . . 325
9.73 rad_transfer/fld/kappa_dif.f90 File Reference . . . 326
9.73.1 Detailed Description . . . 326
9.73.2 Function Documentation . . . 326
9.73.2.1 cal_kappa_dif . . . 326
9.74 rad_transfer/fld/limites_dif.f90 File Reference . . . 327
9.74.1 Detailed Description . . . 327
9.74.2 Function Documentation . . . 327
9.74.2.1 limites_dif . . . 327
9.75 rad_transfer/fld/modules_dif.f90 File Reference . . . 328

9.75.1 Detailed Description . . . 329
9.76 rad_transfer/fld/para_dif.f90 File Reference . . . 330
9.76.1 Detailed Description . . . 330
9.76.2 Function Documentation . . . 330
9.76.2.1 communications_dif . . . 330
9.77 rad_transfer/fld/pasdt_dif.f90 File Reference . . . 331
9.77.1 Detailed Description . . . 331
9.77.2 Function Documentation . . . 331
9.77.2.1 pasdt_dif . . . 331
9.78 rad_transfer/m1/allocate_array_ray.f90 File Reference . . . 332
9.78.1 Detailed Description . . . 332
9.78.2 Function Documentation . . . 332
9.78.2.1 allocate_array_ray . . . 332
9.79 rad_transfer/m1/cal_b3d.f90 File Reference . . . 333
9.79.1 Detailed Description . . . 333
9.79.2 Function Documentation . . . 333
9.79.2.1 cal_b3d . . . 333
9.80 rad_transfer/m1/cal_x3d.f90 File Reference . . . 335
9.80.1 Detailed Description . . . 335
9.80.2 Function Documentation . . . 335
9.80.2.1 allocate_gmres . . . 335
9.80.2.2 cal_x3d . . . 335
9.80.2.3 calmoinsun_x3d . . . 336
9.81 rad_transfer/m1/cl_ana_ray.f90 File Reference . . . 337
9.81.1 Detailed Description . . . 337
9.81.2 Function Documentation . . . 337
9.81.2.1 cl_ana_gen_ray . . . 337
9.81.2.2 cl_ana_ray . . . 337
9.82 rad_transfer/m1/evol_ray.f90 File Reference . . . 339
9.82.1 Detailed Description . . . 339
9.82.2 Function Documentation . . . 339
9.82.2.1 evol_ray . . . 339
9.82.2.2 pente_minmod . . . 340
9.83 rad_transfer/m1/explicite.f90 File Reference . . . 341
9.83.1 Detailed Description . . . 341
9.83.2 Function Documentation . . . 341

9.83.2.1 explicite . . . 341
9.84 rad_transfer/m1/gmres.f90 File Reference . . . 343
9.84.1 Detailed Description . . . 343
9.84.2 Function Documentation . . . 343
9.84.2.1 gmres . . . 343
9.85 rad_transfer/m1/gs.f90 File Reference . . . 345
9.85.1 Detailed Description . . . 345
9.85.2 Function Documentation . . . 345
9.85.2.1 cal_Dedd . . . 345
9.85.2.2 det_3x3 . . . 346
9.85.2.3 det_4x4 . . . 346
9.85.2.4 essai_inv5x5 . . . 346
9.85.2.5 gs_mat . . . 346
9.86 rad_transfer/m1/init_ray.f90 File Reference . . . 348
9.86.1 Detailed Description . . . 348
9.86.2 Function Documentation . . . 348
9.86.2.1 init_ray . . . 348
9.87 rad_transfer/m1/initvargmres.f90 File Reference . . . 350
9.87.1 Detailed Description . . . 350
9.87.2 Function Documentation . . . 350
9.87.2.1 initVarGmres . . . 350
9.88 rad_transfer/m1/inversion_gmres.f90 File Reference . . . 351
9.88.1 Detailed Description . . . 351
9.88.2 Function Documentation . . . 351
9.88.2.1 inversion_gmres . . . 351
9.89 rad_transfer/m1/kappa_ray.f90 File Reference . . . 352
9.89.1 Detailed Description . . . 352
9.89.2 Function Documentation . . . 352
9.89.2.1 cal_kappa . . . 352
9.89.2.2 cal_sigmadiff . . . 352
9.90 rad_transfer/m1/limites_ray.f90 File Reference . . . 354
9.90.1 Detailed Description . . . 354
9.90.2 Function Documentation . . . 354
9.90.2.1 limites_ray . . . 354
9.91 rad_transfer/m1/mat_prod3d.f90 File Reference . . . 356
9.91.1 Detailed Description . . . 356

9.91.2 Function Documentation . . . 356
9.91.2.1 limites_xin . . . 356
9.91.2.2 mat_prod3d . . . 356
9.92 rad_transfer/m1/modules_ray.f90 File Reference . . . 358
9.92.1 Detailed Description . . . 361
9.93 rad_transfer/m1/para_ray.f90 File Reference . . . 362
9.93.1 Detailed Description . . . 364
9.93.2 Function Documentation . . . 364
9.93.2.1 allocate_array_com_ray . . . 364
9.93.2.2 communications_ray . . . 364
9.94 rad_transfer/m1/pasdt_ray.f90 File Reference . . . 366
9.94.1 Detailed Description . . . 366
9.94.2 Function Documentation . . . 366
9.94.2.1 pasdt_ray . . . 366
9.94.2.2 pasdt_ray_exp . . . 366
9.95 rad_transfer/m1/ray_step.f90 File Reference . . . 368
9.95.1 Detailed Description . . . 368
9.95.2 Function Documentation . . . 368
9.95.2.1 ray_step . . . 368
9.96 rad_transfer/m1/sources_ray.f90 File Reference . . . 370
9.96.1 Detailed Description . . . 370
9.96.2 Function Documentation . . . 370
9.96.2.1 derive_source . . . 370
9.96.2.2 source . . . 370
9.97 rad_transfer/m1/valp.f90 File Reference . . . 371
9.97.1 Detailed Description . . . 371
9.97.2 Function Documentation . . . 371
9.97.2.1 interpol_valp . . . 371
9.97.2.2 read_valp . . . 371

Chapter 1

Main Page

Hydrodynamique **E**ulerienne **RA**diative en **C**oordonnées cy**L**indriques **E**t **S**phériques

Copyright ©CEA & Edouard Audit

(edouard.audit@cea.fr)

30/06/2010

This software is governed by the CeCILL license

(see README and LICENSE files for details)

Main contributors to the code:

- Code Architecture : Edouard Audit
- Parallelization : Edouard Audit
- Hydrodynamics : Edouard Audit
- Radiative transfer : Matthias González, Edouard Audit & Neil Vaytet
- MHD : Sebastien Fromang, Patrick Hennebelle & Romain Teyssier
- Gravity : Pascal Tremblin
- HDF5 output : Bruno Thooris

Chapter 2

Directory Hierarchy

2.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

algebra 13
conduction 14
gravity 16
hdf 17
hydro 18
init 19
main 22
mhd 24
rad_transfer 25
  fld 15
  m1 20

Chapter 3

Modules Index

3.1 Modules List

Here is a list of all modules with brief descriptions:

cellules (Contains ?) . . . 27
cl (Contains the variables used to define the boundary conditions) . . . 31
cl_con (Contains the variables used for the conduction boundary conditions) . . . 34
cl_dif (Contains the variables for the FLD boundary conditions) . . . 35
cl_ray (Contains the boundary conditions variables for the M1 radiative transfer) . . . 36
communication (Contains variables arrays used to communicate ghost cell data across cpus) . . . 38
communication_gen (Contains variables arrays used to communicate ghost cell data across cpus) 51
communication_ray (Contains the communication arrays for the M1 radiative transfer) . . . 64
conduction (Contains the variables used for the thermal conduction) . . . 71
const (Contains useful constants) . . . 73
cputime (Contains variables used to compute the cpu time spent is each part of the code) . . . 76
diffusion (Contains the variables for the FLD) . . . 78
divers (Contains various variables) . . . 81
divers_ray (Contains various variables for the M1 radiative transfer) . . . 94
Etat_GDS (Contains the interface values arrays) . . . 99
gammas (Contains the variables determining the heat capacity ratio and thermodynamic relations) 101
geom (Contains all the grid variables and coordinates) . . . 105
gravity (Contains all the gravity variables) . . . 110
histo (Contains variables for histogram computations) . . . 112
mfilm (Contains variables used to create movies) . . . 117
para (Contains variables defining the MPI parameters of the simulation) . . . 120
param_ini (Contains variables used for initialisation of the simulation) . . . 124
parameters (Holds all the general simulation parameters) . . . 130
prime (Contains variables used for prime number computations) . . . 135
rdwrt_h5 (Contains reading and writing hdf5 subroutines) . . . 136
solvers_hydro . . . 137
TabGmres (Contains workspace arrays for GMRES) . . . 146
unites (This routine calls the relevant output routines to write binary or HDF5 files) . . . 149
unites_sortie (Contains the units for output file dumping) . . . 159
valeurs_propres (Contains the array to hold the tabulated M1 eignevalues) . . . 162
var (Contains all the main variable arrays) . . . 163
var_loc (Contains the local variables) . . . 166
varold (Contains all the old (non-updated) variable arrays) . . . 169

varray (Contains the M1 radiative transfer variables) 171

Chapter 4

Data Type Index

4.1 Data Types List

Here are the data types with brief descriptions:

cg__interface 175
conduction_imp_gc__interface 177
diffusion_imp__interface 178
gc__interface 179
TabGmres::interface (GMRES interfaces) 180

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

algebra/cg.f90 (Contains subroutines cg(), vec_prod() and cal_dif()) . . . 181
conduction/conduction.f90 (Contains subroutine conduction_ex()) . . . 183
conduction/conduction_imp_gc.f90 (Contains subroutines conduction_imp_gc(), Mat_prod_con() and cmp_precond_con()) . . . 184
conduction/conduction_step.f90 (Contains subroutine conduction_step()) . . . 186
conduction/init_conduction.f90 (Contains subroutine init_conduction()) . . . 187
conduction/kappa_con.f90 (Contains function cal_kappa_con()) . . . 188
conduction/modules_con.f90 (Contains modules cl_con and conduction) . . . 189
conduction/pasdt_con.f90 (Contains subroutine pasdt_con()) . . . 190
gravity/gc.f90 (Contains subroutines gc(), mat_prod_gra() and cmp_precond_gra()) . . . 191
gravity/grav_predictor.f90 (Contains subroutine grav_predictor()) . . . 193
gravity/init_gravity.f90 (Contains subroutine init_gravity()) . . . 194
gravity/limites_gra.f90 (Contains subroutines limites_gra() and cal_barycentre()) . . . 195
gravity/modules_gra.f90 (Contains module gravity and subroutine allocate_array_gra()) . . . 197
gravity/pasdt_grav.f90 (Contains subroutine pasdt_grav()) . . . 199
gravity/poisson.f90 (Contains subroutines poisson(), pt_mass() and cst_gravity()) . . . 200
gravity/update_gravity.f90 (Contains subroutine update_gravity()) . . . 202
hdf/film_hdf.f90 (Contains subroutine film_hdf()) . . . 203
hdf/rd_restart_h5.f90 (Contains subroutine rd_restart_h5()) . . . 204
hdf/rdwrt_h5.f90 (Contains module rdwrt_h5 and subroutines dump_1d_array_int_h5(), dump_1d_array_h5(), dump_3d_array_h5(), dump_4d_array_h5(), dump_1D_array_string_h5(), get_1d_array_int_h5(), get_1d_array_h5(), get_3d_array_h5() and get_4d_array_h5(), get_1D_array_string_h5()) . . . 205
hdf/utils_h5.f90 (Contains subroutines output_data(), get_filename(), get_dataexist() and get_nl()) . . . 206
hdf/wrt_film_h5.f90 (Contains subroutine wrt_film_h5()) . . . 207
hdf/wrt_main_h5.f90 (Contains subroutine wrt_main_h5()) . . . 208
hdf/wrt_output_h5.f90 (Contains subroutine wrt_output_h5()) . . . 209
hdf/wrt_restart_h5.f90 (Contains subroutine wrt_restart_h5()) . . . 210
hydro/eos.f90 (Contains subroutines init_eos(), pressure(), temperature(), sound_speed(), energy() and heat_capacity()) . . . 211
hydro/evol_hydro.f90 (Contains subroutines evol_hydro() and pente_minmod()) . . . 214
hydro/fill_new_arrays.f90 (Contains subroutine fill_new_arrays()) . . . 216

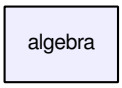
hydro/histogramme.f90 (Contains module histo and subroutines init_histo() and cal_-histogramme()) . . . 217
hydro/hydro_step.f90 (Contains subroutine hydro_step()) . . . 220
hydro/monitoring.f90 (Contains subroutine monitoring()) . . . 223
hydro/pasdt_hydro.f90 (Contains subroutine pasdt_hydro()) . . . 224
hydro/rh.f90 (Contains subroutine rh() and function f()) . . . 225
hydro/solvers.f90 (Contains subroutines solver_relax(), solver_relaxation(), solver_exact(), solver_cc(), solver_acoustic(), solver_eos(), solver_g(), solver_iso(), starpu(), guessp(), prefun(), sample(), starpu_iso(), guessp_iso(), prefun_iso(), sample_iso() and cal_-gamma()) . . . 226
init/init.f90 (Contains subroutines init() and dimensions()) . . . 228
init/init_out.f90 (Contains subroutine init_out()) . . . 229
main/aff.f90 (Contains subroutines aff_new(), make_name(), mk_dir_out(), convtoasc() and sortie_ecran()) . . . 230
main/allocate_array.f90 (Contains subroutine allocate_array()) . . . 233
main/check_flags.f90 (Contains subroutine check_flags()) . . . 234
main/check_size.f90 (Contains subroutine check_size()) . . . 235
main/cl_ana.f90 (Contains subroutines cl_ana() and cl_ana_gen()) . . . 236
main/cpu.f90 (Contains subroutine cpu_()) . . . 238
main/film.f90 (Contains module mfilm and subroutines film_() and init_film()) . . . 239
main/geom.f90 (Contains subroutines def_geom() and cal_geom()) . . . 241
main/init_phys.f90 (Contains subroutine init_phys()) . . . 243
main/limites.f90 (Contains subroutine limites()) . . . 245
main/limites_generique.f90 (Contains subroutine limites_gen()) . . . 246
main/main.f90 (Contains main program HERACLES and subroutines compute_cputime(), out-put() and close_program() and functions Arret(), MPI_Wtime(), tremain()) . . . 247
main/modules.f90 (Contains modules parameters, var_loc, geom, var, varold, etat_gds, unites, unites_sortie, divers, param_ini, cl, gammas, cellules, const and cputime) . . . 254
main/para.f90 (Contains modules communication and para and subroutines init_para(), allocate_-array_com(), communications(), com_1D(), com_2D() and com_3D()) . . . 272
main/para_generique.f90 (Contains module communication_gen and subroutines allocate_-array_com_gen(), communications_gen(), com_1D_gen(), com_2D_gen() and com_-3D()) . . . 280
main/pasdt.f90 (Contains subroutine pasdt()) . . . 286
main/prime.f90 (Contains module prime and subroutines find_prime(), primedec() and cpudec()) 287
main/read_params.f90 (Contains subroutines read_params() and default_params()) . . . 289
main/restart.f90 (Contains subroutine restart_new()) . . . 291
main/set_units_out.f90 (Contains subroutine set_units_out()) . . . 292
main/slopes.f90 (Contains subroutine slopes() and functions minmod(), moncen(), moyharm() and vavl()) . . . 293
main/user_init.f90 (Contains subroutine user_init()) . . . 296
main/user_output.f90 (Contains subroutine user_output()) . . . 297
main/user_step.f90 (Contains subroutine user_step()) . . . 298
mhd/ctoprim.f90 (Contains subroutine ctoprim()) . . . 299
mhd/evol_mhd.f90 (Contains subroutine evol_mhd()) . . . 300
mhd/godunov_utils.f90 (Subroutines upwind(), lax_friedrich(), hll(), hlld(), find_mhd_flux(), find_mhd_flux2(), find_speed_info(), find_speed_fast(), find_speed_alfven(), hydro_-acoustic(), athena_roe(), eigenvalues() and eigen_cons()) . . . 302
mhd/pasdt_mhd.f90 (Subroutine pasdt_mhd()) . . . 309
mhd/trace.f90 (Subroutines trace1d(), trace2d() and trace3d()) . . . 310
mhd/umuscl.f90 (Subroutines cmpflxm(), cmp_mag_flx(), fast_mhd_speed() and uslope()) . . . 312
mhd/update.f90 (Subroutine update()) . . . 316
rad_transfer/exp_comobile_ray.f90 (Contains subroutine exp_comobile_ray()) . . . 317
rad_transfer/exp_source_ray.f90 (Contains subroutine exp_source_ray()) . . . 318

rad_transfer/fld/diffusion.f90 (Contains subroutine diffusion_ex()) 320
rad_transfer/fld/diffusion_imp_gc.f90 (Contains subroutines diffusion_imp(), mat_prod_dif() and cmp_precond_dif()) 321
rad_transfer/fld/diffusion_step.f90 (Contains subroutine diffusion_step()) 323
rad_transfer/fld/init_diffusion.f90 (Contains subroutines init_diffusion() and allocate_array_dif() and functions source_dif() and derive_source_dif()) 324
rad_transfer/fld/kappa_dif.f90 (Contains subroutine cal_kappa_dif()) 326
rad_transfer/fld/limites_dif.f90 (Contains subroutine limites_dif()) 327
rad_transfer/fld/modules_dif.f90 (Contains modules cl_dif and diffusion) 328
rad_transfer/fld/para_dif.f90 (Contains subroutine communications_dif()) 330
rad_transfer/fld/pasdt_dif.f90 (Contains subroutine pasdt_dif()) 331
rad_transfer/m1/allocate_array_ray.f90 (Contains subroutine allocate_array_ray()) 332
rad_transfer/m1/cal_b3d.f90 (Contains subroutine cal_b3d()) 333
rad_transfer/m1/cal_x3d.f90 (Contains subroutines cal_x3d(), calmoinsun_x3d() and allocate_gmres()) 335
rad_transfer/m1/cl_ana_ray.f90 (Contains subroutines cl_ana_ray() and cl_ana_gen_ray()) . . . 337
rad_transfer/m1/evol_ray.f90 (Contains subroutine evol_ray()) 339
rad_transfer/m1/explicite.f90 (Contains subroutine explicite()) 341
rad_transfer/m1/gmres.f90 (Contains subroutine gmres()) 343
rad_transfer/m1/gs.f90 (Contains subroutines gs_mat(), cal_Dedd() and essai_inv5x5 and functions det_3x3() and det_4x4()) 345
rad_transfer/m1/init_ray.f90 (Contains subroutine init_ray()) 348
rad_transfer/m1/initvargmres.f90 (Contains subroutine initVarGmres()) 350
rad_transfer/m1/inversion_gmres.f90 (Contains subroutine inversion_gmres()) 351
rad_transfer/m1/kappa_ray.f90 (Contains subroutines cal_kappa() and cal_sigmadiff()) 352
rad_transfer/m1/limites_ray.f90 (Contains subroutine limites_ray()) 354
rad_transfer/m1/mat_prod3d.f90 (Contains subroutines mat_prod3d(), com_1D_xin(), com_2D_xin(), com_3D_xin() and limites_xin()) 356
rad_transfer/m1/modules_ray.f90 (Contains modules varray, divers_ray, cl_ray, valeurs_propres, TabGmres, VarGmres and interfaces) 358
rad_transfer/m1/para_ray.f90 (Contains module communication_ray and subroutines allocate_array_com_ray(), communications_ray(), com_1D_ray(), com_2D_ray() and com_3D_ray()) 362
rad_transfer/m1/pasdt_ray.f90 (Contains subroutines pasdt_ray() and pasdt_ray_exp()) 366
rad_transfer/m1/ray_step.f90 (Contains subroutine ray_step()) 368
rad_transfer/m1/sources_ray.f90 (Contains functions source() and derive_source()) 370
rad_transfer/m1/valp.f90 (Contains subroutine read_valp() and function interpol_valp()) 371

Chapter 6

Directory Documentation

6.1 algebra/ Directory Reference

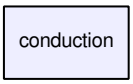


Files

- file cg.f90

  Contains subroutines cg(), vec_prod() and cal_dif().

6.2 conduction/ Directory Reference



Files

- file conduction.f90

  Contains subroutine conduction_ex().

- file conduction_imp_gc.f90

  Contains subroutines conduction_imp_gc(), Mat_prod_con() and cmp_precond_con().

- file conduction_step.f90

  Contains subroutine conduction_step().

- file init_conduction.f90

  Contains subroutine init_conduction().

- file kappa_con.f90

  Contains function cal_kappa_con().

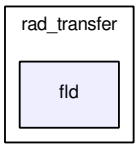
- file modules_con.f90

  Contains modules cl_con and conduction.

- file pasdt_con.f90

  Contains subroutine pasdt_con().

6.3 rad_transfer/fld/ Directory Reference



Files

- file diffusion.f90

  Contains subroutine diffusion_ex().

- file diffusion_imp_gc.f90

  Contains subroutines diffusion_imp(), mat_prod_dif() and cmp_precond_dif().

- file diffusion_step.f90

  Contains subroutine diffusion_step().

- file init_diffusion.f90

  Contains subroutines init_diffusion() and allocate_array_dif() and functions source_dif() and derive_source_dif().

- file kappa_dif.f90

  Contains subroutine cal_kappa_dif().

- file limites_dif.f90

  Contains subroutine limites_dif().

- file modules_dif.f90

  Contains modules cl_dif and diffusion.

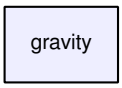
- file para_dif.f90

  Contains subroutine communications_dif().

- file pasdt_dif.f90

  Contains subroutine pasdt_dif().

6.4 gravity/ Directory Reference



Files

- file gc.f90

  Contains subroutines gc(), mat_prod_gra() and cmp_precond_gra().

- file grav_predictor.f90

  Contains subroutine grav_predictor().

- file init_gravity.f90

  Contains subroutine init_gravity().

- file limites_gra.f90

  Contains subroutines limites_gra() and cal_barycentre().

- file modules_gra.f90

  Contains module gravity and subroutine allocate_array_gra().

- file pasdt_grav.f90

  Contains subroutine pasdt_grav().

- file poisson.f90

  Contains subroutines poisson(), pt_mass() and cst_gravity().

- file update_gravity.f90

  Contains subroutine update_gravity().

6.5 hdf/ Directory Reference

Files

- file film_hdf.f90

  Contains subroutine film_hdf().

- file rd_restart_h5.f90

  Contains subroutine rd_restart_h5().

- file rdwrt_h5.f90

  Contains module rdwrt_h5 and subroutines dump_1d_array_int_h5(), dump_1d_array_h5(), dump_3d_-array_h5(), dump_4d_array_h5(), dump_1D_array_string_h5(), get_1d_array_int_h5(), get_1d_array_-h5(), get_3d_array_h5() and get_4d_array_h5(), get_1D_array_string_h5().

- file utils_h5.f90

  Contains subroutines output_data(), get_filename(), get_dataexist() and get_nl().

- file wrt_film_h5.f90

  Contains subroutine wrt_film_h5().

- file wrt_main_h5.f90

  Contains subroutine wrt_main_h5().

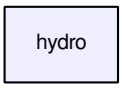
- file wrt_output_h5.f90

  Contains subroutine wrt_output_h5().

- file wrt_restart_h5.f90

  Contains subroutine wrt_restart_h5().

6.6 hydro/ Directory Reference



Files

- file eos.f90

  Contains subroutines init_eos(), pressure(), temperature(), sound_speed(), energy() and heat_capacity().

- file evol_hydro.f90

  Contains subroutines evol_hydro() and pente_minmod().

- file fill_new_arrays.f90

  Contains subroutine fill_new_arrays().

- file histogramme.f90

  Contains module histo and subroutines init_histo() and cal_histogramme().

- file hydro_step.f90

  Contains subroutine hydro_step().

- file monitoring.f90

  Contains subroutine monitoring().

- file pasdt_hydro.f90

  Contains subroutine pasdt_hydro().

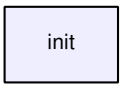
- file rh.f90

  Contains subroutine rh() and function f().

- file solvers.f90

  Contains subroutines solver_relax(), solver_relaxation(), solver_exact(), solver_cc(), solver_acoustic(), solver_eos(), solver_g(), solver_iso(), starpu(), guessp(), prefun(), sample(), starpu_iso(), guessp_iso(), prefun_iso(), sample_iso() and cal_gamma().

6.7 init/ Directory Reference



Files

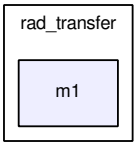
- file init.f90

  Contains subroutines init() and dimensions().

- file init_out.f90

  Contains subroutine init_out().

6.8 rad_transfer/m1/ Directory Reference



Files

- file allocate_array_ray.f90

  Contains subroutine allocate_array_ray().

- file cal_b3d.f90

  Contains subroutine cal_b3d().

- file cal_x3d.f90

  Contains subroutines cal_x3d(), calmoinsun_x3d() and allocate_gmres().

- file cl_ana_ray.f90

  Contains subroutines cl_ana_ray() and cl_ana_gen_ray().

- file evol_ray.f90

  Contains subroutine evol_ray().

- file explicite.f90

  Contains subroutine explicite().

- file gmres.f90

  Contains subroutine gmres().

- file gs.f90

  Contains subroutines gs_mat(), cal_Dedd() and essai_inv5x5 and functions det_3x3() and det_4x4().

- file init_ray.f90

  Contains subroutine init_ray().

- file initvargmres.f90

  Contains subroutine initVarGmres().

- file inversion_gmres.f90

  Contains subroutine inversion_gmres().

- file kappa_ray.f90

  Contains subroutines cal_kappa() and cal_sigmadiff().

- file limites_ray.f90

Contains subroutine limites_ray().

- file mat_prod3d.f90

  Contains subroutines mat_prod3d(), com_1D_xin(), com_2D_xin(), com_3D_xin() and limites_xin().

- file modules_ray.f90

  Contains modules varray, divers_ray, cl_ray, valeurs_propres, TabGmres, VarGmres and interfaces.

- file para_ray.f90

  Contains module communication_ray and subroutines allocate_array_com_ray(), communications_ray(), com_1D_ray(), com_2D_ray() and com_3D_ray().

- file pasdt_ray.f90

  Contains subroutines pasdt_ray() and pasdt_ray_exp().

- file ray_step.f90

  Contains subroutine ray_step().

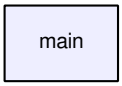
- file sources_ray.f90

  Contains functions source() and derive_source().

- file valp.f90

  Contains subroutine read_valp() and function interpol_valp().

6.9 main/ Directory Reference



Files

- file aff.f90

  Contains subroutines aff_new(), make_name(), mk_dir_out(), convtoasc() and sortie_ecran().

- file allocate_array.f90

  Contains subroutine allocate_array().

- file check_flags.f90

  Contains subroutine check_flags().

- file check_size.f90

  Contains subroutine check_size().

- file cl_ana.f90

  Contains subroutines cl_ana() and cl_ana_gen().

- file cpu.f90

  Contains subroutine cpu_().

- file film.f90

  Contains module mfilm and subroutines film_() and init_film().

- file geom.f90

  Contains subroutines def_geom() and cal_geom().

- file init_phys.f90

  Contains subroutine init_phys().

- file limites.f90

  Contains subroutine limites().

- file limites_generique.f90

  Contains subroutine limites_gen().

- file main.f90

  Contains main program HERACLES and subroutines compute_cputime(), output() and close_program() and functions Arret(), MPI_Wtime(), tremain().

- file modules.f90

  Contains modules parameters, var_loc, geom, var, varold, etat_gds, unites, unites_sortie, divers, param_ini, cl, gammas, cellules, const and cputime.

- file para.f90

  Contains modules communication and para and subroutines init_para(), allocate_array_com(), communications(), com_1D(), com_2D() and com_3D().

- file para_generique.f90

  Contains module communication_gen and subroutines allocate_array_com_gen(), communications_gen(), com_1D_gen(), com_2D_gen() and com_3D().

- file pasdt.f90

  Contains subroutine pasdt().

- file prime.f90

  Contains module prime and subroutines find_prime(), primedec() and cpudec().

- file read_params.f90

  Contains subroutines read_params() and default_params().

- file restart.f90

  Contains subroutine restart_new().

- file set_units_out.f90

  Contains subroutine set_units_out().

- file slopes.f90

  Contains subroutine slopes() and functions minmod(), moncen(), moyharm() and vavl().

- file user_init.f90

  Contains subroutine user_init().

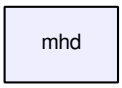
- file user_output.f90

  Contains subroutine user_output().

- file user_step.f90

  Contains subroutine user_step().

6.10 mhd/ Directory Reference



Files

- file ctoprim.f90

  Contains subroutine ctoprim().

- file evol_mhd.f90

  Contains subroutine evol_mhd().

- file godunov_utils.f90

  contains subroutines upwind(), lax_friedrich(), hll(), hlld(), find_mhd_flux(), find_mhd_flux2(), find_speed_info(), find_speed_fast(), find_speed_alfven(), hydro_acoustic(), athena_roe(), eigenvalues() and eigen_cons()

- file pasdt_mhd.f90

  contains subroutine pasdt_mhd()

- file trace.f90

  contains subroutines trace1d(), trace2d() and trace3d()

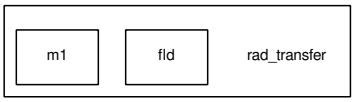
- file umuscl.f90

  contains subroutines cmpflxm(), cmp_mag_flx(), fast_mhd_speed() and uslope()

- file update.f90

  contains subroutine update()

6.11 rad_transfer/ Directory Reference



Directories

- directory fld
- directory m1

Files

- file exp_comobile_ray.f90

  Contains subroutine exp_comobile_ray().

- file exp_source_ray.f90

  Contains subroutine exp_source_ray().

Chapter 7

Module Documentation

7.1 cellules Module Reference

Contains ?

Variables

- integer, dimension(:,:), allocatable Imax
  Imax.
- integer, dimension(:,:), allocatable Imax2
  Imax2.
- integer, dimension(:), allocatable fff_t
  fff_t
- integer N_cell
  N_cell.
- integer fff
  fff
- integer ioffset
  ioffset
- integer ifich
  ifich
- integer size_offset
  size_offset
- integer ifilm = 0
  ifilm
- real, dimension(:,:,:), allocatable Pcell

Pcell.

- real, dimension(:,:), allocatable Pmax

  Pmax.

- real, dimension(:,:), allocatable Pmax2

  Pmax2.

- real dx_cell

  dx_cell

- real t_film

  t_film

- real dt_film

  dt_film

7.1.1 Detailed Description

Contains ?

7.1.2 Variable Documentation

7.1.2.1 real cellules::dt_film

dt_film

Definition at line 441 of file modules.f90.

7.1.2.2 real cellules::dx_cell

dx_cell

Definition at line 439 of file modules.f90.

7.1.2.3 integer cellules::fff

fff

Definition at line 431 of file modules.f90.

7.1.2.4 integer,dimension(:),allocatable cellules::fff_t

fff_t

Definition at line 429 of file modules.f90.

7.1.2.5 integer cellules::ifich

ifich

Definition at line 433 of file modules.f90.

7.1.2.6 integer cellules::ifilm = 0

ifilm

Definition at line 435 of file modules.f90.

7.1.2.7 integer,dimension(:,:),allocatable cellules::Imax

Imax.

Definition at line 427 of file modules.f90.

7.1.2.8 integer,dimension(:,:),allocatable cellules::Imax2

Imax2.

Definition at line 428 of file modules.f90.

7.1.2.9 integer cellules::ioffset

ioffset

Definition at line 432 of file modules.f90.

7.1.2.10 integer cellules::N_cell

N_cell.

Definition at line 430 of file modules.f90.

7.1.2.11 real,dimension(:,:,:),allocatable cellules::Pcell

Pcell.

Definition at line 436 of file modules.f90.

7.1.2.12 real,dimension(:,:),allocatable cellules::Pmax

Pmax.

Definition at line 437 of file modules.f90.

7.1.2.13 real,dimension(:,:),allocatable cellules::Pmax2

Pmax2.

Definition at line 438 of file modules.f90.

7.1.2.14 integer cellules::size_offset

size_offset

Definition at line 434 of file modules.f90.

7.1.2.15 real cellules::t_film

t_film

Definition at line 440 of file modules.f90.

7.2 cl Module Reference

Contains the variables used to define the boundary conditions.

Variables

- integer, dimension(6) icl_glob

  Global boundaries types for faces 1 to 6.

- integer, dimension(6) icl

  Local boundaries types for faces 1 to 6.

- integer, dimension(6) icl0

  icl

- real, dimension(:,:), allocatable rhou_l

  Imposed value for the gas momentum inside the ghost cells.

- real, dimension(:,:), allocatable u_l

  Imposed value for the gas velocity inside the ghost cells.

- real, dimension(:), allocatable rho_l

  Imposed value for the gas density inside the ghost cells.

- real, dimension(:), allocatable E_l

  Imposed value for the gas energy inside the ghost cells.

- real, dimension(:), allocatable P_l

  Imposed value for the gas pressure inside the ghost cells.

- real, dimension(:), allocatable T_l

  Imposed value for the gas temperature inside the ghost cells.

- real, dimension(:,:), allocatable fx_l

  Imposed value for the passive scalar inside the ghost cells.

- real, dimension(:,:), allocatable B_l

  Imposed value for the magnetic field inside the ghost cells.

7.2.1 Detailed Description

Contains the variables used to define the boundary conditions.

7.2.2 Variable Documentation

7.2.2.1 real,dimension(:,:),allocatable cl::B_l

Imposed value for the magnetic field inside the ghost cells.

Definition at line 385 of file modules.f90.

Referenced by allocate_array(), and limites().

7.2.2.2 real,dimension(:),allocatable cl::E_l

Imposed value for the gas energy inside the ghost cells.

Definition at line 381 of file modules.f90.

Referenced by allocate_array(), and limites().

7.2.2.3 real,dimension(:,:),allocatable cl::fx_l

Imposed value for the passive scalar inside the ghost cells.

Definition at line 384 of file modules.f90.

Referenced by allocate_array(), and limites().

7.2.2.4 integer,dimension(6) cl::icl

Local boundaries types for faces 1 to 6.

Definition at line 376 of file modules.f90.

Referenced by gc__interface::cmp_precond_gra(), conduction_ex(), def_geom(), default_params(), diffusion_ex(), evol_hydro(), evol_mhd(), init(), and init_para().

7.2.2.5 integer,dimension(6) cl::icl0

icl

Definition at line 377 of file modules.f90.

Referenced by init_para().

7.2.2.6 integer,dimension(6) cl::icl_glob

Global boundaries types for faces 1 to 6.

Definition at line 375 of file modules.f90.

Referenced by check_flags(), default_params(), init_para(), and read_params().

7.2.2.7 real,dimension(:),allocatable cl::P_l

Imposed value for the gas pressure inside the ghost cells.

Definition at line 382 of file modules.f90.

Referenced by allocate_array().

7.2.2.8 real,dimension(:),allocatable cl::rho_l

Imposed value for the gas density inside the ghost cells.

Definition at line 380 of file modules.f90.

Referenced by allocate_array(), and limites().

7.2.2.9 real,dimension(:,:),allocatable cl::rhou_l

Imposed value for the gas momentum inside the ghost cells.

Definition at line 378 of file modules.f90.

Referenced by allocate_array(), and limites().

7.2.2.10 real,dimension(:),allocatable cl::T_l

Imposed value for the gas temperature inside the ghost cells.

Definition at line 383 of file modules.f90.

Referenced by allocate_array(), and limites_ray().

7.2.2.11 real,dimension(:,:),allocatable cl::u_l

Imposed value for the gas velocity inside the ghost cells.

Definition at line 379 of file modules.f90.

Referenced by allocate_array().

7.3 cl_con Module Reference

Contains the variables used for the conduction boundary conditions.

Variables

- integer, dimension(6) icl_con
  Local conduction boundary conditions.
- integer, dimension(6) icl_con_glob
  Global conduction boundary conditions.
- real, dimension(:), allocatable T_l
  Temperature in the conduction boundaries.

7.3.1 Detailed Description

Contains the variables used for the conduction boundary conditions.

7.3.2 Variable Documentation

7.3.2.1 integer,dimension(6) cl_con::icl_con

Local conduction boundary conditions.

Definition at line 17 of file modules_con.f90.

Referenced by conduction_imp_gc__interface::cmp_precond_con(), default_params(), and init_para().

7.3.2.2 integer,dimension(6) cl_con::icl_con_glob

Global conduction boundary conditions.

Definition at line 18 of file modules_con.f90.

Referenced by default_params(), init_conduction(), and init_para().

7.3.2.3 real,dimension(:),allocatable cl_con::T_l

Temperature in the conduction boundaries.

Definition at line 19 of file modules_con.f90.

Referenced by init_conduction().

7.4 cl_dif Module Reference

Contains the variables for the FLD boundary conditions.

Variables

- integer, dimension(6) icl_dif
  Type of local FLD boundary conditions.
- integer, dimension(6) icl_dif_glob
  Type of global FLD boundary conditions.
- real, dimension(:), allocatable Er_dif_l
  Radiative energy in the boundaries for the FLD.

7.4.1 Detailed Description

Contains the variables for the FLD boundary conditions.

7.4.2 Variable Documentation

7.4.2.1 real,dimension(:),allocatable cl_dif::Er_dif_l

Radiative energy in the boundaries for the FLD.

Definition at line 18 of file modules_dif.f90.

Referenced by allocate_array_dif(), and limites_dif().

7.4.2.2 integer,dimension(6) cl_dif::icl_dif

Type of local FLD boundary conditions.

Definition at line 16 of file modules_dif.f90.

Referenced by diffusion_imp__interface::cmp_precond_dif(), init_para(), and limites_dif().

7.4.2.3 integer,dimension(6) cl_dif::icl_dif_glob

Type of global FLD boundary conditions.

Definition at line 17 of file modules_dif.f90.

Referenced by default_params(), init_para(), and read_params().

7.5 cl_ray Module Reference

Contains the boundary conditions variables for the M1 radiative transfer.

Variables

- integer, dimension(6) icl_ray
  Type of radiative boundary conditions.
- integer, dimension(6) icl_ray_glob
  Type of global radiative boundary conditions.
- real, dimension(:), allocatable Er_l
  Radiative energy imposed in the ghost cells.
- real, dimension(:), allocatable Tr_l
  Gas temperature imposed in the ghost cells.
- real, dimension(:,:), allocatable Fr_l
  Radiative flux imposed in the ghost cells.

7.5.1 Detailed Description

Contains the boundary conditions variables for the M1 radiative transfer.

7.5.2 Variable Documentation

7.5.2.1 real,dimension(:),allocatable cl_ray::Er_l

Radiative energy imposed in the ghost cells.

Definition at line 84 of file modules_ray.f90.

Referenced by allocate_array_ray(), limites_ray(), and limites_xin().

7.5.2.2 real,dimension(:,:),allocatable cl_ray::Fr_l

Radiative flux imposed in the ghost cells.

Definition at line 86 of file modules_ray.f90.

Referenced by allocate_array_ray(), limites_ray(), and limites_xin().

7.5.2.3 integer,dimension(6) cl_ray::icl_ray

Type of radiative boundary conditions.

Definition at line 82 of file modules_ray.f90.

Referenced by cal_b3d(), gs_mat(), init_para(), and mat_prod3d().

7.5.2.4 integer,dimension(6) cl_ray::icl_ray_glob

Type of global radiative boundary conditions.

Definition at line 83 of file modules_ray.f90.

Referenced by check_flags(), default_params(), init_para(), and read_params().

7.5.2.5 real,dimension(:),allocatable cl_ray::Tr_l

Gas temperature imposed in the ghost cells.

Definition at line 85 of file modules_ray.f90.

Referenced by allocate_array_ray().

7.6 communication Module Reference

Contains variables arrays used to communicate ghost cell data across cpus.

Variables

- real, dimension(:,:,:,:), allocatable face1_e
  Face 1 e.
- real, dimension(:,:,:,:), allocatable face2_e
  Face 2 e.
- real, dimension(:,:,:,:), allocatable face3_e
  Face 3 e.
- real, dimension(:,:,:,:), allocatable face4_e
  Face 4 e.
- real, dimension(:,:,:,:), allocatable face5_e
  Face 5 e.
- real, dimension(:,:,:,:), allocatable face6_e
  Face 6 e.
- real, dimension(:,:,:,:), allocatable face1_r
  Face 1 r.
- real, dimension(:,:,:,:), allocatable face2_r
  Face 2 r.
- real, dimension(:,:,:,:), allocatable face3_r
  Face 3 r.
- real, dimension(:,:,:,:), allocatable face4_r
  Face 4 r.
- real, dimension(:,:,:,:), allocatable face5_r
  Face 5 r.
- real, dimension(:,:,:,:), allocatable face6_r
  Face 6 r.
- real, dimension(:,:,:,:), allocatable ar13_e
  Edge 1 e.
- real, dimension(:,:,:,:), allocatable ar14_e
  Edge 2 e.
- real, dimension(:,:,:,:), allocatable ar23_e

Edge 3 e.

- real, dimension(:,:,:,:), allocatable ar24_e
  Edge 4 e.
- real, dimension(:,:,:,:), allocatable ar15_e
  Edge 5 e.
- real, dimension(:,:,:,:), allocatable ar25_e
  Edge 6 e.
- real, dimension(:,:,:,:), allocatable ar35_e
  Edge 7 e.
- real, dimension(:,:,:,:), allocatable ar45_e
  Edge 8 e.
- real, dimension(:,:,:,:), allocatable ar16_e
  Edge 9 e.
- real, dimension(:,:,:,:), allocatable ar26_e
  Edge 10 e.
- real, dimension(:,:,:,:), allocatable ar36_e
  Edge 11 e.
- real, dimension(:,:,:,:), allocatable ar46_e
  Edge 12 e.
- real, dimension(:,:,:,:), allocatable ar13_r
  Edge 1 r.
- real, dimension(:,:,:,:), allocatable ar14_r
  Edge 2 r.
- real, dimension(:,:,:,:), allocatable ar23_r
  Edge 3 r.
- real, dimension(:,:,:,:), allocatable ar24_r
  Edge 4 r.
- real, dimension(:,:,:,:), allocatable ar15_r
  Edge 5 r.
- real, dimension(:,:,:,:), allocatable ar25_r
  Edge 6 r.
- real, dimension(:,:,:,:), allocatable ar35_r
  Edge 7 r.

- real, dimension(:,:,:,:), allocatable ar45_r
  Edge 8 r.
- real, dimension(:,:,:,:), allocatable ar16_r
  Edge 9 r.
- real, dimension(:,:,:,:), allocatable ar26_r
  Edge 10 r.
- real, dimension(:,:,:,:), allocatable ar36_r
  Edge 11 r.
- real, dimension(:,:,:,:), allocatable ar46_r
  Edge 12 r.
- real, dimension(:,:,:,:), allocatable coin13_e
  Corner 2D 1 e.
- real, dimension(:,:,:,:), allocatable coin14_e
  Corner 2D 2 e.
- real, dimension(:,:,:,:), allocatable coin23_e
  Corner 2D 3 e.
- real, dimension(:,:,:,:), allocatable coin24_e
  Corner 2D 4 e.
- real, dimension(:,:,:,:), allocatable coin13_r
  Corner 2D 1 r.
- real, dimension(:,:,:,:), allocatable coin14_r
  Corner 2D 2 r.
- real, dimension(:,:,:,:), allocatable coin23_r
  Corner 2D 3 r.
- real, dimension(:,:,:,:), allocatable coin24_r
  Corner 2D 4 r.
- real, dimension(:,:,:,:), allocatable coin135_e
  Corner 3D 1 e.
- real, dimension(:,:,:,:), allocatable coin136_e
  Corner 3D 2 e.
- real, dimension(:,:,:,:), allocatable coin145_e
  Corner 3D 3 e.
- real, dimension(:,:,:,:), allocatable coin146_e
  Corner 3D 4 e.

- real, dimension(:,:,:,:), allocatable coin235_e
  Corner 3D 5 e.
- real, dimension(:,:,:,:), allocatable coin236_e
  Corner 3D 6 e.
- real, dimension(:,:,:,:), allocatable coin245_e
  Corner 3D 7 e.
- real, dimension(:,:,:,:), allocatable coin246_e
  Corner 3D 8 e.
- real, dimension(:,:,:,:), allocatable coin135_r
  Corner 3D 1 r.
- real, dimension(:,:,:,:), allocatable coin136_r
  Corner 3D 2 r.
- real, dimension(:,:,:,:), allocatable coin145_r
  Corner 3D 3 r.
- real, dimension(:,:,:,:), allocatable coin146_r
  Corner 3D 4 r.
- real, dimension(:,:,:,:), allocatable coin235_r
  Corner 3D 5 r.
- real, dimension(:,:,:,:), allocatable coin236_r
  Corner 3D 6 r.
- real, dimension(:,:,:,:), allocatable coin245_r
  Corner 3D 7 r.
- real, dimension(:,:,:,:), allocatable coin246_r
  Corner 3D 8 r.

7.6.1 Detailed Description

Contains variables arrays used to communicate ghost cell data across cpus.

7.6.2 Variable Documentation

7.6.2.1 real,dimension(:,:,:,:),allocatable communication::ar13_e

Edge 1 e.

Definition at line 34 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.2 real,dimension(:,:,:,:),allocatable communication::ar13_r

Edge 1 r.

Definition at line 46 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.3 real,dimension(:,:,:,:),allocatable communication::ar14_e

Edge 2 e.

Definition at line 35 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.4 real,dimension(:,:,:,:),allocatable communication::ar14_r

Edge 2 r.

Definition at line 47 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.5 real,dimension(:,:,:,:),allocatable communication::ar15_e

Edge 5 e.

Definition at line 38 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.6 real,dimension(:,:,:,:),allocatable communication::ar15_r

Edge 5 r.

Definition at line 50 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.7 real,dimension(:,:,:,:),allocatable communication::ar16_e

Edge 9 e.

Definition at line 42 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.8 real,dimension(:,:,:,:),allocatable communication::ar16_r

Edge 9 r.

Definition at line 54 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.9 real,dimension(:,:,:,:),allocatable communication::ar23_e

Edge 3 e.

Definition at line 36 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.10 real,dimension(:,:,:,:),allocatable communication::ar23_r

Edge 3 r.

Definition at line 48 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.11 real,dimension(:,:,:,:),allocatable communication::ar24_e

Edge 4 e.

Definition at line 37 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.12 real,dimension(:,:,:,:),allocatable communication::ar24_r

Edge 4 r.

Definition at line 49 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.13 real,dimension(:,:,:,:),allocatable communication::ar25_e

Edge 6 e.

Definition at line 39 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.14 real,dimension(:,:,:,:),allocatable communication::ar25_r

Edge 6 r.

Definition at line 51 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.15 real,dimension(:,:,:,:),allocatable communication::ar26_e

Edge 10 e.

Definition at line 43 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.16 real,dimension(:,:,:,:),allocatable communication::ar26_r

Edge 10 r.

Definition at line 55 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.17 real,dimension(:,:,:,:),allocatable communication::ar35_e

Edge 7 e.

Definition at line 40 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.18 real,dimension(:,:,:,:),allocatable communication::ar35_r

Edge 7 r.

Definition at line 52 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.19 real,dimension(:,:,:,:),allocatable communication::ar36_e

Edge 11 e.

Definition at line 44 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.20 real,dimension(:,:,:,:),allocatable communication::ar36_r

Edge 11 r.

Definition at line 56 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.21 real,dimension(:,:,:,:),allocatable communication::ar45_e

Edge 8 e.

Definition at line 41 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.22 real,dimension(:,:,:,:),allocatable communication::ar45_r

Edge 8 r.

Definition at line 53 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.23 real,dimension(:,:,:,:),allocatable communication::ar46_e

Edge 12 e.

Definition at line 45 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.24 real,dimension(:,:,:,:),allocatable communication::ar46_r

Edge 12 r.

Definition at line 57 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.25 real,dimension(:,:,:,:),allocatable communication::coin135_e

Corner 3D 1 e.

Definition at line 66 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.26 real,dimension(:,:,:,:),allocatable communication::coin135_r

Corner 3D 1 r.

Definition at line 74 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.27 real,dimension(:,:,:,:),allocatable communication::coin136_e

Corner 3D 2 e.

Definition at line 67 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.28 real,dimension(:,:,:,:),allocatable communication::coin136_r

Corner 3D 2 r.

Definition at line 75 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.29 real,dimension(:,:,:,:),allocatable communication::coin13_e

Corner 2D 1 e.

Definition at line 58 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.30 real,dimension(:,:,:,:),allocatable communication::coin13_r

Corner 2D 1 r.

Definition at line 62 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.31 real,dimension(:,:,:,:),allocatable communication::coin145_e

Corner 3D 3 e.

Definition at line 68 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.32 real,dimension(:,:,:,:),allocatable communication::coin145_r

Corner 3D 3 r.

Definition at line 76 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.33 real,dimension(:,:,:,:),allocatable communication::coin146_e

Corner 3D 4 e.

Definition at line 69 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.34 real,dimension(:,:,:,:),allocatable communication::coin146_r

Corner 3D 4 r.

Definition at line 77 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.35 real,dimension(:,:,:,:),allocatable communication::coin14_e

Corner 2D 2 e.

Definition at line 59 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.36 real,dimension(:,:,:,:),allocatable communication::coin14_r

Corner 2D 2 r.

Definition at line 63 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.37 real,dimension(:,:,:,:),allocatable communication::coin235_e

Corner 3D 5 e.

Definition at line 70 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.38 real,dimension(:,:,:,:),allocatable communication::coin235_r

Corner 3D 5 r.

Definition at line 78 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.39 real,dimension(:,:,:,:),allocatable communication::coin236_e

Corner 3D 6 e.

Definition at line 71 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.40 real,dimension(:,:,:,:),allocatable communication::coin236_r

Corner 3D 6 r.

Definition at line 79 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.41 real,dimension(:,:,:,:),allocatable communication::coin23_e

Corner 2D 3 e.

Definition at line 60 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.42 real,dimension(:,:,:,:),allocatable communication::coin23_r

Corner 2D 3 r.

Definition at line 64 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.43 real,dimension(:,:,:,:),allocatable communication::coin245_e

Corner 3D 7 e.

Definition at line 72 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.44 real,dimension(:,:,:,:),allocatable communication::coin245_r

Corner 3D 7 r.

Definition at line 80 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.45 real,dimension(:,:,:,:),allocatable communication::coin246_e

Corner 3D 8 e.

Definition at line 73 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.46 real,dimension(:,:,:,:),allocatable communication::coin246_r

Corner 3D 8 r.

Definition at line 81 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.47 real,dimension(:,:,:,:),allocatable communication::coin24_e

Corner 2D 4 e.

Definition at line 61 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.48 real,dimension(:,:,:,:),allocatable communication::coin24_r

Corner 2D 4 r.

Definition at line 65 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.49 real,dimension(:,:,:,:),allocatable communication::face1_e

Face 1 e.

Definition at line 22 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.50 real,dimension(:,:,:,:),allocatable communication::face1_r

Face 1 r.

Definition at line 28 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.51 real,dimension(:,:,:,:),allocatable communication::face2_e

Face 2 e.

Definition at line 23 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.52 real,dimension(:,:,:,:),allocatable communication::face2_r

Face 2 r.

Definition at line 29 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.53 real,dimension(:,:,:,:),allocatable communication::face3_e

Face 3 e.

Definition at line 24 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.54 real,dimension(:,:,:,:),allocatable communication::face3_r

Face 3 r.

Definition at line 30 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.55 real,dimension(:,:,:,:),allocatable communication::face4_e

Face 4 e.

Definition at line 25 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.56 real,dimension(:,:,:,:),allocatable communication::face4_r

Face 4 r.

Definition at line 31 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.57 real,dimension(:,:,:,:),allocatable communication::face5_e

Face 5 e.

Definition at line 26 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.58 real,dimension(:,:,:,:),allocatable communication::face5_r

Face 5 r.

Definition at line 32 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.59 real,dimension(:,:,:,:),allocatable communication::face6_e

Face 6 e.

Definition at line 27 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.6.2.60 real,dimension(:,:,:,:),allocatable communication::face6_r

Face 6 r.

Definition at line 33 of file para.f90.

Referenced by allocate_array_com(), communications(), and mat_prod3d().

7.7 communication_gen Module Reference

Contains variables arrays used to communicate ghost cell data across cpus.

Variables

- real, dimension(:,:,:,:), allocatable face1_e
  Face 1 e.
- real, dimension(:,:,:,:), allocatable face2_e
  Face 2 e.
- real, dimension(:,:,:,:), allocatable face3_e
  Face 3 e.
- real, dimension(:,:,:,:), allocatable face4_e
  Face 4 e.
- real, dimension(:,:,:,:), allocatable face5_e
  Face 5 e.
- real, dimension(:,:,:,:), allocatable face6_e
  Face 6 e.
- real, dimension(:,:,:,:), allocatable face1_r
  Face 1 r.
- real, dimension(:,:,:,:), allocatable face2_r
  Face 2 r.
- real, dimension(:,:,:,:), allocatable face3_r
  Face 3 r.
- real, dimension(:,:,:,:), allocatable face4_r
  Face 4 r.
- real, dimension(:,:,:,:), allocatable face5_r
  Face 5 r.
- real, dimension(:,:,:,:), allocatable face6_r
  Face 6 r.
- real, dimension(:,:,:,:), allocatable ar13_e
  Edge 1 e.
- real, dimension(:,:,:,:), allocatable ar14_e
  Edge 2 e.
- real, dimension(:,:,:,:), allocatable ar23_e

Edge 3 e.

- real, dimension(:,:,:,:), allocatable ar24_e
  Edge 4 e.
- real, dimension(:,:,:,:), allocatable ar15_e
  Edge 5 e.
- real, dimension(:,:,:,:), allocatable ar25_e
  Edge 6 e.
- real, dimension(:,:,:,:), allocatable ar35_e
  Edge 7 e.
- real, dimension(:,:,:,:), allocatable ar45_e
  Edge 8 e.
- real, dimension(:,:,:,:), allocatable ar16_e
  Edge 9 e.
- real, dimension(:,:,:,:), allocatable ar26_e
  Edge 10 e.
- real, dimension(:,:,:,:), allocatable ar36_e
  Edge 11 e.
- real, dimension(:,:,:,:), allocatable ar46_e
  Edge 12 e.
- real, dimension(:,:,:,:), allocatable ar13_r
  Edge 1 r.
- real, dimension(:,:,:,:), allocatable ar14_r
  Edge 2 r.
- real, dimension(:,:,:,:), allocatable ar23_r
  Edge 3 r.
- real, dimension(:,:,:,:), allocatable ar24_r
  Edge 4 r.
- real, dimension(:,:,:,:), allocatable ar15_r
  Edge 5 r.
- real, dimension(:,:,:,:), allocatable ar25_r
  Edge 6 r.
- real, dimension(:,:,:,:), allocatable ar35_r
  Edge 7 r.

- real, dimension(:,:,:,:), allocatable ar45_r
  Edge 8 r.
- real, dimension(:,:,:,:), allocatable ar16_r
  Edge 9 r.
- real, dimension(:,:,:,:), allocatable ar26_r
  Edge 10 r.
- real, dimension(:,:,:,:), allocatable ar36_r
  Edge 11 r.
- real, dimension(:,:,:,:), allocatable ar46_r
  Edge 12 r.
- real, dimension(:,:,:,:), allocatable coin13_e
  Corner 2D 1 e.
- real, dimension(:,:,:,:), allocatable coin14_e
  Corner 2D 2 e.
- real, dimension(:,:,:,:), allocatable coin23_e
  Corner 2D 3 e.
- real, dimension(:,:,:,:), allocatable coin24_e
  Corner 2D 4 e.
- real, dimension(:,:,:,:), allocatable coin13_r
  Corner 2D 1 r.
- real, dimension(:,:,:,:), allocatable coin14_r
  Corner 2D 2 r.
- real, dimension(:,:,:,:), allocatable coin23_r
  Corner 2D 3 r.
- real, dimension(:,:,:,:), allocatable coin24_r
  Corner 2D 4 r.
- real, dimension(:,:,:,:), allocatable coin135_e
  Corner 3D 1 e.
- real, dimension(:,:,:,:), allocatable coin136_e
  Corner 3D 2 e.
- real, dimension(:,:,:,:), allocatable coin145_e
  Corner 3D 3 e.
- real, dimension(:,:,:,:), allocatable coin146_e
  Corner 3D 4 e.

- real, dimension(:,:,:,:), allocatable coin235_e

  Corner 3D 5 e.

- real, dimension(:,:,:,:), allocatable coin236_e

  Corner 3D 6 e.

- real, dimension(:,:,:,:), allocatable coin245_e

  Corner 3D 7 e.

- real, dimension(:,:,:,:), allocatable coin246_e

  Corner 3D 8 e.

- real, dimension(:,:,:,:), allocatable coin135_r

  Corner 3D 1 r.

- real, dimension(:,:,:,:), allocatable coin136_r

  Corner 3D 2 r.

- real, dimension(:,:,:,:), allocatable coin145_r

  Corner 3D 3 r.

- real, dimension(:,:,:,:), allocatable coin146_r

  Corner 3D 4 r.

- real, dimension(:,:,:,:), allocatable coin235_r

  Corner 3D 5 r.

- real, dimension(:,:,:,:), allocatable coin236_r

  Corner 3D 6 r.

- real, dimension(:,:,:,:), allocatable coin245_r

  Corner 3D 7 r.

- real, dimension(:,:,:,:), allocatable coin246_r

  Corner 3D 8 r.

- integer, parameter Nmax = 2

  nmax

7.7.1 Detailed Description

Contains variables arrays used to communicate ghost cell data across cpus.

7.7.2 Variable Documentation

7.7.2.1 real,dimension(:,:,:,:),allocatable communication_gen::ar13_e

Edge 1 e.

Definition at line 34 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.2 real,dimension(:,:,:,:),allocatable communication_gen::ar13_r

Edge 1 r.

Definition at line 46 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.3 real,dimension(:,:,:,:),allocatable communication_gen::ar14_e

Edge 2 e.

Definition at line 35 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.4 real,dimension(:,:,:,:),allocatable communication_gen::ar14_r

Edge 2 r.

Definition at line 47 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.5 real,dimension(:,:,:,:),allocatable communication_gen::ar15_e

Edge 5 e.

Definition at line 38 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.6 real,dimension(:,:,:,:),allocatable communication_gen::ar15_r

Edge 5 r.

Definition at line 50 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.7 real,dimension(:,:,:,:),allocatable communication_gen::ar16_e

Edge 9 e.

Definition at line 42 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.8 real,dimension(:,:,:,:),allocatable communication_gen::ar16_r

Edge 9 r.

Definition at line 54 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.9 real,dimension(:,:,:,:),allocatable communication_gen::ar23_e

Edge 3 e.

Definition at line 36 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.10 real,dimension(:,:,:,:),allocatable communication_gen::ar23_r

Edge 3 r.

Definition at line 48 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.11 real,dimension(:,:,:,:),allocatable communication_gen::ar24_e

Edge 4 e.

Definition at line 37 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.12 real,dimension(:,:,:,:),allocatable communication_gen::ar24_r

Edge 4 r.

Definition at line 49 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.13 real,dimension(:,:,:,:),allocatable communication_gen::ar25_e

Edge 6 e.

Definition at line 39 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.14 real,dimension(:,:,:,:),allocatable communication_gen::ar25_r

Edge 6 r.

Definition at line 51 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.15 real,dimension(:,:,:,:),allocatable communication_gen::ar26_e

Edge 10 e.

Definition at line 43 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.16 real,dimension(:,:,:,:),allocatable communication_gen::ar26_r

Edge 10 r.

Definition at line 55 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.17 real,dimension(:,:,:,:),allocatable communication_gen::ar35_e

Edge 7 e.

Definition at line 40 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.18 real,dimension(:,:,:,:),allocatable communication_gen::ar35_r

Edge 7 r.

Definition at line 52 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.19 real,dimension(:,:,:,:),allocatable communication_gen::ar36_e

Edge 11 e.

Definition at line 44 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.20 real,dimension(:,:,:,:),allocatable communication_gen::ar36_r

Edge 11 r.

Definition at line 56 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.21 real,dimension(:,:,:,:),allocatable communication_gen::ar45_e

Edge 8 e.

Definition at line 41 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.22 real,dimension(:,:,:,:),allocatable communication_gen::ar45_r

Edge 8 r.

Definition at line 53 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.23 real,dimension(:,:,:,:),allocatable communication_gen::ar46_e

Edge 12 e.

Definition at line 45 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.24 real,dimension(:,:,:,:),allocatable communication_gen::ar46_r

Edge 12 r.

Definition at line 57 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.25 real,dimension(:,:,:,:),allocatable communication_gen::coin135_e

Corner 3D 1 e.

Definition at line 66 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.26 real,dimension(:,:,:,:),allocatable communication_gen::coin135_r

Corner 3D 1 r.

Definition at line 74 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.27 real,dimension(:,:,:,:),allocatable communication_gen::coin136_e

Corner 3D 2 e.

Definition at line 67 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.28 real,dimension(:,:,:,:),allocatable communication_gen::coin136_r

Corner 3D 2 r.

Definition at line 75 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.29 real,dimension(:,:,:,:),allocatable communication_gen::coin13_e

Corner 2D 1 e.

Definition at line 58 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.30 real,dimension(:,:,:,:),allocatable communication_gen::coin13_r

Corner 2D 1 r.

Definition at line 62 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.31 real,dimension(:,:,:,:),allocatable communication_gen::coin145_e

Corner 3D 3 e.

Definition at line 68 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.32 real,dimension(:,:,:,:),allocatable communication_gen::coin145_r

Corner 3D 3 r.

Definition at line 76 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.33 real,dimension(:,:,:,:),allocatable communication_gen::coin146_e

Corner 3D 4 e.

Definition at line 69 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.34 real,dimension(:,:,:,:),allocatable communication_gen::coin146_r

Corner 3D 4 r.

Definition at line 77 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.35 real,dimension(:,:,:,:),allocatable communication_gen::coin14_e

Corner 2D 2 e.

Definition at line 59 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.36 real,dimension(:,:,:,:),allocatable communication_gen::coin14_r

Corner 2D 2 r.

Definition at line 63 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.37 real,dimension(:,:,:,:),allocatable communication_gen::coin235_e

Corner 3D 5 e.

Definition at line 70 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.38 real,dimension(:,:,:,:),allocatable communication_gen::coin235_r

Corner 3D 5 r.

Definition at line 78 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.39 real,dimension(:,:,:,:),allocatable communication_gen::coin236_e

Corner 3D 6 e.

Definition at line 71 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.40 real,dimension(:,:,:,:),allocatable communication_gen::coin236_r

Corner 3D 6 r.

Definition at line 79 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.41 real,dimension(:,:,:,:),allocatable communication_gen::coin23_e

Corner 2D 3 e.

Definition at line 60 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.42 real,dimension(:,:,:,:),allocatable communication_gen::coin23_r

Corner 2D 3 r.

Definition at line 64 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.43 real,dimension(:,:,:,:),allocatable communication_gen::coin245_e

Corner 3D 7 e.

Definition at line 72 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.44 real,dimension(:,:,:,:),allocatable communication_gen::coin245_r

Corner 3D 7 r.

Definition at line 80 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.45 real,dimension(:,:,:,:),allocatable communication_gen::coin246_e

Corner 3D 8 e.

Definition at line 73 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.46 real,dimension(:,:,:,:),allocatable communication_gen::coin246_r

Corner 3D 8 r.

Definition at line 81 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.47 real,dimension(:,:,:,:),allocatable communication_gen::coin24_e

Corner 2D 4 e.

Definition at line 61 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.48 real,dimension(:,:,:,:),allocatable communication_gen::coin24_r

Corner 2D 4 r.

Definition at line 65 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.49 real,dimension(:,:,:,:),allocatable communication_gen::face1_e

Face 1 e.

Definition at line 22 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.50 real,dimension(:,:,:,:),allocatable communication_gen::face1_r

Face 1 r.

Definition at line 28 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.51 real,dimension(:,:,:,:),allocatable communication_gen::face2_e

Face 2 e.

Definition at line 23 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.52 real,dimension(:,:,:,:),allocatable communication_gen::face2_r

Face 2 r.

Definition at line 29 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.53 real,dimension(:,:,:,:),allocatable communication_gen::face3_e

Face 3 e.

Definition at line 24 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.54 real,dimension(:,:,:,:),allocatable communication_gen::face3_r

Face 3 r.

Definition at line 30 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.55 real,dimension(:,:,:,:),allocatable communication_gen::face4_e

Face 4 e.

Definition at line 25 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.56 real,dimension(:,:,:,:),allocatable communication_gen::face4_r

Face 4 r.

Definition at line 31 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.57 real,dimension(:,:,:,:),allocatable communication_gen::face5_e

Face 5 e.

Definition at line 26 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.58 real,dimension(:,:,:,:),allocatable communication_gen::face5_r

Face 5 r.

Definition at line 32 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.59 real,dimension(:,:,:,:),allocatable communication_gen::face6_e

Face 6 e.

Definition at line 27 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.60 real,dimension(:,:,:,:),allocatable communication_gen::face6_r

Face 6 r.

Definition at line 33 of file para_generique.f90.

Referenced by allocate_array_com_gen(), and communications_gen().

7.7.2.61 integer,parameter communication_gen::Nmax = 2

nmax

Definition at line 82 of file para_generique.f90.

Referenced by allocate_array_com_gen().

7.8 communication_ray Module Reference

Contains the communication arrays for the M1 radiative transfer.

Variables

- real, dimension(:,:,:,:), allocatable face1_e
  real, dimension(:,:,:,:), allocatable :: face2_e !<
- real, dimension(:,:,:,:), allocatable face3_e
  real, dimension(:,:,:,:), allocatable :: face4_e !<
- real, dimension(:,:,:,:), allocatable face5_e
  real, dimension(:,:,:,:), allocatable :: face6_e !<
- real, dimension(:,:,:,:), allocatable face1_r
  real, dimension(:,:,:,:), allocatable :: face2_r !<
- real, dimension(:,:,:,:), allocatable face3_r
  real, dimension(:,:,:,:), allocatable :: face4_r !<
- real, dimension(:,:,:,:), allocatable face5_r
  real, dimension(:,:,:,:), allocatable :: face6_r !<
- real, dimension(:,:,:,:), allocatable ar13_e
  real, dimension(:,:,:,:), allocatable :: ar14_e !<
- real, dimension(:,:,:,:), allocatable ar23_e
  real, dimension(:,:,:,:), allocatable :: ar24_e !<
- real, dimension(:,:,:,:), allocatable ar15_e
  real, dimension(:,:,:,:), allocatable :: ar25_e !<
- real, dimension(:,:,:,:), allocatable ar35_e
  real, dimension(:,:,:,:), allocatable :: ar45_e !<
- real, dimension(:,:,:,:), allocatable ar16_e
  real, dimension(:,:,:,:), allocatable :: ar26_e !<
- real, dimension(:,:,:,:), allocatable ar36_e
  real, dimension(:,:,:,:), allocatable :: ar46_e !<
- real, dimension(:,:,:,:), allocatable ar13_r
  real, dimension(:,:,:,:), allocatable :: ar14_r !<
- real, dimension(:,:,:,:), allocatable ar23_r
  real, dimension(:,:,:,:), allocatable :: ar24_r !<
- real, dimension(:,:,:,:), allocatable ar15_r

real, dimension(:,:,:,:), allocatable :: ar25_r !<

- real, dimension(:,:,:,:), allocatable ar35_r
  real, dimension(:,:,:,:), allocatable :: ar45_r !<
- real, dimension(:,:,:,:), allocatable ar16_r
  real, dimension(:,:,:,:), allocatable :: ar26_r !<
- real, dimension(:,:,:,:), allocatable ar36_r
  real, dimension(:,:,:,:), allocatable :: ar46_r !<
- real, dimension(:,:,:,:), allocatable coin13_e
  real, dimension(:,:,:,:), allocatable :: coin14_e !<
- real, dimension(:,:,:,:), allocatable coin23_e
  real, dimension(:,:,:,:), allocatable :: coin24_e !<
- real, dimension(:,:,:,:), allocatable coin13_r
  real, dimension(:,:,:,:), allocatable :: coin14_r !<
- real, dimension(:,:,:,:), allocatable coin23_r
  real, dimension(:,:,:,:), allocatable :: coin24_r !<
- real, dimension(:,:,:,:), allocatable coin135_e
  real, dimension(:,:,:,:), allocatable :: coin136_e !<
- real, dimension(:,:,:,:), allocatable coin145_e
  real, dimension(:,:,:,:), allocatable :: coin146_e !<
- real, dimension(:,:,:,:), allocatable coin235_e
  real, dimension(:,:,:,:), allocatable :: coin236_e !<
- real, dimension(:,:,:,:), allocatable coin245_e
  real, dimension(:,:,:,:), allocatable :: coin246_e !<
- real, dimension(:,:,:,:), allocatable coin135_r
  real, dimension(:,:,:,:), allocatable :: coin136_r !<
- real, dimension(:,:,:,:), allocatable coin145_r
  real, dimension(:,:,:,:), allocatable :: coin146_r !<
- real, dimension(:,:,:,:), allocatable coin235_r
  real, dimension(:,:,:,:), allocatable :: coin236_r !<
- real, dimension(:,:,:,:), allocatable coin245_r
  real, dimension(:,:,:,:), allocatable :: coin246_r !<

7.8.1 Detailed Description

Contains the communication arrays for the M1 radiative transfer.

7.8.2 Variable Documentation

7.8.2.1 real,dimension(:,:,:,:),allocatable communication_ray::ar13_e

real, dimension(:,:,:,:), allocatable :: ar14_e !<

Definition at line 31 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.2 real,dimension(:,:,:,:),allocatable communication_ray::ar13_r

real, dimension(:,:,:,:), allocatable :: ar14_r !<

Definition at line 43 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.3 real,dimension(:,:,:,:),allocatable communication_ray::ar15_e

real, dimension(:,:,:,:), allocatable :: ar25_e !<

Definition at line 35 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.4 real,dimension(:,:,:,:),allocatable communication_ray::ar15_r

real, dimension(:,:,:,:), allocatable :: ar25_r !<

Definition at line 47 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.5 real,dimension(:,:,:,:),allocatable communication_ray::ar16_e

real, dimension(:,:,:,:), allocatable :: ar26_e !<

Definition at line 39 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.6 real,dimension(:,:,:,:),allocatable communication_ray::ar16_r

real, dimension(:,:,:,:), allocatable :: ar26_r !<

Definition at line 51 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.7 real,dimension(:,:,:,:),allocatable communication_ray::ar23_e

real, dimension(:,:,:,:), allocatable :: ar24_e !<

Definition at line 33 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.8 real,dimension(:,:,:,:),allocatable communication_ray::ar23_r

real, dimension(:,:,:,:), allocatable :: ar24_r !<

Definition at line 45 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.9 real,dimension(:,:,:,:),allocatable communication_ray::ar35_e

real, dimension(:,:,:,:), allocatable :: ar45_e !<

Definition at line 37 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.10 real,dimension(:,:,:,:),allocatable communication_ray::ar35_r

real, dimension(:,:,:,:), allocatable :: ar45_r !<

Definition at line 49 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.11 real,dimension(:,:,:,:),allocatable communication_ray::ar36_e

real, dimension(:,:,:,:), allocatable :: ar46_e !<

Definition at line 41 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.12 real,dimension(:,:,:,:),allocatable communication_ray::ar36_r

real, dimension(:,:,:,:), allocatable :: ar46_r !<

Definition at line 53 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.13 real,dimension(:,:,:,:),allocatable communication_ray::coin135_e

real, dimension(:,:,:,:), allocatable :: coin136_e !<

Definition at line 63 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.14 real,dimension(:,:,:,:),allocatable communication_ray::coin135_r

real, dimension(:,:,:,:), allocatable :: coin136_r !<

Definition at line 71 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.15 real,dimension(:,:,:,:),allocatable communication_ray::coin13_e

real, dimension(:,:,:,:), allocatable :: coin14_e !<

Definition at line 55 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.16 real,dimension(:,:,:,:),allocatable communication_ray::coin13_r

real, dimension(:,:,:,:), allocatable :: coin14_r !<

Definition at line 59 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.17 real,dimension(:,:,:,:),allocatable communication_ray::coin145_e

real, dimension(:,:,:,:), allocatable :: coin146_e !<

Definition at line 65 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.18 real,dimension(:,:,:,:),allocatable communication_ray::coin145_r

real, dimension(:,:,:,:), allocatable :: coin146_r !<

Definition at line 73 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.19 real,dimension(:,:,:,:),allocatable communication_ray::coin235_e

real, dimension(:,:,:,:), allocatable :: coin236_e !<

Definition at line 67 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.20 real,dimension(:,:,:,:),allocatable communication_ray::coin235_r

real, dimension(:,:,:,:), allocatable :: coin236_r !<

Definition at line 75 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.21 real,dimension(:,:,:,:),allocatable communication_ray::coin23_e

real, dimension(:,:,:,:), allocatable :: coin24_e !<

Definition at line 57 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.22 real,dimension(:,:,:,:),allocatable communication_ray::coin23_r

real, dimension(:,:,:,:), allocatable :: coin24_r !<

Definition at line 61 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.23 real,dimension(:,:,:,:),allocatable communication_ray::coin245_e

real, dimension(:,:,:,:), allocatable :: coin246_e !<

Definition at line 69 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.24 real,dimension(:,:,:,:),allocatable communication_ray::coin245_r

real, dimension(:,:,:,:), allocatable :: coin246_r !<

Definition at line 77 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.25 real,dimension(:,:,:,:),allocatable communication_ray::face1_e

real, dimension(:,:,:,:), allocatable :: face2_e !<

Definition at line 19 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.26 real,dimension(:,:,:,:),allocatable communication_ray::face1_r

real, dimension(:,:,:,:), allocatable :: face2_r !<

Definition at line 25 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.27 real,dimension(:,:,:,:),allocatable communication_ray::face3_e

real, dimension(:,:,:,:), allocatable :: face4_e !<

Definition at line 21 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.28 real,dimension(:,:,:,:),allocatable communication_ray::face3_r

real, dimension(:,:,:,:), allocatable :: face4_r !<

Definition at line 27 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.29 real,dimension(:,:,:,:),allocatable communication_ray::face5_e

real, dimension(:,:,:,:), allocatable :: face6_e !<

Definition at line 23 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.8.2.30 real,dimension(:,:,:,:),allocatable communication_ray::face5_r

real, dimension(:,:,:,:), allocatable :: face6_r !<

Definition at line 29 of file para_ray.f90.

Referenced by allocate_array_com_ray(), and communications_ray().

7.9 conduction Module Reference

Contains the variables used for the thermal conduction.

Variables

- real varrel_con

  real :: tolx_con !<

- real vardt_con

  real :: dt_con_exp !<

- real dt_con_imp

  real :: dt_con !<

- logical implicit_con

  If .true.

- integer nitetot_con

  Conduction total number of iterations.

- logical precond_con

  Perform matrix preconditionning if .true.

7.9.1 Detailed Description

Contains the variables used for the thermal conduction.

7.9.2 Variable Documentation

7.9.2.1 real conduction::dt_con_imp

real :: dt_con !<

Definition at line 37 of file modules_con.f90.

Referenced by conduction_imp_gc__interface::cmp_precond_con(), and pasdt_con().

7.9.2.2 logical conduction::implicit_con

If .true. the conduction is computed using an implicit method

Definition at line 39 of file modules_con.f90.

Referenced by conduction_step(), init_conduction(), and pasdt_con().

7.9.2.3 integer conduction::nitetot_con

Conduction total number of iterations.

Definition at line 40 of file modules_con.f90.

Referenced by conduction_imp_gc__interface::cmp_precond_con().

7.9.2.4 logical conduction::precond_con

Perform matrix preconditionning if .true.

Definition at line 41 of file modules_con.f90.

Referenced by conduction_imp_gc__interface::cmp_precond_con(), and init_conduction().

7.9.2.5 real conduction::vardt_con

real :: dt_con_exp !<

Definition at line 35 of file modules_con.f90.

Referenced by conduction_imp_gc__interface::cmp_precond_con(), and init_conduction().

7.9.2.6 real conduction::varrel_con

real :: tolx_con !<

Definition at line 33 of file modules_con.f90.

Referenced by conduction_imp_gc__interface::cmp_precond_con(), and init_conduction().

7.10 const Module Reference

Contains useful constants.

Variables

- real, parameter bigreal = 1.0e+30

  A big real number.

- real, parameter zero = 0.0

  0

- real, parameter one = 1.0

  1

- real, parameter two = 2.0

  2

- real, parameter three = 3.0

  3

- real, parameter four = 4.0

  4

- real, parameter two3rd = 2.0/3.0

  2/3

- real, parameter half = 0.5

  1/2

- real, parameter third = 1.0/3.0

  1/3

- real, parameter forth = 0.25

  1/4

- real, parameter sixth = 1.0/6.0

  1/6

7.10.1 Detailed Description

Contains useful constants.

7.10.2 Variable Documentation

7.10.2.1 real,parameter const::bigreal = 1.0e+30

A big real number.

Definition at line 455 of file modules.f90.

7.10.2.2 real,parameter const::forth = 0.25

1/4

Definition at line 464 of file modules.f90.

Referenced by cmp_mag_flx().

7.10.2.3 real,parameter const::four = 4.0

4

Definition at line 460 of file modules.f90.

Referenced by fast_mhd_speed().

7.10.2.4 real,parameter const::half = 0.5

1/2

Definition at line 462 of file modules.f90.

Referenced by athena_roe(), cmp_mag_flx(), cmpflxm(), cst_gravity(), ctoprim(), fast_mhd_speed(), find_mhd_flux(), find_mhd_flux2(), find_speed_fast(), find_speed_info(), grav_predictor(), hll(), hlld(), hydro_acoustic(), lax_friedrich(), pasdt_mhd(), pt_mass(), trace1d(), trace2d(), trace3d(), update_gravity(), upwind(), and uslope().

7.10.2.5 real,parameter const::one = 1.0

1

Definition at line 457 of file modules.f90.

Referenced by cmpflxm(), ctoprim(), eigen_cons(), find_mhd_flux(), find_mhd_flux2(), hlld(), hydro_acoustic(), and uslope().

7.10.2.6 real,parameter const::sixth = 1.0/6.0

1/6

Definition at line 465 of file modules.f90.

7.10.2.7 real,parameter const::third = 1.0/3.0

1/3

Definition at line 463 of file modules.f90.

7.10.2.8 real,parameter const::three = 3.0

3

Definition at line 459 of file modules.f90.

7.10.2.9 real,parameter const::two = 2.0

2

Definition at line 458 of file modules.f90.

Referenced by athena_roe(), slopes(), trace1d(), trace2d(), and trace3d().

7.10.2.10 real,parameter const::two3rd = 2.0/3.0

2/3

Definition at line 461 of file modules.f90.

7.10.2.11 real,parameter const::zero = 0.0

0

Definition at line 456 of file modules.f90.

Referenced by athena_roe(), cmp_mag_flx(), ctoprim(), fast_mhd_speed(), find_mhd_flux(), hll(), hlld(), hydro_acoustic(), and uslope().

7.11 cputime Module Reference

Contains variables used to compute the cpu time spent is each part of the code.

Variables

- real temps1
  Time 1.
- real temps_ref
  Time spent computing cooling.
- real temps_conduc
  Time spent computing thermal conduction.
- real temps_ray
  Time spent computing radiative transfer.
- real temps_hydro
  Time spent computing hydrodynamics.
- real temps_io
  Time spent performing outputs.
- real temps_comm
  Time spent performing communications.
- real temps_dif
  Time spent computing flux limited diffusion.

7.11.1 Detailed Description

Contains variables used to compute the cpu time spent is each part of the code.

7.11.2 Variable Documentation

7.11.2.1 real cputime::temps1

Time 1.

Definition at line 480 of file modules.f90.

Referenced by compute_cputime(), and heracles().

7.11.2.2 real cputime::temps_comm

Time spent performing communications.

Definition at line 486 of file modules.f90.

Referenced by cal_b3d(), calmoinsun_x3d(), compute_cputime(), evol_ray(), explicite(), gs_mat(), heracles(), mat_prod3d(), and read_params().

7.11.2.3 real cputime::temps_conduc

Time spent computing thermal conduction.

Definition at line 482 of file modules.f90.

Referenced by compute_cputime(), heracles(), and read_params().

7.11.2.4 real cputime::temps_dif

Time spent computing flux limited diffusion.

Definition at line 487 of file modules.f90.

Referenced by compute_cputime(), heracles(), and read_params().

7.11.2.5 real cputime::temps_hydro

Time spent computing hydrodynamics.

Definition at line 484 of file modules.f90.

Referenced by compute_cputime(), heracles(), and read_params().

7.11.2.6 real cputime::temps_io

Time spent performing outputs.

Definition at line 485 of file modules.f90.

Referenced by compute_cputime(), heracles(), and read_params().

7.11.2.7 real cputime::temps_ray

Time spent computing radiative transfer.

Definition at line 483 of file modules.f90.

Referenced by compute_cputime(), heracles(), and read_params().

7.11.2.8 real cputime::temps_ref

Time spent computing cooling.

Definition at line 481 of file modules.f90.

Referenced by compute_cputime(), and read_params().

7.12 diffusion Module Reference

Contains the variables for the FLD.

Variables

- real, dimension(:,:,:), allocatable Er_dif
  FLD radiative energy.
- real varrel_dif
  Maximum variable variations allowed.
- real vardt_dif
  Maximum timestep variation allowed.
- real dt_dif_exp
  FLD explicit timestep.
- real dt_dif_imp
  FLD implicit timestep.
- real dt_dif
  FLD timestep.
- character(len=9) fld_limiter
  FLD flux limiter.
- logical Dif_imp
  Compute FLD implicitely if .true.
- logical precond_dif
  Perform matrix preconditionning if .true.
- integer nitetot_dif
  FLD total number of iterations.

7.12.1 Detailed Description

Contains the variables for the FLD.

7.12.2 Variable Documentation

7.12.2.1 logical diffusion::Dif_imp

Compute FLD implicitely if .true.

Definition at line 39 of file modules_dif.f90.

Referenced by init_diffusion(), and pasdt_dif().

7.12.2.2 real diffusion::dt_dif

FLD timestep.

Definition at line 37 of file modules_dif.f90.

Referenced by pasdt_dif().

7.12.2.3 real diffusion::dt_dif_exp

FLD explicit timestep.

Definition at line 35 of file modules_dif.f90.

7.12.2.4 real diffusion::dt_dif_imp

FLD implicit timestep.

Definition at line 36 of file modules_dif.f90.

Referenced by diffusion_imp__interface::cmp_precond_dif(), and pasdt_dif().

7.12.2.5 real,dimension(:,:,:),allocatable diffusion::Er_dif

FLD radiative energy.

Definition at line 32 of file modules_dif.f90.

Referenced by aff_new(), allocate_array_dif(), diffusion_imp__interface::cmp_precond_dif(), communications_dif(), diffusion_ex(), exp_comobile_ray(), exp_source_ray(), limites_dif(), and restart_new().

7.12.2.6 character(len=9) diffusion::fld_limiter

FLD flux limiter.

Definition at line 38 of file modules_dif.f90.

Referenced by cmp_precond_dif(), diffusion_ex(), exp_source_ray(), init_diffusion(), and mat_prod_-dif().

7.12.2.7 integer diffusion::nitetot_dif

FLD total number of iterations.

Definition at line 41 of file modules_dif.f90.

Referenced by diffusion_imp__interface::cmp_precond_dif(), and init_diffusion().

7.12.2.8 logical diffusion::precond_dif

Perform matrix preconditionning if .true.

Definition at line 40 of file modules_dif.f90.

Referenced by diffusion_imp__interface::cmp_precond_dif(), and init_diffusion().

7.12.2.9 real diffusion::vardt_dif

Maximum timestep variation allowed.

Definition at line 34 of file modules_dif.f90.

Referenced by diffusion_imp__interface::cmp_precond_dif(), and init_diffusion().

7.12.2.10 real diffusion::varrel_dif

Maximum variable variations allowed.

Definition at line 33 of file modules_dif.f90.

Referenced by diffusion_imp__interface::cmp_precond_dif(), and init_diffusion().

7.13 divers Module Reference

Contains various variables.

Variables

- integer, dimension(20) fourdim
  f
- integer, dimension(5) suppdim
  f
- integer Nimp
  Write 000 output file every Nimp timesteps.
- integer i_restart
  File number to restart from.
- integer nsx
  x domain subdivision factor
- integer nsy
  y domain subdivision factor
- integer nsz
  z domain subdivision factor
- integer nrepr
  nr
- integer irepr
  ir
- integer nreprd
  nr
- integer ndata
  nd
- integer nsupp
  Number of additional arrays required.
- integer nout
  Total number of outputs.
- integer iout
  iout
- integer noutd

no

- real, dimension(1000) tout

  Output time.

- real, dimension(1000) trepr

  Restart time.

- real temps

  Time.

- real dt

  Timestep.

- real dtold

  Old timestep.

- real tend

  End time.

- real alpha

  α

- real dt_max

  Maximum timestep.

- real dt_glob

  Global timestep.

- real dt_mhd

  MHD timestep.

- real dt_cfl

  CFL limited timestep.

- real dt_ff

  Gravitational free-fall limited timestep.

- real tcpu_min

  tcpu_min

- real tcpu_max

  tcpu_max

- real tcpu_ini

  tcpu_ini

- real tcpu_last

  tcpu_last en seconde

- real mu
  μ
- character(len=80), dimension(20) datanames
  datanames
- character(len=80), dimension(20) dataunits
  dataunits
- character(len=80), dimension(5) suppnames
  Names of additional arrays.
- character(len=80), dimension(5) suppunits
  Units of additional arrays.
- character(len=80) comments
  comments
- character(len=10) Solver
  Type of Hydro/MHD Riemann solver.
- character(len=5) name_of_job
  Name of job.
- character(len=3) sortie
  Output.
- logical reprise
  Perform restart from previous run if .true.
- logical COM
  com
- logical subcycle
  subcycle
- logical limdx
  limdx
- logical limnite
  limnite
- logical limfr
  limfr
- logical REF
  Include cooling if .true.
- logical DETO
  Include detonation if .true.

- logical CONDUC

  Include thermal conduction if .true.

- logical DIFFU

  Include flux limited diffusion if .true.

- logical RAY

  Include radiative transfer if .true.

- logical HYDRO

  Include hydrodynamics if .true.

- logical GRAV

  Include gravity if .true.

- logical HISTOGRAMME

  Computes histograms if .true.

- logical ISOTHERME

  Uses isothermal equation of state if .true.

- logical STIR

  Include stirring if .true.

- logical verbose

  Prints out extra information for debugging if .true.

- logical swip

  Include swip if .true.

- logical rankine

  Include Rankine if .true.

7.13.1 Detailed Description

Contains various variables.

7.13.2 Variable Documentation

7.13.2.1 real divers::alpha

α

Definition at line 277 of file modules.f90.

7.13.2.2 logical divers::COM

com

Definition at line 297 of file modules.f90.

7.13.2.3 character(len=80) divers::comments

comments

Definition at line 292 of file modules.f90.

Referenced by default_params(), read_params(), and wrt_main_h5().

7.13.2.4 logical divers::CONDUC

Include thermal conduction if .true.

Definition at line 304 of file modules.f90.

Referenced by check_flags(), default_params(), heracles(), pasdt(), print_configuration(), rd_restart_h5(), read_params(), wrt_main_h5(), and wrt_restart_h5().

7.13.2.5 character(len=80),dimension(20) divers::datanames

datanames

Definition at line 288 of file modules.f90.

Referenced by default_params(), read_params(), wrt_film_h5(), wrt_main_h5(), and wrt_output_h5().

7.13.2.6 character(len=80),dimension(20) divers::dataunits

dataunits

Definition at line 289 of file modules.f90.

Referenced by default_params(), read_params(), and wrt_main_h5().

7.13.2.7 logical divers::DETO

Include detonation if .true.

Definition at line 303 of file modules.f90.

Referenced by default_params(), and read_params().

7.13.2.8 logical divers::DIFFU

Include flux limited diffusion if .true.

Definition at line 305 of file modules.f90.

Referenced by aff_new(), check_flags(), default_params(), evol_hydro(), evol_mhd(), exp_source_ray(), heracles(), pasdt(), print_configuration(), rd_restart_h5(), read_params(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.13.2.9 real divers::dt

Timestep.

Definition at line 274 of file modules.f90.

Referenced by aff_new(), cal_b3d(), cmp_precond_con(), conduction_imp_gc__interface::cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), evol_ray(), exp_comobile_ray(), explicite(), gs_mat(), heracles(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), pasdt(), rd_restart_h5(), restart_new(), update_gravity(), wrt_main_h5(), and wrt_restart_h5().

7.13.2.10 real divers::dt_cfl

CFL limited timestep.

Definition at line 281 of file modules.f90.

Referenced by pasdt(), and pasdt_hydro().

7.13.2.11 real divers::dt_ff

Gravitational free-fall limited timestep.

Definition at line 282 of file modules.f90.

Referenced by pasdt(), and pasdt_grav().

7.13.2.12 real divers::dt_glob

Global timestep.

Definition at line 279 of file modules.f90.

7.13.2.13 real divers::dt_max

Maximum timestep.

Definition at line 278 of file modules.f90.

Referenced by default_params(), heracles(), pasdt(), pasdt_dif(), pasdt_ray(), print_configuration(), and read_params().

7.13.2.14 real divers::dt_mhd

MHD timestep.

Definition at line 280 of file modules.f90.

Referenced by pasdt(), and pasdt_mhd().

7.13.2.15 real divers::dtold

Old timestep.

Definition at line 275 of file modules.f90.

Referenced by aff_new(), evol_ray(), pasdt(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_-restart_h5().

7.13.2.16 integer,dimension(20) divers::fourdim

f

Definition at line 256 of file modules.f90.

Referenced by default_params(), and wrt_main_h5().

7.13.2.17 logical divers::GRAV

Include gravity if .true.

Definition at line 308 of file modules.f90.

Referenced by check_flags(), default_params(), evol_hydro(), evol_mhd(), heracles(), pasdt(), print_-configuration(), rd_restart_h5(), read_params(), wrt_main_h5(), and wrt_restart_h5().

7.13.2.18 logical divers::HISTOGRAMME

Computes histograms if .true.

Definition at line 309 of file modules.f90.

Referenced by default_params(), read_params(), and user_output().

7.13.2.19 logical divers::HYDRO

Include hydrodynamics if .true.

Definition at line 307 of file modules.f90.

Referenced by check_flags(), default_params(), pasdt(), print_configuration(), and read_params().

7.13.2.20 integer divers::i_restart

File number to restart from.

Definition at line 259 of file modules.f90.

Referenced by default_params(), rd_restart_h5(), read_params(), and restart_new().

7.13.2.21 integer divers::iout

iout

Definition at line 269 of file modules.f90.

Referenced by aff_new(), gs_mat(), heracles(), init_out(), wrt_main_h5(), wrt_output_h5(), and wrt_-restart_h5().

7.13.2.22 integer divers::irepr

ir

Definition at line 264 of file modules.f90.

Referenced by wrt_main_h5().

7.13.2.23 logical divers::ISOTHERME

Uses isothermal equation of state if .true.

Definition at line 310 of file modules.f90.

Referenced by default_params(), and read_params().

7.13.2.24 logical divers::limdx

limdx

Definition at line 299 of file modules.f90.

7.13.2.25 logical divers::limfr

limfr

Definition at line 301 of file modules.f90.

7.13.2.26 logical divers::limnite

limnite

Definition at line 300 of file modules.f90.

7.13.2.27 real divers::mu

μ

Definition at line 287 of file modules.f90.

Referenced by aff_new(), cal_histogramme(), Energy(), heat_capacity(), init(), rd_restart_h5(), restart_new(), rh(), Temperature(), wrt_main_h5(), and wrt_restart_h5().

7.13.2.28 character(len=5) divers::name_of_job

Name of job.

Definition at line 294 of file modules.f90.

Referenced by default_params(), read_params(), and wrt_main_h5().

7.13.2.29 integer divers::ndata

nd

Definition at line 266 of file modules.f90.

Referenced by default_params(), read_params(), wrt_film_h5(), wrt_main_h5(), and wrt_output_h5().

7.13.2.30 integer divers::Nimp

Write 000 output file every Nimp timesteps.

Definition at line 258 of file modules.f90.

Referenced by default_params(), heracles(), and read_params().

7.13.2.31 integer divers::nout

Total number of outputs.

Definition at line 268 of file modules.f90.

Referenced by init_out(), and print_configuration().

7.13.2.32 integer divers::noutd

no

Definition at line 270 of file modules.f90.

Referenced by default_params(), and wrt_main_h5().

7.13.2.33 integer divers::nrepr

nr

Definition at line 263 of file modules.f90.

Referenced by default_params(), and read_params().

7.13.2.34 integer divers::nreprd

nr

Definition at line 265 of file modules.f90.

Referenced by default_params(), and wrt_main_h5().

7.13.2.35 integer divers::nsupp

Number of additional arrays required.

Definition at line 267 of file modules.f90.

Referenced by allocate_array(), default_params(), fill_new_arrays(), read_params(), and wrt_main_h5().

7.13.2.36 integer divers::nsx

x domain subdivision factor

Definition at line 260 of file modules.f90.

Referenced by conduction_step(), default_params(), diffusion_step(), hydro_step(), print_configuration(), and read_params().

7.13.2.37 integer divers::nsy

y domain subdivision factor

Definition at line 261 of file modules.f90.

Referenced by conduction_step(), default_params(), diffusion_step(), hydro_step(), print_configuration(), and read_params().

7.13.2.38 integer divers::nsz

z domain subdivision factor

Definition at line 262 of file modules.f90.

Referenced by conduction_step(), default_params(), diffusion_step(), hydro_step(), print_configuration(), and read_params().

7.13.2.39 logical divers::rankine

Include Rankine if .true.

Definition at line 314 of file modules.f90.

Referenced by default_params(), and read_params().

7.13.2.40 logical divers::RAY

Include radiative transfer if .true.

Definition at line 306 of file modules.f90.

Referenced by aff_new(), check_flags(), default_params(), evol_hydro(), evol_mhd(), exp_source_ray(), heracles(), pasdt(), print_configuration(), rd_restart_h5(), read_params(), restart_new(), rh(), wrt_main_h5(), and wrt_restart_h5().

7.13.2.41 logical divers::REF

Include cooling if .true.

Definition at line 302 of file modules.f90.

Referenced by default_params(), and read_params().

7.13.2.42 logical divers::reprise

Perform restart from previous run if .true.

Definition at line 296 of file modules.f90.

Referenced by default_params(), heracles(), print_configuration(), and read_params().

7.13.2.43 character(len=10) divers::Solver

Type of Hydro/MHD Riemann solver.

Definition at line 293 of file modules.f90.

Referenced by default_params(), evol_hydro(), and read_params().

7.13.2.44 character(len=3) divers::sortie

Output.

Definition at line 295 of file modules.f90.

7.13.2.45 logical divers::STIR

Include stirring if .true.

Definition at line 311 of file modules.f90.

Referenced by default_params(), and read_params().

7.13.2.46 logical divers::subcycle

subcycle

Definition at line 298 of file modules.f90.

7.13.2.47 integer,dimension(5) divers::suppdim

f

Definition at line 257 of file modules.f90.

Referenced by allocate_array(), default_params(), read_params(), and wrt_main_h5().

7.13.2.48 character(len=80),dimension(5) divers::suppnames

Names of additional arrays.

Definition at line 290 of file modules.f90.

Referenced by default_params(), read_params(), and wrt_main_h5().

7.13.2.49 character(len=80),dimension(5) divers::suppunits

Units of additional arrays.

Definition at line 291 of file modules.f90.

Referenced by default_params(), read_params(), and wrt_main_h5().

7.13.2.50 logical divers::swip

Include swip if .true.

Definition at line 313 of file modules.f90.

7.13.2.51 real divers::tcpu_ini

tcpu_ini

Definition at line 285 of file modules.f90.

Referenced by heracles().

7.13.2.52 real divers::tcpu_last

tcpu_last en seconde

Definition at line 286 of file modules.f90.

Referenced by Arret().

7.13.2.53 real divers::tcpu_max

tcpu_max

Definition at line 284 of file modules.f90.

Referenced by Arret().

7.13.2.54 real divers::tcpu_min

tcpu_min

Definition at line 283 of file modules.f90.

7.13.2.55 real divers::temps

Time.

Definition at line 273 of file modules.f90.

Referenced by aff_new(), cal_histogramme(), film_(), film_hdf(), gs_mat(), heracles(), init(), init_film(), init_out(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.13.2.56 real divers::tend

End time.

Definition at line 276 of file modules.f90.

Referenced by default_params(), heracles(), init_out(), print_configuration(), and read_params().

7.13.2.57 real,dimension(1000) divers::tout

Output time.

Definition at line 271 of file modules.f90.

Referenced by aff_new(), default_params(), gs_mat(), heracles(), init_out(), read_params(), wrt_output_h5(), and wrt_restart_h5().

7.13.2.58 real,dimension(1000) divers::trepr

Restart time.

Definition at line 272 of file modules.f90.

Referenced by default_params(), and read_params().

7.13.2.59 logical divers::verbose

Prints out extra information for debugging if .true.

Definition at line 312 of file modules.f90.

Referenced by cal_b3d(), cal_barycentre(), cal_dif(), cmp_precond_con(), conduction_imp_gc__interface::cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), gc__interface::cmp_precond_gra(), communications(), communications_gen(), communications_ray(), conduction_ex(), default_params(), diffusion_ex(), diffusion_step(), evol_hydro(), evol_mhd(), exp_comobile_ray(), exp_source_ray(), explicite(), film_(), film_hdf(), gs_mat(), heracles(), hydro_step(), init_film(), limites(), limites_dif(), limites_gen(), limites_gra(), limites_ray(), limites_xin(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), pasdt(), pasdt_con(), pasdt_dif(), pasdt_grav(), pasdt_hydro(), pasdt_mhd(), pasdt_ray(), pasdt_ray_exp(), read_params(), solvers_hydro::solver_eos(), solvers_hydro::solver_relax(), update(), and vec_prod().

7.14 divers_ray Module Reference

Contains various variables for the M1 radiative transfer.

Variables

- integer, dimension(6) ordre_des_faces_pour_le_ray

  integer :: nsx !<

- integer nsy

  integer :: nsz !<

- integer slope_type_ray

  Type of slope limiter for radiative transfer.

- integer nitetot_ray

  Total number of iterations in the implicit resolution.

- integer nitemax_ray

  Maximum number of iterations per timestep allowed in the implicit resolution.

- real varrel_ray

  Maximum quantity variation allowed in an implicit timestep.

- real alpha

  real :: alpha_explicite !<

- real varrel_gaz

  real :: dt_ray_imp !< Implicit radiative transfer timestep

- real dt_ray_exp

  Explicit radiative transfer timestep.

- real dt_ray_old

  Old radiative transfer timestep.

- real dt_ray

  Radiative transfer timestep.

- real dt_ray_glob

  Global radiative transfer timestep.

- real rho1

  Density used to compute opacity κ.

- real T1

  Temperature used to compute opacity κ.

- real alpha_var

real :: valp_min !< Minimum allowed value for eigenvalues

- logical IMP_RAY

  Solves radiative transfer implicitely if .true.

- logical EXP_RAY

  Solves radiative transfer explicitely if .true.

- logical rankine

  logical :: precond_ray !< Performs matrix preconditionning if .true.

- logical cal_valp

  Computes M1 eigenvalues if .true.

- logical deja_converge

  character (len=10) :: matrix_inv_solver !< Type of implicit solver ('GS', 'GMRES', 'GS-GMRES', 'GMRES-GS' or 'yoyo')

- character(len=2) rad_trans_model

  Type of radiative transfer model: 'M1' or 'P1'.

7.14.1 Detailed Description

Contains various variables for the M1 radiative transfer.

7.14.2 Variable Documentation

7.14.2.1 real divers_ray::alpha

real :: alpha_explicite !<

Definition at line 48 of file modules_ray.f90.

Referenced by cal_b3d(), gs_mat(), init_ray(), and mat_prod3d().

7.14.2.2 real divers_ray::alpha_var

real :: valp_min !< Minimum allowed value for eigenvalues

Definition at line 58 of file modules_ray.f90.

7.14.2.3 logical divers_ray::cal_valp

Computes M1 eigenvalues if .true.

Definition at line 64 of file modules_ray.f90.

Referenced by cal_b3d(), explicite(), init_ray(), and mat_prod3d().

7.14.2.4 logical divers_ray::deja_converge

character (len=10) :: matrix_inv_solver !< Type of implicit solver ('GS', 'GMRES', 'GS-GMRES', 'GMRES-GS' or 'yoyo')

Definition at line 65 of file modules_ray.f90.

Referenced by evol_ray(), and inversion_gmres().

7.14.2.5 real divers_ray::dt_ray

Radiative transfer timestep.

Definition at line 54 of file modules_ray.f90.

Referenced by heracles(), pasdt(), and pasdt_ray().

7.14.2.6 real divers_ray::dt_ray_exp

Explicit radiative transfer timestep.

Definition at line 52 of file modules_ray.f90.

Referenced by heracles(), pasdt_ray(), and pasdt_ray_exp().

7.14.2.7 real divers_ray::dt_ray_glob

Global radiative transfer timestep.

Definition at line 55 of file modules_ray.f90.

Referenced by pasdt_ray(), and pasdt_ray_exp().

7.14.2.8 real divers_ray::dt_ray_old

Old radiative transfer timestep.

Definition at line 53 of file modules_ray.f90.

Referenced by pasdt_ray().

7.14.2.9 logical divers_ray::EXP_RAY

Solves radiative transfer explicitely if .true.

Definition at line 61 of file modules_ray.f90.

Referenced by evol_ray(), and pasdt_ray().

7.14.2.10 logical divers_ray::IMP_RAY

Solves radiative transfer implicitely if .true.

Definition at line 60 of file modules_ray.f90.

Referenced by pasdt_ray().

7.14.2.11 integer divers_ray::nitemax_ray

Maximum number of iterations per timestep allowed in the implicit resolution.

Definition at line 46 of file modules_ray.f90.

Referenced by gs_mat(), and init_ray().

7.14.2.12 integer divers_ray::nitetot_ray

Total number of iterations in the implicit resolution.

Definition at line 45 of file modules_ray.f90.

7.14.2.13 integer divers_ray::nsy

integer :: nsz !<

Definition at line 42 of file modules_ray.f90.

7.14.2.14 integer,dimension(6) divers_ray::ordre_des_faces_pour_le_ray

integer :: nsx !<

Definition at line 40 of file modules_ray.f90.

Referenced by init_ray(), and limites_ray().

7.14.2.15 character (len=2) divers_ray::rad_trans_model

Type of radiative transfer model: 'M1' or 'P1'.

Definition at line 67 of file modules_ray.f90.

Referenced by cal_b3d(), cal_Dedd(), explicite(), gs_mat(), init_ray(), mat_prod3d(), and print_configuration().

7.14.2.16 logical divers_ray::rankine

logical :: precond_ray !< Performs matrix preconditionning if .true.

Definition at line 62 of file modules_ray.f90.

7.14.2.17 real divers_ray::rho1

Density used to compute opacity κ.

Definition at line 56 of file modules_ray.f90.

Referenced by cal_kappa().

7.14.2.18 integer divers_ray::slope_type_ray

Type of slope limiter for radiative transfer.

Definition at line 44 of file modules_ray.f90.

Referenced by cal_b3d(), explicite(), gs_mat(), init_ray(), mat_prod3d(), and print_configuration().

7.14.2.19 real divers_ray::T1

Temperature used to compute opacity κ.

Definition at line 57 of file modules_ray.f90.

7.14.2.20 real divers_ray::varrel_gaz

real :: dt_ray_imp !< Implicit radiative transfer timestep

Definition at line 50 of file modules_ray.f90.

Referenced by init_ray(), and pasdt_ray().

7.14.2.21 real divers_ray::varrel_ray

Maximum quantity variation allowed in an implicit timestep.

Definition at line 47 of file modules_ray.f90.

Referenced by init_ray(), pasdt_ray(), and print_configuration().

7.15 Etat_GDS Module Reference

Contains the interface values arrays.

Variables

- real, dimension(:,:,:,:,:), allocatable uS
  Value of the gas velocity at the cell interface.
- real, dimension(:,:,:,:,:), allocatable FxS
  Value of the passive scalar at the cell interface.
- real, dimension(:,:,:,:), allocatable rhoS
  Value of the gas momentum at the cell interface.
- real, dimension(:,:,:,:), allocatable pS
  Value of the gas pressure at the cell interface.
- real, dimension(:,:,:,:), allocatable eS
  Value of the total gas energy at the cell interface.

7.15.1 Detailed Description

Contains the interface values arrays.

7.15.2 Variable Documentation

7.15.2.1 real,dimension(:,:,:,:),allocatable Etat_GDS::eS

Value of the total gas energy at the cell interface.

Definition at line 158 of file modules.f90.

7.15.2.2 real,dimension(:,:,:,:,:),allocatable Etat_GDS::FxS

Value of the passive scalar at the cell interface.

Definition at line 155 of file modules.f90.

7.15.2.3 real,dimension(:,:,:,:),allocatable Etat_GDS::pS

Value of the gas pressure at the cell interface.

Definition at line 157 of file modules.f90.

7.15.2.4 real,dimension(:,:,:,:),allocatable Etat_GDS::rhoS

Value of the gas momentum at the cell interface.

Definition at line 156 of file modules.f90.

7.15.2.5 real,dimension(:,:,:,:,:),allocatable Etat_GDS::uS

Value of the gas velocity at the cell interface.

Definition at line 154 of file modules.f90.

Referenced by evol_mhd().

7.16 gammas Module Reference

Contains the variables determining the heat capacity ratio and thermodynamic relations.

Variables

- real mu_reac

  μ_{reac}

- real gamma

  Ratio of specific heats γ.

- real gamma1

  γ_1

- real gamma2

  γ_1

- real g1

  $\frac{\gamma - 1}{2\gamma}$

- real g2

  $\frac{\gamma + 1}{2\gamma}$

- real g3

  $\frac{2\gamma}{\gamma - 1}$

- real g4

  $\frac{2}{\gamma - 1}$

- real g5

  $\frac{2}{\gamma + 1}$

- real g6

  $\frac{\gamma - 1}{\gamma + 1}$

- real g7

  $\frac{\gamma - 1}{2}$

- real g8

  $\gamma - 1$

- real Cs_iso

  Isothermal sound speed.

- real Cs_iso2

  Isothermal sound speed 2.

7.16.1 Detailed Description

Contains the variables determining the heat capacity ratio and thermodynamic relations.

7.16.2 Variable Documentation

7.16.2.1 real gammas::Cs_iso

Isothermal sound speed.

Definition at line 412 of file modules.f90.

Referenced by solvers_hydro::solver_iso().

7.16.2.2 real gammas::Cs_iso2

Isothermal sound speed 2.

Definition at line 413 of file modules.f90.

Referenced by evol_hydro(), and solvers_hydro::solver_iso().

7.16.2.3 real gammas::g1

$$\frac{\gamma - 1}{2\gamma}$$

Definition at line 404 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), dimensions(), solvers_hydro::guessp(), solvers_hydro::prefun(), and solvers_hydro::sample().

7.16.2.4 real gammas::g2

$$\frac{\gamma + 1}{2\gamma}$$

Definition at line 405 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), dimensions(), solvers_hydro::prefun(), solvers_hydro::sample(), and solvers_hydro::solver_cc().

7.16.2.5 real gammas::g3

$$\frac{2\gamma}{\gamma - 1}$$

Definition at line 406 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), dimensions(), solvers_hydro::guessp(), and solvers_hydro::sample().

7.16.2.6 real gammas::g4

$$\frac{2}{\gamma - 1}$$

Definition at line 407 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), dimensions(), solvers_hydro::guessp(), solvers_hydro::prefun(), solvers_hydro::sample(), solvers_hydro::solver_exact(), and solvers_hydro::solver_g().

7.16.2.7 real gammas::g5

$$\frac{2}{\gamma+1}$$

Definition at line 408 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), dimensions(), solvers_hydro::guessp(), solvers_hydro::prefun(), and solvers_hydro::sample().

7.16.2.8 real gammas::g6

$$\frac{\gamma-1}{\gamma+1}$$

Definition at line 409 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), dimensions(), solvers_hydro::guessp(), solvers_hydro::prefun(), and solvers_hydro::sample().

7.16.2.9 real gammas::g7

$$\frac{\gamma-1}{2}$$

Definition at line 410 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), dimensions(), solvers_hydro::guessp(), and solvers_hydro::sample().

7.16.2.10 real gammas::g8

$$\gamma-1$$

Definition at line 411 of file modules.f90.

Referenced by solvers_hydro::cal_gamma(), and dimensions().

7.16.2.11 real gammas::gamma

Ratio of specific heats γ.

Definition at line 401 of file modules.f90.

Referenced by aff_new(), solvers_hydro::cal_gamma(), cal_histogramme(), ctoprim(), dimensions(), eigen_cons(), eigenvalues(), Energy(), evol_hydro(), evol_mhd(), fast_mhd_speed(), find_mhd_flux(), find_mhd_flux2(), find_speed_fast(), find_speed_info(), heat_capacity(), hlld(), hydro_acoustic(), init(), pasdt_mhd(), Pressure(), rd_restart_h5(), restart_new(), rh(), solvers_hydro::sample(), solvers_hydro::solver_acoustic(), solvers_hydro::solver_cc(), solvers_hydro::solver_exact(), solvers_hydro::solver_g(), solvers_hydro::solver_relax(), sortie_ecran(), Sound_speed(), Temperature(), trace1d(), trace2d(), trace3d(), wrt_main_h5(), and wrt_restart_h5().

7.16.2.12 real gammas::gamma1

γ_1

Definition at line 402 of file modules.f90.

Referenced by solvers_hydro::cal_gamma().

7.16.2.13 real gammas::gamma2

γ_1

Definition at line 403 of file modules.f90.

Referenced by solvers_hydro::cal_gamma().

7.16.2.14 real gammas::mu_reac

μ_{reac}

Definition at line 400 of file modules.f90.

Referenced by solvers_hydro::cal_gamma().

7.17 geom Module Reference

Contains all the grid variables and coordinates.

Variables

- integer, dimension(3) nshift_gr
  nshift
- real, dimension(:,:,:,:), allocatable ds
  Surface element.
- real, dimension(:,:,:), allocatable dv
  Volume element.
- real, dimension(:,:), allocatable x
  Local cell interface coordinate array.
- real, dimension(:,:), allocatable x_glob
  Global cell interface coordinate array.
- real, dimension(:,:), allocatable dx
  Line element.
- real, dimension(:,:), allocatable dxc
  Cell centred line element.
- real, dimension(:,:), allocatable xcloc
  Local cell centre coordinate array.
- real, dimension(3) shift_gr
  shiftgr
- real, dimension(3) Lbox
  Size of computational box.
- real, dimension(3) box_min
  Negative coordinate offset.
- real, dimension(3) box_max
  Positive coordinate offset.
- character(len=5), dimension(3) geom_dir
  Geometry directions.
- character(len=5) geometrie
  Grid geometry.
- logical Cartesien

Cartesian grid if .true.

- logical Cylindrique

  Cylindrical grid if .true.

- logical Spherique

  Spherical grid if .true.

7.17.1 Detailed Description

Contains all the grid variables and coordinates.

7.17.2 Variable Documentation

7.17.2.1 real,dimension(3) geom::box_max

Positive coordinate offset.

Definition at line 88 of file modules.f90.

7.17.2.2 real,dimension(3) geom::box_min

Negative coordinate offset.

Definition at line 87 of file modules.f90.

7.17.2.3 logical geom::Cartesien

Cartesian grid if .true.

Definition at line 91 of file modules.f90.

Referenced by cal_geom(), check_flags(), cmpflxm(), evol_hydro(), print_configuration(), pt_mass(), trace1d(), trace2d(), trace3d(), and update().

7.17.2.4 logical geom::Cylindrique

Cylindrical grid if .true.

Definition at line 92 of file modules.f90.

Referenced by cal_geom(), check_flags(), cmpflxm(), print_configuration(), pt_mass(), trace1d(), trace2d(), trace3d(), and update().

7.17.2.5 real,dimension(:,:,:,:),allocatable geom::ds

Surface element.

Definition at line 78 of file modules.f90.

Referenced by cal_b3d(), cal_geom(), cmp_precond_con(), cmp_precond_dif(), cmpflxm(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), exp_comobile_ray(), explicite(), gs_mat(), mat_prod3d(), Mat_prod_con(), and mat_prod_dif().

7.17.2.6 real,dimension(:,:,:),allocatable geom::dv

Volume element.

Definition at line 79 of file modules.f90.

Referenced by cal_b3d(), cal_geom(), cmp_precond_con(), cmp_precond_dif(), cmpflxm(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), exp_comobile_ray(), explicite(), gs_mat(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), and update().

7.17.2.7 real,dimension(:,:),allocatable geom::dx

Line element.

Definition at line 82 of file modules.f90.

Referenced by cal_b3d(), cal_geom(), cmp_precond_con(), cmp_precond_dif(), cmpflxm(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), explicite(), gs_mat(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), trace1d(), trace2d(), trace3d(), and update().

7.17.2.8 real,dimension(:,:),allocatable geom::dxc

Cell centred line element.

Definition at line 83 of file modules.f90.

Referenced by cal_b3d(), cal_geom(), cmp_precond_con(), cmp_precond_dif(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), explicite(), gs_mat(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), pente_minmod(), and slopes().

7.17.2.9 character(len=5),dimension(3) geom::geom_dir

Geometry directions.

Definition at line 89 of file modules.f90.

Referenced by cal_b3d(), check_flags(), cmp_precond_dif(), diffusion_ex(), evol_hydro(), evol_mhd(), exp_comobile_ray(), exp_source_ray(), explicite(), grav_predictor(), gs_mat(), mat_prod3d(), mat_prod_dif(), pasdt_con(), pasdt_dif(), pasdt_hydro(), pasdt_mhd(), pasdt_ray_exp(), print_configuration(), and update_gravity().

7.17.2.10 character(len=5) geom::geometrie

Grid geometry.

Definition at line 90 of file modules.f90.

Referenced by aff_new(), cal_geom(), check_flags(), print_configuration(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.17.2.11 real,dimension(3) geom::Lbox

Size of computational box.

Definition at line 86 of file modules.f90.

Referenced by gc__interface::cmp_precond_gra(), and print_configuration().

7.17.2.12 integer,dimension(3) geom::nshift_gr

nshift

Definition at line 77 of file modules.f90.

Referenced by aff_new(), allocate_array(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.17.2.13 real,dimension(3) geom::shift_gr

shiftgr

Definition at line 85 of file modules.f90.

Referenced by aff_new(), allocate_array(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.17.2.14 logical geom::Spherique

Spherical grid if .true.

Definition at line 93 of file modules.f90.

Referenced by cal_geom(), check_flags(), cmpflxm(), grav_predictor(), print_configuration(), pt_mass(), trace1d(), trace2d(), trace3d(), update(), and update_gravity().

7.17.2.15 real,dimension(:,:),allocatable geom::x

Local cell interface coordinate array.

Definition at line 80 of file modules.f90.

Referenced by aff_new(), allocate_array(), cal_b3d(), cal_barycentre(), cmp_precond_con(), conduction_imp_gc__interface::cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), conduction_ex(), cst_gravity(), def_geom(), diffusion_ex(), evol_hydro(), evol_mhd(), explicite(), gs_mat(), init(), limites_gra(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), mat_prod_gra(), pasdt_con(), pasdt_dif(), pasdt_hydro(), pasdt_mhd(), pasdt_ray_exp(), pt_mass(), rd_restart_h5(), restart_new(), sortie_ecran(), update_gravity(), and wrt_restart_h5().

7.17.2.16 real,dimension(:,:),allocatable geom::x_glob

Global cell interface coordinate array.

Definition at line 81 of file modules.f90.

Referenced by aff_new(), allocate_array(), init(), print_configuration(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.17.2.17 real,dimension(:,:),allocatable geom::xcloc

Local cell centre coordinate array.

Definition at line 84 of file modules.f90.

Referenced by cal_b3d(), cal_geom(), cmp_precond_con(), cmp_precond_dif(), cmpflxm(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), exp_comobile_ray(), exp_source_ray(), explicite(), grav_predictor(), gs_mat(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), trace1d(), trace2d(), trace3d(), and update().

7.18 gravity Module Reference

Contains all the gravity variables.

Variables

- integer gravity_type
- integer ISOLE
- real, dimension(:,:,:), allocatable phi
  Gravitational potential Φ.
- real, dimension(:,:,:), allocatable phiold
  Old graviational potential $\Phi_{\rm old}$.
- real, dimension(4) gravity_params
  Gravity parameters.

7.18.1 Detailed Description

Contains all the gravity variables.

7.18.2 Variable Documentation

7.18.2.1 real,dimension(4) gravity::gravity_params

Gravity parameters.

Definition at line 21 of file modules_gra.f90.

Referenced by cst_gravity(), init_gravity(), and pt_mass().

7.18.2.2 integer gravity::gravity_type

Definition at line 17 of file modules_gra.f90.

Referenced by init_gravity(), and poisson().

7.18.2.3 integer gravity::ISOLE

Definition at line 18 of file modules_gra.f90.

Referenced by gc__interface::cmp_precond_gra(), init_gravity(), and limites_gra().

7.18.2.4 real,dimension(:,:,:),allocatable gravity::phi

Gravitational potential Φ.

Definition at line 19 of file modules_gra.f90.

Referenced by allocate_array_gra(), gc__interface::cmp_precond_gra(), cst_gravity(), evol_hydro(), evol_mhd(), poisson(), pt_mass(), and update_gravity().

7.18.2.5 real,dimension(:,:,:),allocatable gravity::phiold

Old graviational potential $\Phi_{\rm old}$.

Definition at line 20 of file modules_gra.f90.

Referenced by allocate_array_gra(), poisson(), and update_gravity().

7.19 histo Module Reference

Contains variables for histogram computations.

Variables

- integer Npoint
  Npoint.
- real, dimension(:), allocatable hP
  hp
- real, dimension(:), allocatable hN
  hn
- real, dimension(:), allocatable hT
  ht
- real, dimension(:), allocatable P
  p
- real, dimension(:), allocatable N
  n
- real, dimension(:), allocatable T
  t
- real, dimension(:), allocatable hPg
  hPg
- real, dimension(:), allocatable hNg
  hNg
- real, dimension(:), allocatable hTg
  hTg
- real Pmin
  Pmin.
- real Pmax
  Pmax.
- real r_P
  r_P
- real lr_P
  lr_P
- real Nmin

Nmin.

- real Nmax
  Nmax.
- real r_N
  r_N
- real lr_N
  lr_N
- real Tmin
  Tmin.
- real Tmax
  Tmax.
- real r_T
  r_T
- real lr_T
  lr_T

7.19.1 Detailed Description

Contains variables for histogram computations.

7.19.2 Variable Documentation

7.19.2.1 real,dimension(:),allocatable histo::hN

hn

Definition at line 21 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.2 real,dimension(:),allocatable histo::hNg

hNg

Definition at line 27 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.3 real,dimension(:),allocatable histo::hP

hp

Definition at line 20 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.4 real,dimension(:),allocatable histo::hPg

hPg

Definition at line 26 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.5 real,dimension(:),allocatable histo::hT

ht

Definition at line 22 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.6 real,dimension(:),allocatable histo::hTg

hTg

Definition at line 28 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.7 real histo::lr_N

lr_N

Definition at line 36 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.8 real histo::lr_P

lr_P

Definition at line 32 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.9 real histo::lr_T

lr_T

Definition at line 40 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.10 real,dimension(:),allocatable histo::N

n

Definition at line 24 of file histogramme.f90.

Referenced by init_histo().

7.19.2.11 real histo::Nmax

Nmax.

Definition at line 34 of file histogramme.f90.

Referenced by init_histo().

7.19.2.12 real histo::Nmin

Nmin.

Definition at line 33 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.13 integer histo::Npoint

Npoint.

Definition at line 19 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.14 real,dimension(:),allocatable histo::P

p

Definition at line 23 of file histogramme.f90.

Referenced by init_histo().

7.19.2.15 real histo::Pmax

Pmax.

Definition at line 30 of file histogramme.f90.

Referenced by init_histo().

7.19.2.16 real histo::Pmin

Pmin.

Definition at line 29 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.19.2.17 real histo::r_N

r_N

Definition at line 35 of file histogramme.f90.

Referenced by init_histo().

7.19.2.18 real histo::r_P

r_P

Definition at line 31 of file histogramme.f90.

Referenced by init_histo().

7.19.2.19 real histo::r_T

r_T

Definition at line 39 of file histogramme.f90.

Referenced by init_histo().

7.19.2.20 real,dimension(:),allocatable histo::T

t

Definition at line 25 of file histogramme.f90.

Referenced by init_histo().

7.19.2.21 real histo::Tmax

Tmax.

Definition at line 38 of file histogramme.f90.

Referenced by init_histo().

7.19.2.22 real histo::Tmin

Tmin.

Definition at line 37 of file histogramme.f90.

Referenced by cal_histogramme(), and init_histo().

7.20 mfilm Module Reference

Contains variables used to create movies.

Variables

- integer ifilm

  Current number of film output.

- real dt_film

  Timestep between film outputs.

- real t_film

  t_film

- real t_start

  t_start

- real t_end

  t_end

- logical FILM

  Creates movie outputs if .true.

- integer nl1

  nl1

- integer nl2

  nl2

- integer nl3

  nl3

7.20.1 Detailed Description

Contains variables used to create movies.

7.20.2 Variable Documentation

7.20.2.1 real mfilm::dt_film

Timestep between film outputs.

Definition at line 20 of file film.f90.

Referenced by film_(), film_hdf(), and init_film().

7.20.2.2 logical mfilm::FILM

Creates movie outputs if .true.

Definition at line 24 of file film.f90.

Referenced by default_params(), heracles(), and read_params().

7.20.2.3 integer mfilm::ifilm

Current number of film output.

Definition at line 19 of file film.f90.

Referenced by film_(), and film_hdf().

7.20.2.4 integer mfilm::nl1

nl1

Definition at line 25 of file film.f90.

Referenced by init_film().

7.20.2.5 integer mfilm::nl2

nl2

Definition at line 26 of file film.f90.

Referenced by init_film().

7.20.2.6 integer mfilm::nl3

nl3

Definition at line 27 of file film.f90.

Referenced by init_film().

7.20.2.7 real mfilm::t_end

t_end

Definition at line 23 of file film.f90.

Referenced by init_film().

7.20.2.8 real mfilm::t_film

t_film

Definition at line 21 of file film.f90.

Referenced by film_(), film_hdf(), init_film(), and wrt_film_h5().

7.20.2.9 real mfilm::t_start

t_start

Definition at line 22 of file film.f90.

Referenced by init_film().

7.21 para Module Reference

Contains variables defining the MPI parameters of the simulation.

Variables

- integer, dimension(:,:,:), allocatable grid_cpu
  Number of cpus in the x,y,z, directions.
- integer, dimension(:,:), allocatable x_start_cpu
  Starting x,y,z coordinates which cpu holds.
- integer, dimension(:,:), allocatable x_end_cpu
  Ending x,y,z coordinates which cpu holds.
- integer, dimension(:), allocatable lim_x
  lim_x
- integer, dimension(:), allocatable lim_y
  lim_y
- integer, dimension(:), allocatable lim_z
  lim_z
- integer, dimension(-1:1,-1:1,-1:1) voisin
  Cpu neighbours.
- integer, dimension(3) grid_pos
  grid_pos
- integer, dimension(3) x_start
  x_start
- integer, dimension(3) x_end
  x_end
- integer ncpu
  Total number of cpus.
- integer mype
  Cpu rank.
- integer ncpu_x
  Number of cpus in the x direction.
- integer ncpu_y
  Number of cpus in the y direction.
- integer ncpu_z
  Number of cpus in the z direction.

7.21.1 Detailed Description

Contains variables defining the MPI parameters of the simulation.

7.21.2 Variable Documentation

7.21.2.1 integer,dimension(:,:,:),allocatable para::grid_cpu

Number of cpus in the x,y,z, directions.

Definition at line 96 of file para.f90.

Referenced by init_para().

7.21.2.2 integer,dimension(3) para::grid_pos

grid_pos

Definition at line 103 of file para.f90.

Referenced by init_para().

7.21.2.3 integer,dimension(:),allocatable para::lim_x

lim_x

Definition at line 99 of file para.f90.

Referenced by init_para().

7.21.2.4 integer,dimension(:),allocatable para::lim_y

lim_y

Definition at line 100 of file para.f90.

Referenced by init_para().

7.21.2.5 integer,dimension(:),allocatable para::lim_z

lim_z

Definition at line 101 of file para.f90.

Referenced by init_para().

7.21.2.6 integer para::mype

Cpu rank.

Definition at line 107 of file para.f90.

Referenced by aff_new(), Arret(), cal_histogramme(), check_flags(), check_size(), cl_ana(), cl_ana_gen(), cl_ana_gen_ray(), cl_ana_ray(), cg__interface::cmp_precond(), conduction_imp_gc__interface::cmp_precond_con(), diffusion_imp__interface::cmp_precond_dif(), cpu_(), Energy(), evol_hydro(), evol_ray(),

explicite(), film_(), film_hdf(), gmres(), gs_mat(), heat_capacity(), heracles(), init_diffusion(), init_histo(), init_out(), init_para(), init_phys(), init_ray(), inversion_gmres(), mk_dir_out(), monitoring(), output_data(), pasdt(), pasdt_ray(), pasdt_ray_exp(), poisson(), PrimeDec(), rd_restart_h5(), read_params(), restart_new(), rh(), set_units_out(), solvers_hydro::solver_eos(), solvers_hydro::solver_exact(), solvers_hydro::solver_g(), solvers_hydro::solver_relaxation(), solvers_hydro::starpu_iso(), uslope(), wrt_film_h5(), wrt_main_h5(), wrt_output_h5(), and wrt_restart_h5().

7.21.2.7 integer para::ncpu

Total number of cpus.

Definition at line 106 of file para.f90.

Referenced by aff_new(), cpu_(), film_(), heracles(), init_para(), print_configuration(), restart_new(), and wrt_main_h5().

7.21.2.8 integer para::ncpu_x

Number of cpus in the x direction.

Definition at line 108 of file para.f90.

Referenced by cpu_(), default_params(), init_para(), and read_params().

7.21.2.9 integer para::ncpu_y

Number of cpus in the y direction.

Definition at line 109 of file para.f90.

Referenced by cpu_(), default_params(), init_para(), and read_params().

7.21.2.10 integer para::ncpu_z

Number of cpus in the z direction.

Definition at line 110 of file para.f90.

Referenced by cpu_(), default_params(), init_para(), and read_params().

7.21.2.11 integer,dimension(-1:1,-1:1,-1:1) para::voisin

Cpu neighbours.

Definition at line 102 of file para.f90.

Referenced by communications(), communications_gen(), communications_ray(), init_para(), and mat_prod3d().

7.21.2.12 integer,dimension(3) para::x_end

x_end

Definition at line 105 of file para.f90.

Referenced by aff_new(), init_para(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.21.2.13 integer,dimension(:,:),allocatable para::x_end_cpu

Ending x,y,z coordinates which cpu holds.

Definition at line 98 of file para.f90.

Referenced by init_para().

7.21.2.14 integer,dimension(3) para::x_start

x_start

Definition at line 104 of file para.f90.

Referenced by aff_new(), gs_mat(), init(), init_para(), pasdt_ray(), rd_restart_h5(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.21.2.15 integer,dimension(:,:),allocatable para::x_start_cpu

Starting x,y,z coordinates which cpu holds.

Definition at line 97 of file para.f90.

Referenced by film_(), and init_para().

7.22 param_ini Module Reference

Contains variables used for initialisation of the simulation.

Variables

- real rho1
  ρ_1
- real rho2
  ρ_2
- real rho3
  ρ_3
- real rho4
  ρ_4
- real p1
  p_1
- real p2
  p_2
- real p3
  p_3
- real p4
  p_4
- real T1
  T_1
- real T2
  T_2
- real T3
  T_3
- real T4
  T_4
- real E1
  E_1
- real E2
  E_2
- real E3

E_3

- real E4

  E_4

- real u1

  u_1

- real u2

  u_2

- real u3

  u_3

- real u4

  u_4

- real v1

  v_1

- real v2

  v_2

- real v3

  v_3

- real v4

  v_4

- real cs1

  cs_1

- real x1

  x_1

- real x2

  x_2

- real d1

  d_1

- real d2

  d_1

- real ww

  $?$

- real B0

  B_0

- real Temperature_isotherme

  Isothermal temperature.

7.22.1 Detailed Description

Contains variables used for initialisation of the simulation.

7.22.2 Variable Documentation

7.22.2.1 real param_ini::B0

B_0

Definition at line 359 of file modules.f90.

7.22.2.2 real param_ini::cs1

cs_1

Definition at line 353 of file modules.f90.

7.22.2.3 real param_ini::d1

d_1

Definition at line 356 of file modules.f90.

7.22.2.4 real param_ini::d2

d_1

Definition at line 357 of file modules.f90.

7.22.2.5 real param_ini::E1

E_1

Definition at line 341 of file modules.f90.

7.22.2.6 real param_ini::E2

E_2

Definition at line 342 of file modules.f90.

7.22.2.7 real param_ini::E3

E_3

Definition at line 343 of file modules.f90.

7.22.2.8 real param_ini::E4

E_4

Definition at line 344 of file modules.f90.

7.22.2.9 real param_ini::p1

p_1

Definition at line 333 of file modules.f90.

7.22.2.10 real param_ini::p2

p_2

Definition at line 334 of file modules.f90.

7.22.2.11 real param_ini::p3

p_3

Definition at line 335 of file modules.f90.

7.22.2.12 real param_ini::p4

p_4

Definition at line 336 of file modules.f90.

7.22.2.13 real param_ini::rho1

ρ_1

Definition at line 329 of file modules.f90.

7.22.2.14 real param_ini::rho2

ρ_2

Definition at line 330 of file modules.f90.

7.22.2.15 real param_ini::rho3

ρ_3

Definition at line 331 of file modules.f90.

7.22.2.16 real param_ini::rho4

ρ_4

Definition at line 332 of file modules.f90.

7.22.2.17 real param_ini::T1

T_1

Definition at line 337 of file modules.f90.

7.22.2.18 real param_ini::T2

T_2

Definition at line 338 of file modules.f90.

7.22.2.19 real param_ini::T3

T_3

Definition at line 339 of file modules.f90.

7.22.2.20 real param_ini::T4

T_4

Definition at line 340 of file modules.f90.

7.22.2.21 real param_ini::Temperature_isotherme

Isothermal temperature.

Definition at line 360 of file modules.f90.

Referenced by Temperature().

7.22.2.22 real param_ini::u1

u_1

Definition at line 345 of file modules.f90.

7.22.2.23 real param_ini::u2

u_2

Definition at line 346 of file modules.f90.

7.22.2.24 real param_ini::u3

u_3

Definition at line 347 of file modules.f90.

7.22.2.25 real param_ini::u4

u_4

Definition at line 348 of file modules.f90.

7.22.2.26 real param_ini::v1

v_1

Definition at line 349 of file modules.f90.

7.22.2.27 real param_ini::v2

v_2

Definition at line 350 of file modules.f90.

7.22.2.28 real param_ini::v3

v_3

Definition at line 351 of file modules.f90.

7.22.2.29 real param_ini::v4

v_4

Definition at line 352 of file modules.f90.

7.22.2.30 real param_ini::ww

?

Definition at line 358 of file modules.f90.

7.22.2.31 real param_ini::x1

x_1

Definition at line 354 of file modules.f90.

7.22.2.32 real param_ini::x2

x_2

Definition at line 355 of file modules.f90.

7.23 parameters Module Reference

Holds all the general simulation parameters.

Variables

- integer, dimension(:,:), allocatable nx_cpu
  Number of cells in the x,y,z directions per cpu.
- integer, dimension(3) nx_glob
  Global simulation dimensions.
- integer, dimension(3) nx
  Local simulation dimensions.
- integer, dimension(3) nxmin
  Lower limit for arrays.
- integer, dimension(3) nxmax
  Upper limit for arrays.
- integer ndim
  Number of dimensions.
- integer nx_max
  Highest nx.
- integer nx_glob_max
  Global highest nx.
- integer Nbuf
  Number of buffer/ghost cells.
- integer N_vit
  Number of velocity components.
- integer nvar
  Number of hydro/MHD variables.
- integer nFx
  Number of passive scalars.
- integer nvar_ray
  Total number of radiative variables.
- integer slope_type
  Type of slope limiter for Hydro and MHD.
- real, parameter Pi = 3.1415926535898

π

- real, parameter quatre_Pi = 4.∗pi

  4π

- real, parameter smallc = 1.d-30

  Dimensioned small constant.

- real, parameter smallr = 1.d-40

  Dimensioned small real number.

- character(len=10) riemann

  Type of Riemann solver.

- character(len=10) riemann2d

  Type of Riemann solver 2.

7.23.1 Detailed Description

Holds all the general simulation parameters.

7.23.2 Variable Documentation

7.23.2.1 integer parameters::N_vit

Number of velocity components.

Definition at line 30 of file modules.f90.

Referenced by allocate_array(), check_flags(), communications(), init_para(), limites(), pasdt_mhd(), rd_restart_h5(), wrt_main_h5(), and wrt_restart_h5().

7.23.2.2 integer parameters::Nbuf

Number of buffer/ghost cells.

Definition at line 29 of file modules.f90.

Referenced by allocate_array(), allocate_array_com(), allocate_array_com_gen(), allocate_array_com_ray(), check_flags(), communications(), communications_gen(), communications_ray(), default_params(), limites(), limites_gen(), limites_ray(), rd_restart_h5(), read_params(), wrt_main_h5(), and wrt_restart_h5().

7.23.2.3 integer parameters::ndim

Number of dimensions.

Definition at line 26 of file modules.f90.

Referenced by aff_new(), allocate_array(), allocate_array_com(), allocate_array_com_gen(), allocate_array_com_ray(), allocate_array_dif(), allocate_array_ray(), cal_Dedd(), cal_geom(), cal_histogramme(),

cal_x3d(), calmoinsun_x3d(), check_flags(), cg__interface::cmp_precond(), cmp_precond_con(), cmp_precond_dif(), gc__interface::cmp_precond_gra(), communications(), communications_gen(), communications_ray(), conduction_ex(), cpu_(), cst_gravity(), def_geom(), default_params(), diffusion_ex(), evol_hydro(), evol_mhd(), exp_comobile_ray(), exp_source_ray(), explicite(), grav_predictor(), gs_mat(), heracles(), init_conduction(), init_film(), init_para(), init_ray(), limites(), limites_gen(), limites_gra(), limites_ray(), limites_xin(), Mat_prod_con(), mat_prod_dif(), pasdt_con(), pasdt_dif(), pasdt_hydro(), pasdt_mhd(), pasdt_ray(), pasdt_ray_exp(), pente_minmod(), print_configuration(), pt_mass(), rd_restart_h5(), read_params(), restart_new(), slopes(), solvers_hydro::solver_acoustic(), solvers_hydro::solver_cc(), solvers_hydro::solver_eos(), solvers_hydro::solver_exact(), solvers_hydro::solver_g(), solvers_hydro::solver_iso(), solvers_hydro::solver_relax(), update_gravity(), wrt_main_h5(), and wrt_restart_h5().

7.23.2.4 integer parameters::nFx

Number of passive scalars.

Definition at line 32 of file modules.f90.

Referenced by allocate_array(), and evol_hydro().

7.23.2.5 integer parameters::nvar

Number of hydro/MHD variables.

Definition at line 31 of file modules.f90.

Referenced by allocate_array_com(), athena_roe(), communications(), evol_mhd(), find_mhd_flux(), find_mhd_flux2(), hll(), hydro_acoustic(), and init_para().

7.23.2.6 integer parameters::nvar_ray

Total number of radiative variables.

Definition at line 33 of file modules.f90.

Referenced by allocate_array_com_ray(), allocate_gmres(), cal_x3d(), calmoinsun_x3d(), communications_ray(), explicite(), gs_mat(), init_para(), initVarGmres(), and inversion_gmres().

7.23.2.7 integer,dimension(3) parameters::nx

Local simulation dimensions.

Definition at line 23 of file modules.f90.

Referenced by aff_new(), allocate_array(), allocate_array_com(), allocate_array_com_gen(), allocate_array_com_ray(), allocate_gmres(), cal_b3d(), cal_barycentre(), cal_dif(), cal_histogramme(), cal_x3d(), calmoinsun_x3d(), check_size(), cmp_precond_con(), cmp_precond_dif(), gc__interface::cmp_precond_gra(), communications(), communications_gen(), communications_ray(), conduction_ex(), conduction_step(), def_geom(), diffusion_ex(), diffusion_step(), evol_hydro(), evol_mhd(), explicite(), film_(), gs_mat(), hydro_step(), init(), init_para(), initVarGmres(), inversion_gmres(), limites(), limites_gen(), limites_gra(), limites_ray(), limites_xin(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), mat_prod_gra(), monitoring(), pasdt_con(), pasdt_dif(), pasdt_hydro(), pasdt_mhd(), pasdt_ray(), pasdt_ray_exp(), rd_restart_h5(), restart_new(), sortie_ecran(), update_gravity(), vec_prod(), wrt_main_h5(), and wrt_restart_h5().

7.23.2.8 integer,dimension(:,:),allocatable parameters::nx_cpu

Number of cells in the x,y,z directions per cpu.

Definition at line 21 of file modules.f90.

Referenced by film_(), and init_para().

7.23.2.9 integer,dimension(3) parameters::nx_glob

Global simulation dimensions.

Definition at line 22 of file modules.f90.

Referenced by aff_new(), allocate_array(), check_flags(), cpu_(), default_params(), film_(), init(), init_para(), inversion_gmres(), monitoring(), print_configuration(), rd_restart_h5(), read_params(), restart_new(), wrt_main_h5(), and wrt_restart_h5().

7.23.2.10 integer parameters::nx_glob_max

Global highest nx.

Definition at line 28 of file modules.f90.

Referenced by allocate_array().

7.23.2.11 integer parameters::nx_max

Highest nx.

Definition at line 27 of file modules.f90.

Referenced by allocate_array().

7.23.2.12 integer,dimension(3) parameters::nxmax

Upper limit for arrays.

Definition at line 25 of file modules.f90.

Referenced by allocate_array(), allocate_array_dif(), allocate_array_gra(), allocate_array_ray(), gc__interface::cmp_precond_gra(), cst_gravity(), diffusion_ex(), evol_ray(), explicite(), gs_mat(), limites(), limites_ray(), monitoring(), pasdt_hydro(), and pt_mass().

7.23.2.13 integer,dimension(3) parameters::nxmin

Lower limit for arrays.

Definition at line 24 of file modules.f90.

Referenced by allocate_array(), allocate_array_dif(), allocate_array_gra(), allocate_array_ray(), gc__interface::cmp_precond_gra(), cst_gravity(), diffusion_ex(), evol_ray(), explicite(), gs_mat(), limites(), limites_ray(), monitoring(), pasdt_hydro(), and pt_mass().

7.23.2.14 real,parameter parameters::Pi = 3.1415926535898

π

Definition at line 35 of file modules.f90.

Referenced by gc__interface::cmp_precond_gra().

7.23.2.15 real,parameter parameters::quatre_Pi = 4.*pi

4π

Definition at line 36 of file modules.f90.

7.23.2.16 character(len=10) parameters::riemann

Type of Riemann solver.

Definition at line 39 of file modules.f90.

Referenced by cmpflxm(), default_params(), and read_params().

7.23.2.17 character(len=10) parameters::riemann2d

Type of Riemann solver 2.

Definition at line 40 of file modules.f90.

Referenced by cmp_mag_flx(), default_params(), and read_params().

7.23.2.18 integer parameters::slope_type

Type of slope limiter for Hydro and MHD.

Definition at line 34 of file modules.f90.

Referenced by default_params(), evol_hydro(), print_configuration(), read_params(), sortie_ecran(), and uslope().

7.23.2.19 real,parameter parameters::smallc = 1.d-30

Dimensioned small constant.

Definition at line 37 of file modules.f90.

Referenced by cmp_mag_flx(), ctoprim(), eigen_cons(), eigenvalues(), hydro_acoustic(), trace2d(), and trace3d().

7.23.2.20 real,parameter parameters::smallr = 1.d-40

Dimensioned small real number.

Definition at line 38 of file modules.f90.

Referenced by ctoprim(), hydro_acoustic(), trace1d(), trace2d(), and trace3d().

7.24 prime Module Reference

Contains variables used for prime number computations.

Variables

- integer, parameter Pmax = 50000
- integer, dimension(pmax) PrimeNumber
- integer, dimension(pmax) Number
- integer, parameter Pdmax = 200
- integer Nprime

7.24.1 Detailed Description

Contains variables used for prime number computations.

7.24.2 Variable Documentation

7.24.2.1 integer prime::Nprime

Definition at line 25 of file prime.f90.

Referenced by find_prime().

7.24.2.2 integer,dimension(pmax) prime::Number

Definition at line 23 of file prime.f90.

Referenced by find_prime().

7.24.2.3 integer,parameter prime::Pdmax = 200

Definition at line 24 of file prime.f90.

7.24.2.4 integer,parameter prime::Pmax = 50000

Definition at line 21 of file prime.f90.

Referenced by find_prime(), and PrimeDec().

7.24.2.5 integer,dimension(pmax) prime::PrimeNumber

Definition at line 22 of file prime.f90.

Referenced by find_prime(), and PrimeDec().

7.25 rdwrt_h5 Module Reference

Contains reading and writing hdf5 subroutines.

7.25.1 Detailed Description

Contains reading and writing hdf5 subroutines.

7.26 solvers_hydro Module Reference

Functions/Subroutines

- subroutine solver_relax (dll, ull, pll, cll, ell, fll, drr, urr, prr, crr, err, frr, dss, uss, pss, ess, fss, vit, idim, N)

  Relaxation Riemann solver provided by Benjamin Braconnier (26/06/2008).

- subroutine solver_relaxation (dl, ul, el, pl, cl, dr, ur, er, pr, cr, ds, us, es, ps, ndim, vno)

  Relaxation Riemann solver provided by Benjamin Braconnier (26/06/2008).

- subroutine solver_exact (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Exact Riemann solver, taken from 'Riemann Solvers and Numerical Methods for Fluid Dynamics', E.

- subroutine solver_cc (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Two shock Riemann solver.

- subroutine solver_acoustic (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Acoustic (linearised) Riemann solver.

- subroutine solver_eos (dll, ull, pll, cll, ell, fll, drr, urr, prr, crr, err, frr, dss, uss, pss, ess, fss, vit, idim, N)

  Riemann solver from the FLASH paper.

- subroutine solver_g (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Variable gamma Riemann solver.

- subroutine solver_iso (dll, ull, fll, drr, urr, frr, ds, us, fs, Vit, idim, N)

  Isothermal Riemann solver.

- subroutine starpu (p, u, mpa, dl, ul, pl, cl, dr, ur, pr, cr)

  Computes the pressure and velocity in the star region.

- subroutine guessp (pm, dl, ul, pl, cl, dr, ur, pr, cr)

  Computes an intelligent first guess for the pressure.

- subroutine prefun (f, fd, p, dk, pk, ck)

  Evaluate pressure functions FL and FR.

- subroutine sample (pm, um, s, d, u, p, dl, ul, pl, cl, dr, ur, pr, cr)

  Search through the different waves to find the correct solution.

- subroutine starpu_iso (d, u, mpa, dl, ul, dr, ur, cs, cs2)

  Computes the pressure and velocity in the star region.

- subroutine guessp_iso (dm, dl, ul, dr, ur, cs, cs2)

  Computes an intelligent 'first guess' for the pressure.

- subroutine prefun_iso (f, fd, d, dk, cs)

  Computes pressure functions FL and FR.

- subroutine sample_iso (dm, um, s, d, u, dl, ul, dr, ur, cs, cs2)

  Find solution looking through the different waves.

- subroutine cal_gamma (y)

  Computes the value of the specific heats ratio γ according to the mass fraction y.

7.26.1 Function/Subroutine Documentation

7.26.1.1 subroutine solvers_hydro::cal_gamma (real *y*)

Computes the value of the specific heats ratio γ according to the mass fraction y.

Definition at line 1687 of file solvers.f90.

References gammas::g1, gammas::g2, gammas::g3, gammas::g4, gammas::g5, gammas::g6, gammas::g7, gammas::g8, gammas::gamma, gammas::gamma1, gammas::gamma2, and gammas::mu_reac.

Referenced by solver_g().

Here is the caller graph for this function:

7.26.1.2 subroutine solvers_hydro::guessp (real,intent(out) *pm*, real *dl*, real *ul*, real *pl*, real *cl*, real *dr*, real *ur*, real *pr*, real *cr*)

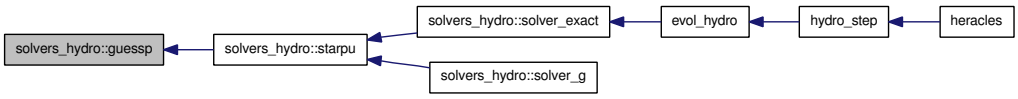
Computes an intelligent first guess for the pressure.

Definition at line 1201 of file solvers.f90.

References gammas::g1, gammas::g3, gammas::g4, gammas::g5, gammas::g6, and gammas::g7.

Referenced by starpu().

Here is the caller graph for this function:



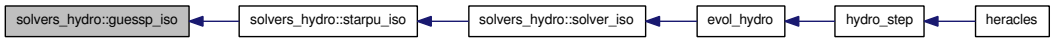
7.26.1.3 subroutine solvers_hydro::guessp_iso (real,intent(out) *dm*, real *dl*, real *ul*, real *dr*, real *ur*, real *cs*, real *cs2*)

Computes an intelligent 'first guess' for the pressure.

Definition at line 1470 of file solvers.f90.

Referenced by starpu_iso().

Here is the caller graph for this function:



7.26.1.4 subroutine solvers_hydro::prefun (real *f*, real *fd*, real *p*, real *dk*, real *pk*, real *ck*)

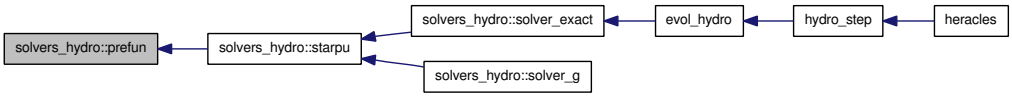
Evaluate pressure functions FL and FR.

Definition at line 1261 of file solvers.f90.

References gammas::g1, gammas::g2, gammas::g4, gammas::g5, and gammas::g6.

Referenced by starpu().

Here is the caller graph for this function:



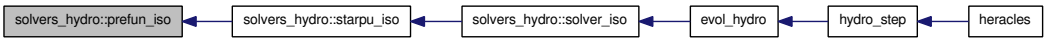
7.26.1.5 subroutine solvers_hydro::prefun_iso (real *f*, real *fd*, real *d*, real *dk*, real *cs*)

Computes pressure functions FL and FR.

Definition at line 1541 of file solvers.f90.

Referenced by starpu_iso().

Here is the caller graph for this function:



7.26.1.6 subroutine solvers_hydro::sample (real,intent(in) *pm*, real,intent(in) *um*, real,intent(in) *s*, real,intent(out) *d*, real,intent(out) *u*, real,intent(out) *p*, real,intent(in) *dl*, real,intent(in) *ul*, real,intent(in) *pl*, real,intent(in) *cl*, real,intent(in) *dr*, real,intent(in) *ur*, real,intent(in) *pr*, real,intent(in) *cr*)

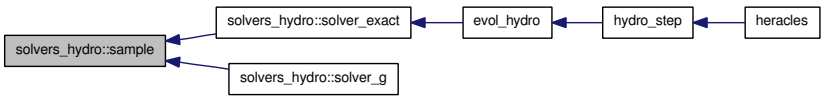
Search through the different waves to find the correct solution.

Definition at line 1297 of file solvers.f90.

References gammas::g1, gammas::g2, gammas::g3, gammas::g4, gammas::g5, gammas::g6, gammas::g7, and gammas::gamma.

Referenced by solver_exact(), and solver_g().

Here is the caller graph for this function:



7.26.1.7 subroutine solvers_hydro::sample_iso (real,intent(in) *dm*, real,intent(in) *um*, real,intent(in) *s*, real,intent(out) *d*, real,intent(out) *u*, real *dl*, real *ul*, real *dr*, real *ur*, real *cs*, real *cs2*)

Find solution looking through the different waves.

Definition at line 1573 of file solvers.f90.

Referenced by solver_iso().

Here is the caller graph for this function:

7.26.1.8 subroutine solvers_hydro::solver_acoustic (real,dimension(1:n),intent(in) *dll*, real,dimension(1:n,1:ndim),intent(in) *ull*, real,dimension(1:n),intent(in) *pll*, real,dimension(1:n,1:nfx),intent(in) *fll*, real,dimension(1:n),intent(in) *drr*, real,dimension(1:n,1:ndim),intent(in) *urr*, real,dimension(1:n),intent(in) *prr*, real,dimension(1:n,1:nfx),intent(in) *frr*, real,dimension(1:n),intent(out) *ds*, real,dimension(1:n,1:ndim),intent(out) *us*, real,dimension(1:n),intent(out) *ps*, real,dimension(1:n,1:nfx),intent(out) *fs*, real,intent(in) *Vit*, integer,intent(in) *idim*, integer *N*)

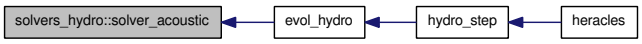
Acoustic (linearised) Riemann solver.

Definition at line 521 of file solvers.f90.

References gammas::gamma, and parameters::ndim.

Referenced by evol_hydro().

Here is the caller graph for this function:



7.26.1.9 subroutine solvers_hydro::solver_cc (real,dimension(1:n),intent(in) *dll*, real,dimension(1:n,1:ndim),intent(in) *ull*, real,dimension(1:n),intent(in) *pll*, real,dimension(1:n,1:nfx),intent(in) *fll*, real,dimension(1:n),intent(in) *drr*, real,dimension(1:n,1:ndim),intent(in) *urr*, real,dimension(1:n),intent(in) *prr*, real,dimension(1:n,1:nfx),intent(in) *frr*, real,dimension(1:n),intent(out) *ds*, real,dimension(1:n,1:ndim),intent(out) *us*, real,dimension(1:n),intent(out) *ps*, real,dimension(1:n,1:nfx),intent(out) *fs*, real,intent(in) *vit*, integer,intent(in) *idim*, integer *N*)

Two shock Riemann solver.

Definition at line 380 of file solvers.f90.

References gammas::g2, gammas::gamma, and parameters::ndim.

7.26.1.10 subroutine solvers_hydro::solver_eos (real,dimension(1:n),intent(in) *dll*, real,dimension(1:n,1:ndim),intent(in) *ull*, real,dimension(1:n),intent(in) *pll*, real,dimension(1:n),intent(in) *cll*, real,dimension(1:n),intent(in) *ell*, real,dimension(1:n,1:nfx),intent(in) *fll*, real,dimension(1:n),intent(in) *drr*, real,dimension(1:n,1:ndim),intent(in) *urr*, real,dimension(1:n),intent(in) *prr*, real,dimension(1:n),intent(in) *crr*, real,dimension(1:n),intent(in) *err*, real,dimension(1:n,1:nfx),intent(in) *frr*, real,dimension(1:n),intent(out) *dss*, real,dimension(1:n,1:ndim),intent(out) *uss*, real,dimension(1:n),intent(out) *pss*, real,dimension(1:n),intent(out) *ess*, real,dimension(1:n,1:nfx),intent(out) *fss*, real,intent(in) *vit*, integer,intent(in) *idim*, integer *N*)

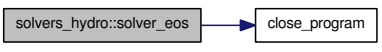
Riemann solver from the FLASH paper.

Definition at line 646 of file solvers.f90.

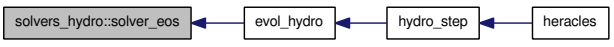
References close_program(), para::mype, parameters::ndim, and divers::verbose.

Referenced by evol_hydro().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.1.11 subroutine solvers_hydro::solver_exact (real,dimension(1:n) *dll*, real,dimension(1:n,1:ndim),intent(in) *ull*, real,dimension(1:n),intent(in) *pll*, real,dimension(1:n,1:nfx),intent(in) *fll*, real,dimension(1:n),intent(in) *drr*, real,dimension(1:n,1:ndim),intent(in) *urr*, real,dimension(1:n),intent(in) *prr*, real,dimension(1:n,1:nfx),intent(in) *frr*, real,dimension(1:n),intent(out) *ds*, real,dimension(1:n,1:ndim),intent(out) *us*, real,dimension(1:n),intent(out) *ps*, real,dimension(1:n,1:nfx),intent(out) *fs*, real,intent(in) *Vit*, integer,intent(in) *idim*, integer *N*)

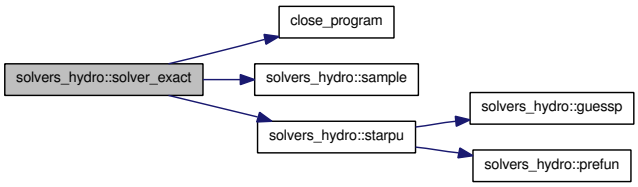
Exact Riemann solver, taken from 'Riemann Solvers and Numerical Methods for Fluid Dynamics', E. F. Toro, 1999.

Definition at line 259 of file solvers.f90.

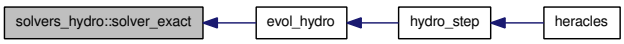
References close_program(), gammas::g4, gammas::gamma, para::mype, parameters::ndim, sample(), and starpu().

Referenced by evol_hydro().

Here is the call graph for this function:



Here is the caller graph for this function:



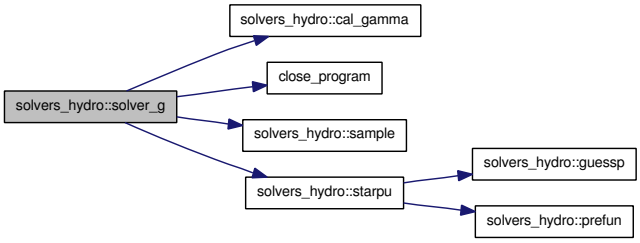
7.26.1.12 subroutine solvers_hydro::solver_g (real,dimension(1:n),intent(in) *dll*, real,dimension(1:n,1:ndim),intent(in) *ull*, real,dimension(1:n),intent(in) *pll*, real,dimension(1:n,1:nfx),intent(in) *fll*, real,dimension(1:n),intent(in) *drr*, real,dimension(1:n,1:ndim),intent(in) *urr*, real,dimension(1:n),intent(in) *prr*, real,dimension(1:n,1:nfx),intent(in) *frr*, real,dimension(1:n),intent(out) *ds*, real,dimension(1:n,1:ndim),intent(out) *us*, real,dimension(1:n),intent(out) *ps*, real,dimension(1:n,1:nfx),intent(out) *fs*, real,intent(in) *Vit*, integer,intent(in) *idim*, integer *N*)

Variable gamma Riemann solver.

Definition at line 968 of file solvers.f90.

References cal_gamma(), close_program(), gammas::g4, gammas::gamma, para::mype, parameters::ndim, sample(), and starpu().

Here is the call graph for this function:



7.26.1.13 subroutine solvers_hydro::solver_iso (real,dimension(1:n),intent(in) *dll*, real,dimension(1:n,1:ndim),intent(in) *ull*, real,dimension(1:n,1:nfx),intent(in) *fll*, real,dimension(1:n),intent(in) *drr*, real,dimension(1:n,1:ndim),intent(in) *urr*, real,dimension(1:n,1:nfx),intent(in) *frr*, real,dimension(1:n),intent(out) *ds*, real,dimension(1:n,1:ndim),intent(out) *us*, real,dimension(1:n,1:nfx),intent(out) *fs*, real *Vit*, integer,intent(in) *idim*, integer *N*)

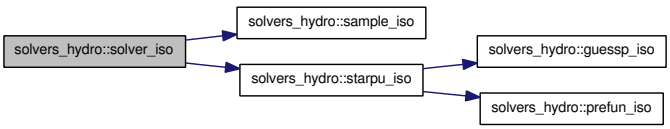
Isothermal Riemann solver.

Definition at line 1065 of file solvers.f90.

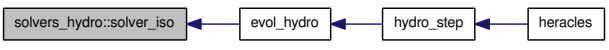
References gammas::Cs_iso, gammas::Cs_iso2, parameters::ndim, sample_iso(), and starpu_iso().

Referenced by evol_hydro().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.1.14 subroutine solvers_hydro::solver_relax (real,dimension(1:n),intent(in) *dll*, real,dimension(1:n,1:ndim),intent(in) *ull*, real,dimension(1:n),intent(in) *pll*, real,dimension(1:n),intent(in),optional *cll*, real,dimension(1:n),intent(in),optional *ell*, real,dimension(1:n,1:nfx),intent(in) *fll*, real,dimension(1:n),intent(in) *drr*, real,dimension(1:n,1:ndim),intent(in) *urr*, real,dimension(1:n),intent(in) *prr*, real,dimension(1:n),intent(in),optional *crr*, real,dimension(1:n),intent(in),optional *err*, real,dimension(1:n,1:nfx),intent(in) *frr*, real,dimension(1:n),intent(out) *dss*, real,dimension(1:n,1:ndim),intent(out) *uss*, real,dimension(1:n),intent(out) *pss*, real,dimension(1:n),intent(out),optional *ess*, real,dimension(1:n,1:nfx),intent(out) *fss*, real,intent(in) *Vit*, integer,intent(in) *idim*, integer *N*)

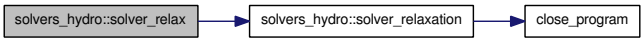
Relaxation Riemann solver provided by Benjamin Braconnier (26/06/2008).

Definition at line 23 of file solvers.f90.

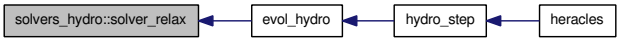
References gammas::gamma, parameters::ndim, solver_relaxation(), and divers::verbose.

Referenced by evol_hydro().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.1.15 subroutine solvers_hydro::solver_relaxation (real,intent(in) *dl*, real,dimension(1:ndim),intent(in) *ul*, real,intent(in) *el*, real,intent(in) *pl*, real,intent(in) *cl*, real,intent(in) *dr*, real,dimension(1:ndim),intent(in) *ur*, real,intent(in) *er*, real,intent(in) *pr*, real,intent(in) *cr*, real,intent(out) *ds*, real,dimension(1:ndim),intent(out) *us*, real,intent(out) *es*, real,intent(out) *ps*, integer,intent(in) *ndim*, real,dimension(1:ndim),intent(inout) *vno*)

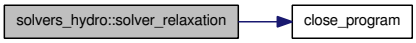
Relaxation Riemann solver provided by Benjamin Braconnier (26/06/2008).

Definition at line 138 of file solvers.f90.

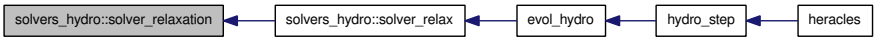
References close_program(), and para::mype.

Referenced by solver_relax().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.1.16 subroutine solvers_hydro::starpu (real *p*, real *u*, real *mpa*, real *dl*, real *ul*, real *pl*, real *cl*, real *dr*, real *ur*, real *pr*, real *cr*)

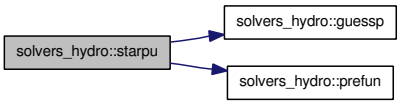
Computes the pressure and velocity in the star region.

Definition at line 1142 of file solvers.f90.

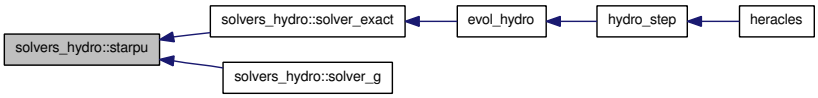
References guessp(), and prefun().

Referenced by solver_exact(), and solver_g().

Here is the call graph for this function:



Here is the caller graph for this function:



7.26.1.17 subroutine solvers_hydro::starpu_iso (real *d*, real *u*, real *mpa*, real *dl*, real *ul*, real *dr*, real *ur*, real *cs*, real *cs2*)

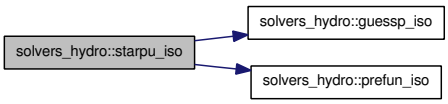
Computes the pressure and velocity in the star region.

Definition at line 1415 of file solvers.f90.

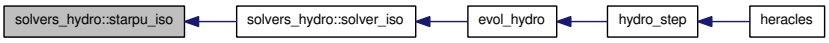
References guessp_iso(), para::mype, and prefun_iso().

Referenced by solver_iso().

Here is the call graph for this function:



Here is the caller graph for this function:



7.27 TabGmres Module Reference

Contains workspace arrays for GMRES.

Data Types

- interface interface

  GMRES interfaces.

Variables

- integer imin

  integer :: imax !<

- integer jmin

  integer :: jmax !<

- integer kmin

  integer :: kmax !<

- real, dimension(:), allocatable work

  real, dimension(:), allocatable :: workred !<

- real, dimension(:), allocatable xin

  real, dimension(:), allocatable :: xout !<

- real, dimension(:), allocatable b

  end module TabGmres

- integer, dimension(7) icntl

  Contains the parameters for GMRES.

- integer itermax

  integer :: lwork !<

- real, dimension(5) cntl

  end module VarGmres

7.27.1 Detailed Description

Contains workspace arrays for GMRES.

7.27.2 Variable Documentation

7.27.2.1 real,dimension(:),allocatable TabGmres::b

end module TabGmres

Definition at line 125 of file modules_ray.f90.

7.27.2.2 real,dimension(5) TabGmres::cntl

end module VarGmres

Definition at line 143 of file modules_ray.f90.

7.27.2.3 integer,dimension(7) TabGmres::icntl

Contains the parameters for GMRES. integer :: restart !<

Definition at line 139 of file modules_ray.f90.

7.27.2.4 integer TabGmres::imin

integer :: imax !<

Definition at line 115 of file modules_ray.f90.

7.27.2.5 integer TabGmres::itermax

integer :: lwork !<

Definition at line 141 of file modules_ray.f90.

7.27.2.6 integer TabGmres::jmin

integer :: jmax !<

Definition at line 117 of file modules_ray.f90.

7.27.2.7 integer TabGmres::kmin

integer :: kmax !<

Definition at line 119 of file modules_ray.f90.

7.27.2.8 real,dimension(:),allocatable TabGmres::work

real, dimension(:), allocatable :: workred !<

Definition at line 121 of file modules_ray.f90.

7.27.2.9 real,dimension(:),allocatable TabGmres::xin

real, dimension(:), allocatable :: xout !<

Definition at line 123 of file modules_ray.f90.

7.28 unites Module Reference

This routine calls the relevant output routines to write binary or HDF5 files.

Variables

- real gramme
  Gram unit.
- real centimetre
  Centimetre unit.
- real seconde
  Second unit.
- real dyne
  Dyne unit.
- real erg
  Erg unit.
- real degre
  Temperature unit.
- real unitm
  Mass unit.
- real unitl
  Length unit.
- real unitt
  Time unit.
- real unitf
  Force unit.
- real unite
  Energy unit.
- real unitd
  Temperature unit.
- real unitmom
  Momentum unit.
- real cm2
  cm^2
- real cm3

cm^3

- real ms

  Millisecond.

- real kms

  Kilometer.

- real an

  Year.

- real Mans

  Million years.

- real Gans

  Billion years.

- real Msol

  Solar mass $M_\odot$.

- real Rsol

  Solar radius $R_\odot$.

- real Lsol

  Solar luminosity $L_\odot$.

- real pc

  Parsec.

- real kpc

  Kiloparsec.

- real Mpc

  Megaparsec.

- real Kelvin

  Kelvin.

- real c

  Speed of light c.

- real c2

  Square of the speed of light c^2.

- real eV

  Electron volt eV.

- real G

  Gravitational constant G.

- real hplanck

  Planck constant h.

- real kB

  Boltzmann constant k_B.

- real uma

  Atomic mass unit u.

- real H0

  Hubble's constant H_0.

- real a_R

  Stefan-Boltzmann constant a_R.

- real Pascal

  Pascal.

- real bar

  Bar.

- real kg

  Kilogram.

- real metre

  Metre.

- real m2

  m^2

- real m3

  m^3

- real joule

  Joule.

- real ns

  Nanosecond.

- real micron

  Micron.

- real Gauss

  Gauss.

- real mu0

  Reduced atomic mass.

7.28.1 Detailed Description

This routine calls the relevant output routines to write binary or HDF5 files.

7.28.2 Variable Documentation

7.28.2.1 real unites::a_R

Stefan-Boltzmann constant a_R.

Definition at line 208 of file modules.f90.

Referenced by dimensions().

7.28.2.2 real unites::an

Year.

Definition at line 190 of file modules.f90.

Referenced by aff_new(), cal_histogramme(), conduction_imp_gc__interface::cmp_precond_con(), dimensions(), heracles(), init_film(), pasdt(), pasdt_con(), pasdt_dif(), pasdt_hydro(), and pasdt_mhd().

7.28.2.3 real unites::bar

Bar.

Definition at line 210 of file modules.f90.

7.28.2.4 real unites::c

Speed of light c.

Definition at line 200 of file modules.f90.

Referenced by cal_b3d(), cal_Dedd(), cal_x3d(), calmoinsun_x3d(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), diffusion_ex(), dimensions(), exp_source_ray(), explicite(), gs_mat(), limites_xin(), mat_prod3d(), mat_prod_dif(), pasdt_dif(), pasdt_ray(), and pasdt_ray_exp().

7.28.2.5 real unites::c2

Square of the speed of light c^2.

Definition at line 201 of file modules.f90.

Referenced by cal_b3d(), dimensions(), explicite(), gs_mat(), and mat_prod3d().

7.28.2.6 real unites::centimetre

Centimetre unit.

Definition at line 174 of file modules.f90.

Referenced by dimensions(), init(), init_gravity(), print_configuration(), set_units_out(), and sortie_ecran().

7.28.2.7 real unites::cm2

cm^2

Definition at line 186 of file modules.f90.

Referenced by dimensions().

7.28.2.8 real unites::cm3

cm^3

Definition at line 187 of file modules.f90.

Referenced by cal_histogramme(), dimensions(), film_(), rh(), and sortie_ecran().

7.28.2.9 real unites::degre

Temperature unit.

Definition at line 178 of file modules.f90.

Referenced by dimensions(), print_configuration(), and rh().

7.28.2.10 real unites::dyne

Dyne unit.

Definition at line 176 of file modules.f90.

Referenced by dimensions().

7.28.2.11 real unites::erg

Erg unit.

Definition at line 177 of file modules.f90.

Referenced by cal_histogramme(), dimensions(), and sortie_ecran().

7.28.2.12 real unites::eV

Electron volt eV.

Definition at line 202 of file modules.f90.

Referenced by dimensions().

7.28.2.13 real unites::G

Gravitational constant G.

Definition at line 203 of file modules.f90.

Referenced by gc__interface::cmp_precond_gra(), dimensions(), limites_gra(), pasdt_grav(), and pt_-mass().

7.28.2.14 real unites::Gans

Billion years.

Definition at line 192 of file modules.f90.

Referenced by dimensions().

7.28.2.15 real unites::Gauss

Gauss.

Definition at line 218 of file modules.f90.

7.28.2.16 real unites::gramme

Gram unit.

Definition at line 173 of file modules.f90.

Referenced by cal_histogramme(), dimensions(), film_(), init_gravity(), print_configuration(), rh(), set_units_out(), and sortie_ecran().

7.28.2.17 real unites::H0

Hubble's constant H_0.

Definition at line 207 of file modules.f90.

Referenced by dimensions().

7.28.2.18 real unites::hplanck

Planck constant h.

Definition at line 204 of file modules.f90.

Referenced by dimensions().

7.28.2.19 real unites::joule

Joule.

Definition at line 215 of file modules.f90.

Referenced by dimensions().

7.28.2.20 real unites::kB

Boltzmann constant k_B.

Definition at line 205 of file modules.f90.

7.28.2.21 real unites::Kelvin

Kelvin.

Definition at line 199 of file modules.f90.

Referenced by dimensions().

7.28.2.22 real unites::kg

Kilogram.

Definition at line 211 of file modules.f90.

Referenced by dimensions().

7.28.2.23 real unites::kms

Kilometer.

Definition at line 189 of file modules.f90.

Referenced by dimensions(), and rh().

7.28.2.24 real unites::kpc

Kiloparsec.

Definition at line 197 of file modules.f90.

Referenced by dimensions().

7.28.2.25 real unites::Lsol

Solar luminosity $L_{\odot}$.

Definition at line 195 of file modules.f90.

Referenced by dimensions().

7.28.2.26 real unites::m2

m^2

Definition at line 213 of file modules.f90.

7.28.2.27 real unites::m3

m^3

Definition at line 214 of file modules.f90.

7.28.2.28 real unites::Mans

Million years.

Definition at line 191 of file modules.f90.

Referenced by dimensions(), and restart_new().

7.28.2.29 real unites::metre

Metre.

Definition at line 212 of file modules.f90.

Referenced by cal_kappa(), cal_kappa_dif(), cal_sigmadiff(), and dimensions().

7.28.2.30 real unites::micron

Micron.

Definition at line 217 of file modules.f90.

7.28.2.31 real unites::Mpc

Megaparsec.

Definition at line 198 of file modules.f90.

Referenced by dimensions().

7.28.2.32 real unites::ms

Millisecond.

Definition at line 188 of file modules.f90.

Referenced by dimensions().

7.28.2.33 real unites::Msol

Solar mass $M_{\odot}$.

Definition at line 193 of file modules.f90.

Referenced by dimensions().

7.28.2.34 real unites::mu0

Reduced atomic mass.

Definition at line 219 of file modules.f90.

7.28.2.35 real unites::ns

Nanosecond.

Definition at line 216 of file modules.f90.

7.28.2.36 real unites::Pascal

Pascal.

Definition at line 209 of file modules.f90.

Referenced by dimensions(), and rh().

7.28.2.37 real unites::pc

Parsec.

Definition at line 196 of file modules.f90.

Referenced by dimensions().

7.28.2.38 real unites::Rsol

Solar radius $R_\odot$.

Definition at line 194 of file modules.f90.

Referenced by dimensions().

7.28.2.39 real unites::seconde

Second unit.

Definition at line 175 of file modules.f90.

Referenced by dimensions(), explicite(), gs_mat(), heracles(), init_gravity(), pasdt(), pasdt_ray_exp(), print_configuration(), set_units_out(), and sortie_ecran().

7.28.2.40 real unites::uma

Atomic mass unit u.

Definition at line 206 of file modules.f90.

Referenced by dimensions(), Energy(), heat_capacity(), rh(), and Temperature().

7.28.2.41 real unites::unitd

Temperature unit.

Definition at line 184 of file modules.f90.

7.28.2.42 real unites::unite

Energy unit.

Definition at line 183 of file modules.f90.

7.28.2.43 real unites::unitf

Force unit.

Definition at line 182 of file modules.f90.

7.28.2.44 real unites::unitl

Length unit.

Definition at line 180 of file modules.f90.

7.28.2.45 real unites::unitm

Mass unit.

Definition at line 179 of file modules.f90.

7.28.2.46 real unites::unitmom

Momentum unit.

Definition at line 185 of file modules.f90.

7.28.2.47 real unites::unitt

Time unit.

Definition at line 181 of file modules.f90.

7.29 unites_sortie Module Reference

Contains the units for output file dumping.

Variables

- real u_M
  Output mass unit.
- real u_L
  Output length unit.
- real u_T
  Output time unit.
- real u_E
  Output energy unit.
- real u_Vol
  Output volume unit.
- real u_dens
  Output density unit.
- real u_evol
  Output volumic energy unit.
- real u_Fr
  Output force unit.
- real u_Vit
  Output velocity unit.
- real u_Mom
  Output momentum unit.

7.29.1 Detailed Description

Contains the units for output file dumping.

7.29.2 Variable Documentation

7.29.2.1 real unites_sortie::u_dens

Output density unit.

Definition at line 238 of file modules.f90.

Referenced by aff_new(), rd_restart_h5(), restart_new(), set_units_out(), and wrt_restart_h5().

7.29.2.2 real unites_sortie::u_E

Output energy unit.

Definition at line 236 of file modules.f90.

Referenced by set_units_out().

7.29.2.3 real unites_sortie::u_evol

Output volumic energy unit.

Definition at line 239 of file modules.f90.

7.29.2.4 real unites_sortie::u_Fr

Output force unit.

Definition at line 240 of file modules.f90.

Referenced by aff_new(), rd_restart_h5(), restart_new(), set_units_out(), and wrt_restart_h5().

7.29.2.5 real unites_sortie::u_L

Output length unit.

Definition at line 234 of file modules.f90.

Referenced by aff_new(), rd_restart_h5(), restart_new(), set_units_out(), wrt_main_h5(), and wrt_restart_-h5().

7.29.2.6 real unites_sortie::u_M

Output mass unit.

Definition at line 233 of file modules.f90.

Referenced by set_units_out().

7.29.2.7 real unites_sortie::u_Mom

Output momentum unit.

Definition at line 242 of file modules.f90.

Referenced by aff_new(), rd_restart_h5(), restart_new(), set_units_out(), and wrt_restart_h5().

7.29.2.8 real unites_sortie::u_T

Output time unit.

Definition at line 235 of file modules.f90.

Referenced by aff_new(), rd_restart_h5(), restart_new(), set_units_out(), wrt_main_h5(), and wrt_restart_-h5().

7.29.2.9 real unites_sortie::u_Vit

Output velocity unit.

Definition at line 241 of file modules.f90.

Referenced by set_units_out().

7.29.2.10 real unites_sortie::u_Vol

Output volume unit.

Definition at line 237 of file modules.f90.

Referenced by set_units_out().

7.30 valeurs_propres Module Reference

Contains the array to hold the tabulated M1 eignevalues.

Variables

- integer n_points

  Number of points in the tabulated eigenvalues curve.

- real, dimension(:,:,:,:), allocatable valp

  Array to hold the tabulated eigenvalues as a function of θ and ϵ.

7.30.1 Detailed Description

Contains the array to hold the tabulated M1 eignevalues.

7.30.2 Variable Documentation

7.30.2.1 integer valeurs_propres::n_points

Number of points in the tabulated eigenvalues curve.

Definition at line 100 of file modules_ray.f90.

Referenced by interpol_valp(), and read_valp().

7.30.2.2 real,dimension(:,:,:,:),allocatable valeurs_propres::valp

Array to hold the tabulated eigenvalues as a function of θ and ϵ.

Definition at line 101 of file modules_ray.f90.

Referenced by interpol_valp(), and read_valp().

7.31 var Module Reference

Contains all the main variable arrays.

Variables

- real, dimension(:,:,:,:), allocatable rhou

  Gas momentum array $\rho(u_x, u_y, u_z)$.

- real, dimension(:,:,:,:), allocatable B

  Magnetic field.

- real, dimension(:,:,:,:), allocatable Fx

  Passive scalar array.

- real, dimension(:,:,:,:), allocatable supp1

  Additional array 1.

- real, dimension(:,:,:,:), allocatable supp2

  Additional array 2.

- real, dimension(:,:,:,:), allocatable supp3

  Additional array 3.

- real, dimension(:,:,:,:), allocatable supp4

  Additional array 4.

- real, dimension(:,:,:,:), allocatable supp5

  Additional array 5.

- real, dimension(:,:,:), allocatable rho

  Gas density array ρ.

- real, dimension(:,:,:), allocatable E

  Total gas energy $E = \frac{1}{2}\rho u^2 + \frac{P}{\gamma - 1}$.

- real, dimension(:,:,:), allocatable Tgaz

  Gas temperature array.

7.31.1 Detailed Description

Contains all the main variable arrays.

7.31.2 Variable Documentation

7.31.2.1 real,dimension(:,:,:,:),allocatable var::B

Magnetic field.

Definition at line 108 of file modules.f90.

Referenced by aff_new(), allocate_array(), cmp_precond_con(), conduction_imp_gc__interface::cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), communications(), conduction_ex(), diffusion_ex(), evol_mhd(), evol_ray(), heracles(), limites(), Mat_prod_con(), mat_prod_dif(), monitoring(), pasdt_con(), pasdt_dif(), pasdt_hydro(), pasdt_mhd(), rd_restart_h5(), restart_new(), and wrt_restart_h5().

7.31.2.2 real,dimension(:,:,:),allocatable var::E

Total gas energy $E = \frac{1}{2}\rho u^2 + \frac{P}{\gamma - 1}$.

Definition at line 116 of file modules.f90.

Referenced by aff_new(), allocate_array(), cal_histogramme(), cmp_precond_con(), conduction_imp_gc__interface::cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), communications(), conduction_ex(), conduction_step(), diffusion_ex(), evol_hydro(), evol_mhd(), evol_ray(), heracles(), init(), limites(), Mat_prod_con(), mat_prod_dif(), monitoring(), pasdt_con(), pasdt_dif(), pasdt_hydro(), pasdt_mhd(), rd_restart_h5(), restart_new(), sortie_ecran(), update_gravity(), and wrt_restart_h5().

7.31.2.3 real,dimension(:,:,:,:),allocatable var::Fx

Passive scalar array.

Definition at line 109 of file modules.f90.

Referenced by allocate_array().

7.31.2.4 real,dimension(:,:,:),allocatable var::rho

Gas density array ρ.

Definition at line 115 of file modules.f90.

Referenced by aff_new(), allocate_array(), cal_b3d(), cal_barycentre(), cal_histogramme(), cmp_precond_con(), conduction_imp_gc__interface::cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), gc__interface::cmp_precond_gra(), communications(), conduction_step(), evol_hydro(), evol_mhd(), evol_ray(), explicite(), fill_new_arrays(), gs_mat(), heracles(), init(), limites(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), monitoring(), pasdt_con(), pasdt_dif(), pasdt_grav(), pasdt_hydro(), pasdt_mhd(), rd_restart_h5(), restart_new(), sortie_ecran(), update_gravity(), and wrt_restart_h5().

7.31.2.5 real,dimension(:,:,:,:),allocatable var::rhou

Gas momentum array $\rho(u_x, u_y, u_z)$.

Definition at line 107 of file modules.f90.

Referenced by aff_new(), allocate_array(), cal_histogramme(), cmp_precond_con(), conduction_imp_gc_-_interface::cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), communications(), conduction_step(), evol_hydro(), evol_mhd(), evol_ray(), heracles(), init(), limites(), Mat_-prod_con(), mat_prod_dif(), monitoring(), pasdt_con(), pasdt_dif(), pasdt_hydro(), pasdt_mhd(), rd_-restart_h5(), restart_new(), sortie_ecran(), update_gravity(), and wrt_restart_h5().

7.31.2.6 real,dimension(:,:,:,:),allocatable var::supp1

Additional array 1.

Definition at line 110 of file modules.f90.

Referenced by allocate_array(), and fill_new_arrays().

7.31.2.7 real,dimension(:,:,:,:),allocatable var::supp2

Additional array 2.

Definition at line 111 of file modules.f90.

Referenced by allocate_array(), and fill_new_arrays().

7.31.2.8 real,dimension(:,:,:,:),allocatable var::supp3

Additional array 3.

Definition at line 112 of file modules.f90.

Referenced by allocate_array().

7.31.2.9 real,dimension(:,:,:,:),allocatable var::supp4

Additional array 4.

Definition at line 113 of file modules.f90.

Referenced by allocate_array().

7.31.2.10 real,dimension(:,:,:,:),allocatable var::supp5

Additional array 5.

Definition at line 114 of file modules.f90.

Referenced by allocate_array().

7.31.2.11 real,dimension(:,:,:),allocatable var::Tgaz

Gas temperature array.

Definition at line 117 of file modules.f90.

Referenced by cal_x3d(), calmoinsun_x3d(), communications_ray(), evol_ray(), explicite(), gs_mat(), heracles(), limites_ray(), and pasdt_ray().

7.32 var_loc Module Reference

Contains the local variables.

Variables

- real, dimension(:,:,:,:), allocatable rhouloc
  Local momentum array.
- real, dimension(:,:,:,:), allocatable uloc
  Local velocity array.
- real, dimension(:,:,:,:), allocatable Fxloc
  Local passive scalar array.
- real, dimension(:,:,:), allocatable rholoc
  Local density array.
- real, dimension(:,:,:), allocatable Ploc
  Local pressure array.
- real, dimension(:,:,:), allocatable Eloc
  Local total gas energy array.
- real, dimension(:,:,:), allocatable Tloc
  Local temperature array.
- real, dimension(:,:,:), allocatable Einloc
  Local internal gas energy array.
- real, dimension(:,:,:), allocatable philoc
  Local gravitational potential array.
- real, dimension(:,:), allocatable xloc
  Local coordinate array.

7.32.1 Detailed Description

Contains the local variables.

7.32.2 Variable Documentation

7.32.2.1 real,dimension(:,:,:),allocatable var_loc::Einloc

Local internal gas energy array.

Definition at line 61 of file modules.f90.

Referenced by evol_hydro(), and evol_mhd().

7.32.2.2 real,dimension(:,:,:),allocatable var_loc::Eloc

Local total gas energy array.

Definition at line 59 of file modules.f90.

Referenced by conduction_ex(), diffusion_ex(), and evol_hydro().

7.32.2.3 real,dimension(:,:,:,:),allocatable var_loc::Fxloc

Local passive scalar array.

Definition at line 56 of file modules.f90.

Referenced by evol_hydro().

7.32.2.4 real,dimension(:,:,:),allocatable var_loc::philoc

Local gravitational potential array.

Definition at line 62 of file modules.f90.

Referenced by evol_hydro(), and evol_mhd().

7.32.2.5 real,dimension(:,:,:),allocatable var_loc::Ploc

Local pressure array.

Definition at line 58 of file modules.f90.

Referenced by evol_hydro().

7.32.2.6 real,dimension(:,:,:),allocatable var_loc::rholoc

Local density array.

Definition at line 57 of file modules.f90.

Referenced by conduction_ex(), diffusion_ex(), and evol_hydro().

7.32.2.7 real,dimension(:,:,:,:),allocatable var_loc::rhouloc

Local momentum array.

Definition at line 54 of file modules.f90.

Referenced by conduction_ex(), diffusion_ex(), and evol_hydro().

7.32.2.8 real,dimension(:,:,:),allocatable var_loc::Tloc

Local temperature array.

Definition at line 60 of file modules.f90.

Referenced by cmp_precond_con(), cmp_precond_dif(), conduction_ex(), diffusion_ex(), Mat_prod_con(), and mat_prod_dif().

7.32.2.9 real,dimension(:,:,:,:),allocatable var_loc::uloc

Local velocity array.

Definition at line 55 of file modules.f90.

Referenced by conduction_ex(), and evol_hydro().

7.32.2.10 real,dimension(:,:),allocatable var_loc::xloc

Local coordinate array.

Definition at line 63 of file modules.f90.

Referenced by cal_b3d(), cal_geom(), cmp_precond_con(), cmp_precond_dif(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), exp_comobile_ray(), exp_source_ray(), explicite(), gs_mat(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), trace2d(), trace3d(), and update().

7.33 varold Module Reference

Contains all the old (non-updated) variable arrays.

Variables

- real, dimension(:,:,:,:), allocatable rhouold

  Old gas momentum.

- real, dimension(:,:,:,:), allocatable Bold

  Old magnetic field.

- real, dimension(:,:,:,:), allocatable Fxold

  Old passive scalar.

- real, dimension(:,:,:), allocatable rhoold

  Old gas density.

- real, dimension(:,:,:), allocatable Eold

  Old total gas energy.

- real, dimension(:,:,:), allocatable Tgazold

  Old gas temperature.

7.33.1 Detailed Description

Contains all the old (non-updated) variable arrays.

7.33.2 Variable Documentation

7.33.2.1 real,dimension(:,:,:,:),allocatable varold::Bold

Old magnetic field.

Definition at line 134 of file modules.f90.

Referenced by allocate_array(), conduction_ex(), evol_mhd(), and heracles().

7.33.2.2 real,dimension(:,:,:),allocatable varold::Eold

Old total gas energy.

Definition at line 137 of file modules.f90.

Referenced by allocate_array(), conduction_ex(), conduction_step(), diffusion_ex(), evol_hydro(), evol_mhd(), and heracles().

7.33.2.3 real,dimension(:,:,:,:),allocatable varold::Fxold

Old passive scalar.

Definition at line 135 of file modules.f90.

Referenced by allocate_array(), and evol_hydro().

7.33.2.4 real,dimension(:,:,:),allocatable varold::rhoold

Old gas density.

Definition at line 136 of file modules.f90.

Referenced by allocate_array(), conduction_ex(), conduction_step(), diffusion_ex(), evol_hydro(), evol_mhd(), heracles(), and update_gravity().

7.33.2.5 real,dimension(:,:,:,:),allocatable varold::rhouold

Old gas momentum.

Definition at line 133 of file modules.f90.

Referenced by allocate_array(), conduction_ex(), conduction_step(), diffusion_ex(), evol_hydro(), evol_mhd(), heracles(), and update_gravity().

7.33.2.6 real,dimension(:,:,:),allocatable varold::Tgazold

Old gas temperature.

Definition at line 138 of file modules.f90.

Referenced by allocate_array_ray(), cal_x3d(), calmoinsun_x3d(), evol_ray(), explicite(), gs_mat(), heracles(), and pasdt_ray().

7.34 varray Module Reference

Contains the M1 radiative transfer variables.

Variables

- real, dimension(:,:,:), allocatable Eray

  Radiative energy.

- real, dimension(:,:,:), allocatable Erayold

  Old radiative energy.

- real, dimension(:,:,:), allocatable Eray_t

  Radiative energy at time $t + 0.5dt$.

- real, dimension(:,:,:,:), allocatable Fray

  Radiative flux.

- real, dimension(:,:,:,:), allocatable Frayold

  Old radiative flux.

- real, dimension(:,:,:,:), allocatable Fray_t

  Radiative flux at time $t + 0.5dt$.

- real, dimension(:,:,:), allocatable ff_t

  real, dimension(:,:,:), allocatable :: kappa_em_t !< Emission opacity $\kappa_{\rm em}$

- real, dimension(:,:,:), allocatable kappa_abs_t

  Absorption opacity $\kappa_{\rm abs}$.

- real, dimension(:,:,:), allocatable sigma_diff

  Scattering coefficient σ_s.

7.34.1 Detailed Description

Contains the M1 radiative transfer variables.

7.34.2 Variable Documentation

7.34.2.1 real,dimension(:,:,:),allocatable varray::Eray

Radiative energy.

Definition at line 17 of file modules_ray.f90.

Referenced by aff_new(), allocate_array_ray(), cal_x3d(), calmoinsun_x3d(), communications_ray(), evol_ray(), exp_comobile_ray(), explicite(), gs_mat(), heracles(), limites_ray(), pasdt_ray(), and restart_new().

7.34.2.2 real,dimension(:,:,:),allocatable varray::Eray_t

Radiative energy at time $t + 0.5dt$.

Definition at line 19 of file modules_ray.f90.

Referenced by allocate_array_ray().

7.34.2.3 real,dimension(:,:,:),allocatable varray::Erayold

Old radiative energy.

Definition at line 18 of file modules_ray.f90.

Referenced by allocate_array_ray(), cal_x3d(), calmoinsun_x3d(), evol_ray(), exp_comobile_ray(), explicite(), gs_mat(), heracles(), and pasdt_ray().

7.34.2.4 real,dimension(:,:,:),allocatable varray::ff_t

real, dimension(:,:,:), allocatable :: kappa_em_t !< Emission opacity $\kappa_{\rm em}$

Definition at line 23 of file modules_ray.f90.

Referenced by allocate_array_ray().

7.34.2.5 real,dimension(:,:,:,:),allocatable varray::Fray

Radiative flux.

Definition at line 20 of file modules_ray.f90.

Referenced by aff_new(), allocate_array_ray(), cal_x3d(), calmoinsun_x3d(), communications_ray(), evol_ray(), exp_comobile_ray(), explicite(), gs_mat(), heracles(), limites_ray(), pasdt_ray(), and restart_new().

7.34.2.6 real,dimension(:,:,:,:),allocatable varray::Fray_t

Radiative flux at time $t + 0.5dt$.

Definition at line 22 of file modules_ray.f90.

Referenced by allocate_array_ray().

7.34.2.7 real,dimension(:,:,:,:),allocatable varray::Frayold

Old radiative flux.

Definition at line 21 of file modules_ray.f90.

Referenced by allocate_array_ray(), cal_x3d(), calmoinsun_x3d(), evol_ray(), exp_comobile_ray(), exp_source_ray(), explicite(), gs_mat(), heracles(), and pasdt_ray().

7.34.2.8 real,dimension(:,:,:),allocatable varray::kappa_abs_t

Absorption opacity $\kappa_{\rm abs}$.

Definition at line 25 of file modules_ray.f90.

Referenced by allocate_array_ray(), cal_b3d(), explicite(), gs_mat(), and mat_prod3d().

7.34.2.9 real,dimension(:,:,:),allocatable varray::sigma_diff

Scattering coefficient σ_s.

Definition at line 26 of file modules_ray.f90.

Referenced by allocate_array_ray(), cal_b3d(), explicite(), gs_mat(), and mat_prod3d().

Chapter 8

Data Type Documentation

8.1 cg__interface Interface Reference

Public Member Functions

- subroutine mat_prod (X, AX)
- subroutine cmp_precond (R, Z)

8.1.1 Detailed Description

Definition at line 30 of file cg.f90.

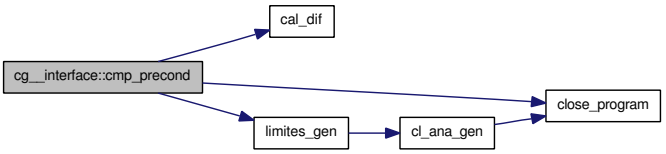
8.1.2 Member Function/Subroutine Documentation

8.1.2.1 subroutine cg__interface::cmp_precond (real,dimension(nxmin(1) *R*, real,dimension(nxmin(1) *Z*)

Definition at line 36 of file cg.f90.

References cal_dif(), close_program(), limites_gen(), para::mype, and parameters::ndim.

Here is the call graph for this function:



8.1.2.2 subroutine cg__interface::mat_prod (real,dimension(nxmin(1) *X*, real,dimension(nxmin(1) *AX*)

Definition at line 31 of file cg.f90.

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

- algebra/cg.f90

8.2 conduction_imp_gc__interface Interface Reference

Public Member Functions

- subroutine mat_prod_con (X, AX)
- subroutine cmp_precond_con (R, Z)

8.2.1 Detailed Description

Definition at line 38 of file conduction_imp_gc.f90.

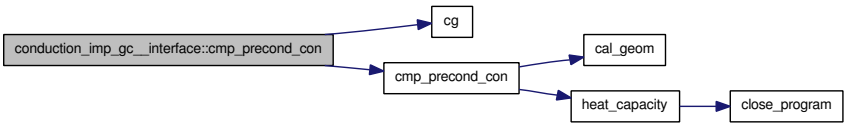
8.2.2 Member Function/Subroutine Documentation

8.2.2.1 subroutine conduction_imp_gc__interface::cmp_precond_con (real,dimension(nxmin(1) *R*, real,dimension(nxmin(1) *Z*)

Definition at line 44 of file conduction_imp_gc.f90.

References unites::an, var::B, cg(), cmp_precond_con(), divers::dt, conduction::dt_con_imp, var::E, cl_con::icl_con, para::mype, conduction::nitetot_con, conduction::precond_con, var::rho, var::rhou, conduction::vardt_con, conduction::varrel_con, divers::verbose, and geom::x.

Here is the call graph for this function:



8.2.2.2 subroutine conduction_imp_gc__interface::mat_prod_con (real,dimension(nxmin(1) *X*, real,dimension(nxmin(1) *AX*)

Definition at line 39 of file conduction_imp_gc.f90.

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

- conduction/conduction_imp_gc.f90

8.3 diffusion_imp__interface Interface Reference

Public Member Functions

- subroutine mat_prod_dif (X, AX)
- subroutine cmp_precond_dif (R, Z)

8.3.1 Detailed Description

Definition at line 41 of file diffusion_imp_gc.f90.

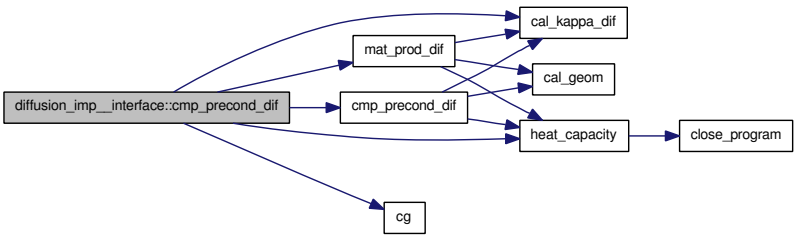
8.3.2 Member Function/Subroutine Documentation

8.3.2.1 subroutine diffusion_imp__interface::cmp_precond_dif (real,dimension(nxmin(1) *R*, real,dimension(nxmin(1) *Z*)

Definition at line 47 of file diffusion_imp_gc.f90.

References var::B, unites::c, cal_kappa_dif(), cg(), cmp_precond_dif(), divers::dt, diffusion::dt_dif_imp, var::E, diffusion::Er_dif, heat_capacity(), cl_dif::icl_dif, mat_prod_dif(), para::mype, diffusion::nitetot_dif, diffusion::precond_dif, var::rho, var::rhou, diffusion::vardt_dif, diffusion::varrel_dif, divers::verbose, and geom::x.

Here is the call graph for this function:



8.3.2.2 subroutine diffusion_imp__interface::mat_prod_dif (real,dimension(nxmin(1) *X*, real,dimension(nxmin(1) *AX*)

Definition at line 42 of file diffusion_imp_gc.f90.

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

- rad_transfer/fld/diffusion_imp_gc.f90

8.4 gc__interface Interface Reference

Public Member Functions

- subroutine mat_prod_gra (X, AX)
- subroutine cmp_precond_gra (R, Z)

8.4.1 Detailed Description

Definition at line 45 of file gc.f90.

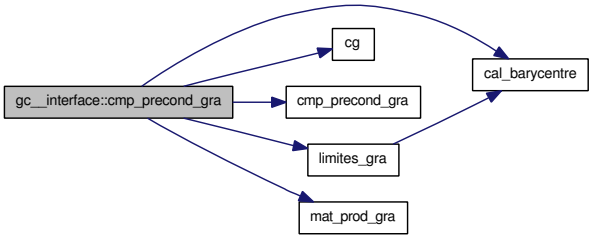
8.4.2 Member Function/Subroutine Documentation

8.4.2.1 subroutine gc__interface::cmp_precond_gra (real,dimension(nxmin(1) *R*, real,dimension(nxmin(1) *Z*)

Definition at line 51 of file gc.f90.

References cal_barycentre(), cg(), cmp_precond_gra(), unites::G, cl::icl, gravity::ISOLE, geom::Lbox, limites_gra(), mat_prod_gra(), parameters::ndim, parameters::nx, parameters::nxmax, parameters::nxmin, gravity::phi, parameters::Pi, var::rho, and divers::verbose.

Here is the call graph for this function:



8.4.2.2 subroutine gc__interface::mat_prod_gra (real,dimension(nxmin(1) *X*, real,dimension(nxmin(1) *AX*)

Definition at line 46 of file gc.f90.

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

- gravity/gc.f90

8.5 TabGmres::interface Interface Reference

GMRES interfaces.

Public Member Functions

- subroutine gmres (n, m, nloc, work, nx, ny, nz, nvar_ray, imin, imax, jmin, jmax, kmin, kmax, workred, lwork, cntl, icntl, info)

8.5.1 Detailed Description

GMRES interfaces.

Definition at line 157 of file modules_ray.f90.

8.5.2 Member Function/Subroutine Documentation

8.5.2.1 subroutine TabGmres::interface::gmres (integer,intent(in) *n*, m, integer,intent(in) *nloc*, real,dimension(lwork) *work*, integer,intent(in) *nx*, ny, integer,intent(in) *nz*, real,dimension(-nvar_ray+1:(imax-imin+(jmax-jmin) *nvar_ray*, integer,intent(in) *imin*, imax, integer,intent(in) *jmin*, jmax, integer,intent(in) *kmin*, kmax, real,dimension(-nvar_ray+1:(imax-imin+(jmax-jmin) *workred*, lwork, cntl, integer,dimension(jmax-jmin+1) *icntl*, info)

Parameters:

n integer, intent(in) :: m !<

nloc integer, intent(in) :: lwork !<

work real , dimension(5) :: cntl !<

nx integer, intent(in) :: ny !<

nz integer, intent(in) :: nvar_ray !<

imin integer, intent(in) :: imax !<

jmin integer, intent(in) :: jmax !<

kmin integer, intent(in) :: kmax !<

icntl integer, dimension(3) :: info !<

Definition at line 158 of file modules_ray.f90.

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

- rad_transfer/m1/modules_ray.f90

Chapter 9

File Documentation

9.1 algebra/cg.f90 File Reference

Contains subroutines cg(), vec_prod() and cal_dif().

Data Types

- interface cg__interface

Functions/Subroutines

- subroutine cg (X, B, icl, mat_prod, precond, cmp_precond, eps, nite, dif)

  This routine performs a preconditioned conjugate gradient method to solve AX=B.

- subroutine vec_prod (pv, x1, x2)

  Computes the vector product x1 ∗ x2.

- subroutine cal_dif (dif, X, PP, alpha)

  Computes the maximum relative difference for the X variable between two iterations.

9.1.1 Detailed Description

Contains subroutines cg(), vec_prod() and cal_dif().

Definition in file cg.f90.

9.1.2 Function Documentation

9.1.2.1 subroutine cal_dif (real *dif*, real,dimension(nxmin(1) *X*, real,dimension(nxmin(1) *PP*, real *alpha*)

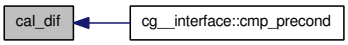
Computes the maximum relative difference for the X variable between two iterations.

Definition at line 207 of file cg.f90.

References parameters::nx, and divers::verbose.

Referenced by cg__interface::cmp_precond().

Here is the caller graph for this function:



9.1.2.2 subroutine cg (real,dimension(nxmin(1) *X*, real,dimension(nxmin(1) *B*, integer,dimension(6) *icl*, cg__interface *mat_prod*, logical,intent(in) *precond*, cg__interface *cmp_precond*, real,intent(in) *eps*, integer,intent(out) *nite*, real,intent(out) *dif*)

This routine performs a preconditioned conjugate gradient method to solve AX=B.

Parameters:

precond if .true. perform matrix preconditionning

eps convergence criterion

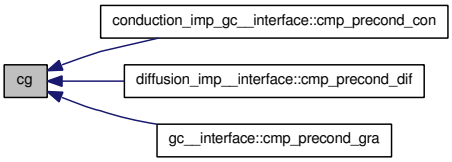
nite number of iterations reached

dif residual norm after convergence

Definition at line 15 of file cg.f90.

Referenced by conduction_imp_gc__interface::cmp_precond_con(), diffusion_imp__interface::cmp_precond_dif(), and gc__interface::cmp_precond_gra().

Here is the caller graph for this function:



9.1.2.3 subroutine vec_prod (real *pv*, real,dimension(nxmin(1) *x1*, real,dimension(nxmin(1) *x2*)

Computes the vector product x1 $*$ x2.

Definition at line 152 of file cg.f90.

References parameters::nx, and divers::verbose.

9.2 conduction/conduction.f90 File Reference

Contains subroutine conduction_ex().

Functions/Subroutines

- subroutine conduction_ex (imin, imax, jmin, jmax, kmin, kmax)

  Computes the explicit thermal gas conduction.

9.2.1 Detailed Description

Contains subroutine conduction_ex().

Definition in file conduction.f90.

9.2.2 Function Documentation

9.2.2.1 subroutine conduction_ex (integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,intent(in) *kmax*)

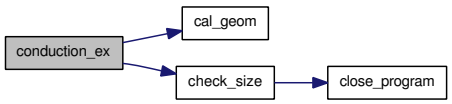
Computes the explicit thermal gas conduction.

Definition at line 14 of file conduction.f90.

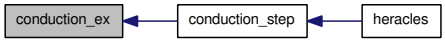
References var::B, varold::Bold, cal_geom(), check_size(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, var_loc::Eloc, varold::Eold, cl::icl, parameters::ndim, parameters::nx, var_loc::rholoc, varold::rhoold, var_loc::rhouloc, varold::rhouold, var_loc::Tloc, var_loc::uloc, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by conduction_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.3 conduction/conduction_imp_gc.f90 File Reference

Contains subroutines conduction_imp_gc(), Mat_prod_con() and cmp_precond_con().

Data Types

- interface conduction_imp_gc__interface

Functions/Subroutines

- subroutine conduction_imp_gc

  Computes the implicit thermal gas conduction using the conjugate gradient method.

- subroutine Mat_prod_con (T, AT)

  Computes the matrix product $A * X$.

- subroutine cmp_precond_con (RR, ZZ)

  This routine computes the preconditioned matrix.

9.3.1 Detailed Description

Contains subroutines conduction_imp_gc(), Mat_prod_con() and cmp_precond_con().

Definition in file conduction_imp_gc.f90.

9.3.2 Function Documentation

9.3.2.1 subroutine cmp_precond_con (real,dimension(nxmin(1) *RR*, real,dimension(nxmin(1) *ZZ*)

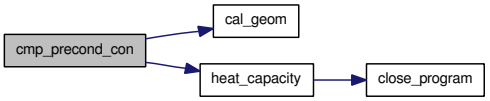
This routine computes the preconditioned matrix.

Definition at line 271 of file conduction_imp_gc.f90.

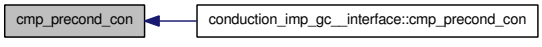
References var::B, cal_geom(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, heat_capacity(), parameters::ndim, parameters::nx, var::rho, var::rhou, var_loc::Tloc, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by conduction_imp_gc__interface::cmp_precond_con().

Here is the call graph for this function:



Here is the caller graph for this function:



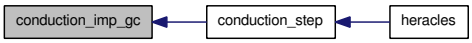
9.3.2.2 subroutine conduction_imp_gc ()

Computes the implicit thermal gas conduction using the conjugate gradient method.

Definition at line 16 of file conduction_imp_gc.f90.

Referenced by conduction_step().

Here is the caller graph for this function:



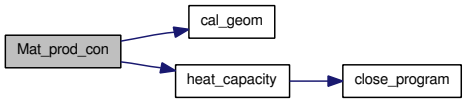
9.3.2.3 subroutine Mat_prod_con (real,dimension(nxmin(1) *T*, real,dimension(nxmin(1) *AT*)

Computes the matrix product A $*$ X.

Definition at line 130 of file conduction_imp_gc.f90.

References var::B, cal_geom(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, heat_capacity(), parameters::ndim, parameters::nx, var::rho, var::rhou, var_loc::Tloc, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Here is the call graph for this function:



9.4 conduction/conduction_step.f90 File Reference

Contains subroutine conduction_step().

Functions/Subroutines

- subroutine conduction_step (t_conduc, t_comm)

  Performs the conduction evolution at each timestep.

9.4.1 Detailed Description

Contains subroutine conduction_step().

Definition in file conduction_step.f90.

9.4.2 Function Documentation

9.4.2.1 subroutine conduction_step (real *t_conduc*, real *t_comm*)

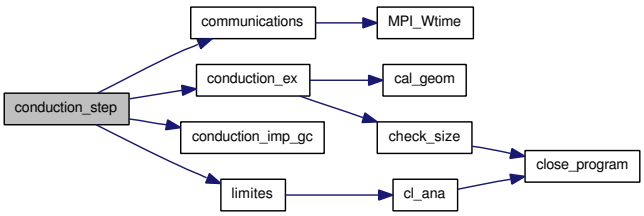
Performs the conduction evolution at each timestep.

Definition at line 14 of file conduction_step.f90.

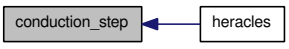
References communications(), conduction_ex(), conduction_imp_gc(), var::E, varold::Eold, conduction::implicit_con, limites(), divers::nsx, divers::nsy, divers::nsz, parameters::nx, var::rho, varold::rhoold, var::rhou, and varold::rhouold.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:



9.5 conduction/init_conduction.f90 File Reference

Contains subroutine init_conduction().

Functions/Subroutines

- subroutine init_conduction

  Performs the initialisation for including thermal conduction in the simulation.

9.5.1 Detailed Description

Contains subroutine init_conduction().

Definition in file init_conduction.f90.

9.5.2 Function Documentation

9.5.2.1 subroutine init_conduction ()

Performs the initialisation for including thermal conduction in the simulation.

Definition at line 15 of file init_conduction.f90.

References cl_con::icl_con_glob, conduction::implicit_con, parameters::ndim, conduction::precond_con, cl_con::T_l, conduction::vardt_con, and conduction::varrel_con.

Referenced by init_phys().

Here is the caller graph for this function:

9.6 conduction/kappa_con.f90 File Reference

Contains function cal_kappa_con().

Functions/Subroutines

- real cal_kappa_con (rho, T)

  Computes the Spitzer opacity for thermal conduction.

9.6.1 Detailed Description

Contains function cal_kappa_con().

Definition in file kappa_con.f90.

9.6.2 Function Documentation

9.6.2.1 real cal_kappa_con (real *rho*, real *T*)

Computes the Spitzer opacity for thermal conduction.

Definition at line 14 of file kappa_con.f90.

9.7 conduction/modules_con.f90 File Reference

Contains modules cl_con and conduction.

Modules

- module cl_con

  Contains the variables used for the conduction boundary conditions.

- module conduction

  Contains the variables used for the thermal conduction.

Variables

- integer, dimension(6) cl_con::icl_con

  Local conduction boundary conditions.

- integer, dimension(6) cl_con::icl_con_glob

  Global conduction boundary conditions.

- real, dimension(:), allocatable cl_con::T_l

  Temperature in the conduction boundaries.

- real conduction::varrel_con

  real :: tolx_con !<

- real conduction::vardt_con

  real :: dt_con_exp !<

- real conduction::dt_con_imp

  real :: dt_con !<

- logical conduction::implicit_con

  If .true.

- integer conduction::nitetot_con

  Conduction total number of iterations.

- logical conduction::precond_con

  Perform matrix preconditionning if .true.

9.7.1 Detailed Description

Contains modules cl_con and conduction.

Definition in file modules_con.f90.

9.8 conduction/pasdt_con.f90 File Reference

Contains subroutine pasdt_con().

Functions/Subroutines

- subroutine pasdt_con

  Computes the maximum timestep dictated by thermal conduction.

9.8.1 Detailed Description

Contains subroutine pasdt_con().

Definition in file pasdt_con.f90.

9.8.2 Function Documentation

9.8.2.1 subroutine pasdt_con ()

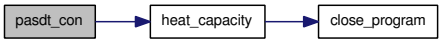
Computes the maximum timestep dictated by thermal conduction.

Definition at line 15 of file pasdt_con.f90.

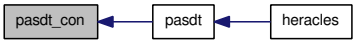
References unites::an, var::B, conduction::dt_con_imp, var::E, geom::geom_dir, heat_capacity(), conduction::implicit_con, parameters::ndim, parameters::nx, var::rho, var::rhou, divers::verbose, and geom::x.

Referenced by pasdt().

Here is the call graph for this function:



Here is the caller graph for this function:



9.9 gravity/gc.f90 File Reference

Contains subroutines gc(), mat_prod_gra() and cmp_precond_gra().

Data Types

- interface gc__interface

Functions/Subroutines

- subroutine gc (init)

  Performs the conjugate gradient algorithm for the gravity potential calculation.

- subroutine mat_prod_gra (T, AT)

  Performs the matrix product.

- subroutine cmp_precond_gra (RR, ZZ)

  This routine computes the preconditioned matrix.

9.9.1 Detailed Description

Contains subroutines gc(), mat_prod_gra() and cmp_precond_gra(). This file contains the routines necessary to perform conjugate gradients when including self-gravity in the simulation.

Definition in file gc.f90.

9.9.2 Function Documentation

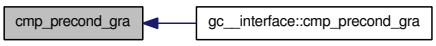
9.9.2.1 subroutine cmp_precond_gra (real,dimension(nxmin(1) *RR*, real,dimension(nxmin(1) *ZZ*)

This routine computes the preconditioned matrix.

Definition at line 188 of file gc.f90.

Referenced by gc__interface::cmp_precond_gra().

Here is the caller graph for this function:



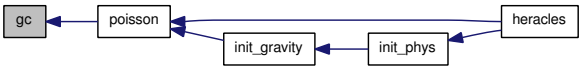
9.9.2.2 subroutine gc (logical *init*)

Performs the conjugate gradient algorithm for the gravity potential calculation.

Definition at line 20 of file gc.f90.

Referenced by poisson().

Here is the caller graph for this function:



9.9.2.3 subroutine mat_prod_gra (real,dimension(nxmin(1) *T*, real,dimension(nxmin(1) *AT*)

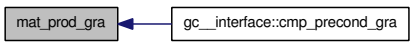
Performs the matrix product.

Definition at line 124 of file gc.f90.

References parameters::nx, and geom::x.

Referenced by gc__interface::cmp_precond_gra().

Here is the caller graph for this function:



9.10 gravity/grav_predictor.f90 File Reference

Contains subroutine grav_predictor().

Functions/Subroutines

- subroutine grav_predictor (u, phi, dt, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Performs prediction step in the gravity calculation.

9.10.1 Detailed Description

Contains subroutine grav_predictor().

Definition in file grav_predictor.f90.

9.10.2 Function Documentation

9.10.2.1 subroutine grav_predictor (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax,1:n_vit) *u*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *phi*, real *dt*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

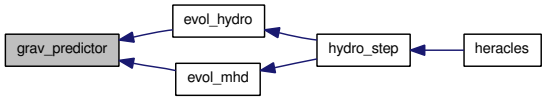
Performs prediction step in the gravity calculation.

Definition at line 14 of file grav_predictor.f90.

References geom::geom_dir, const::half, parameters::ndim, geom::Spherique, and geom::xcloc.

Referenced by evol_hydro(), and evol_mhd().

Here is the caller graph for this function:



9.11 gravity/init_gravity.f90 File Reference

Contains subroutine init_gravity().

Functions/Subroutines

- subroutine init_gravity

  Initialises all the gravity parameters and variables.

9.11.1 Detailed Description

Contains subroutine init_gravity().

Definition in file init_gravity.f90.

9.11.2 Function Documentation

9.11.2.1 subroutine init_gravity ()

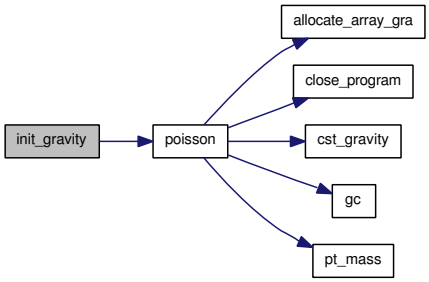
Initialises all the gravity parameters and variables.

Definition at line 14 of file init_gravity.f90.

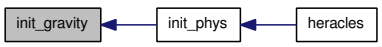
References unites::centimetre, unites::gramme, gravity::gravity_params, gravity::gravity_type, gravity::ISOLE, poisson(), and unites::seconde.

Referenced by init_phys().

Here is the call graph for this function:



Here is the caller graph for this function:



9.12 gravity/limites_gra.f90 File Reference

Contains subroutines limites_gra() and cal_barycentre().

Functions/Subroutines

- subroutine limites_gra (Tab)

  Defines the boundary conditions for the gravity.

- subroutine cal_barycentre (xbar, ybar, zbar, pbar)

  Computes the barycentre of the grid.

9.12.1 Detailed Description

Contains subroutines limites_gra() and cal_barycentre().

Definition in file limites_gra.f90.

9.12.2 Function Documentation

9.12.2.1 subroutine cal_barycentre (real *xbar*, real *ybar*, real *zbar*, real *pbar*)

Computes the barycentre of the grid.

Definition at line 291 of file limites_gra.f90.

References parameters::nx, var::rho, divers::verbose, and geom::x.

Referenced by gc__interface::cmp_precond_gra(), and limites_gra().

Here is the caller graph for this function:

9.12.2.2 subroutine limites_gra (real,dimension(nxmin(1) *Tab*)

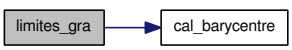
Defines the boundary conditions for the gravity.

Definition at line 14 of file limites_gra.f90.

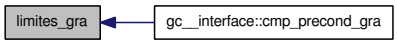
References cal_barycentre(), unites::G, gravity::ISOLE, parameters::ndim, parameters::nx, divers::verbose, and geom::x.

Referenced by gc__interface::cmp_precond_gra().

Here is the call graph for this function:



Here is the caller graph for this function:



9.13 gravity/modules_gra.f90 File Reference

Contains module gravity and subroutine allocate_array_gra().

Modules

- module gravity

  Contains all the gravity variables.

Functions/Subroutines

- subroutine allocate_array_gra

  Allocates the gravity arrays.

Variables

- integer gravity::gravity_type
- integer gravity::ISOLE
- real, dimension(:,:,:), allocatable gravity::phi

  Gravitational potential Φ.

- real, dimension(:,:,:), allocatable gravity::phiold

  Old graviational potential Φ_{old}.

- real, dimension(4) gravity::gravity_params

  Gravity parameters.

9.13.1 Detailed Description

Contains module gravity and subroutine allocate_array_gra().

Definition in file modules_gra.f90.

9.13.2 Function Documentation

9.13.2.1 subroutine allocate_array_gra ()

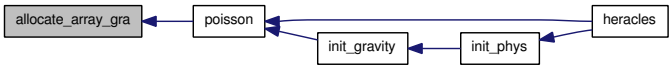
Allocates the gravity arrays.

Definition at line 33 of file modules_gra.f90.

References parameters::nxmax, parameters::nxmin, gravity::phi, and gravity::phiold.

Referenced by poisson().

Here is the caller graph for this function:



9.14 gravity/pasdt_grav.f90 File Reference

Contains subroutine pasdt_grav().

Functions/Subroutines

- subroutine pasdt_grav

  Computes the minimum free-fall time in the grid and calculates a free-fall limited timestep.

9.14.1 Detailed Description

Contains subroutine pasdt_grav().

Definition in file pasdt_grav.f90.

9.14.2 Function Documentation

9.14.2.1 subroutine pasdt_grav ()

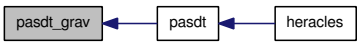
Computes the minimum free-fall time in the grid and calculates a free-fall limited timestep.

Definition at line 15 of file pasdt_grav.f90.

References divers::dt_ff, unites::G, var::rho, and divers::verbose.

Referenced by pasdt().

Here is the caller graph for this function:



9.15 gravity/poisson.f90 File Reference

Contains subroutines poisson(), pt_mass() and cst_gravity().

Functions/Subroutines

- subroutine poisson (init)

  This routine calls the potential computation routines according to gravity_type.

- subroutine pt_mass

  This routine computes a point-mass gravitational potential.

- subroutine cst_gravity

  This routine computes a potential for a constant gravitational field.

9.15.1 Detailed Description

Contains subroutines poisson(), pt_mass() and cst_gravity(). This file contains the routines which compute different types of gravitational potential.

Definition in file poisson.f90.

9.15.2 Function Documentation

9.15.2.1 subroutine cst_gravity ()

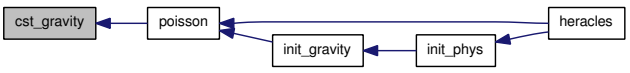
This routine computes a potential for a constant gravitational field. g = gravity_params(1:ndim)

Definition at line 137 of file poisson.f90.

References gravity::gravity_params, const::half, parameters::ndim, parameters::nxmax, parameters::nxmin, gravity::phi, and geom::x.

Referenced by poisson().

Here is the caller graph for this function:



9.15.2.2 subroutine poisson (logical *init*)

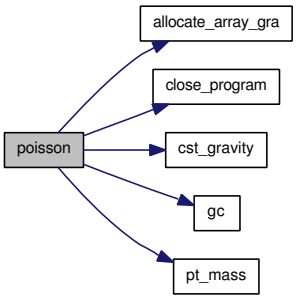
This routine calls the potential computation routines according to gravity_type.

Definition at line 19 of file poisson.f90.

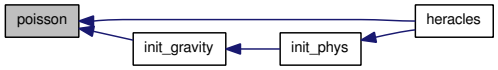
References allocate_array_gra(), close_program(), cst_gravity(), gc(), gravity::gravity_type, para::mype, gravity::phi, gravity::phiold, and pt_mass().

Referenced by heracles(), and init_gravity().

Here is the call graph for this function:



Here is the caller graph for this function:



9.15.2.3 subroutine pt_mass ()

This routine computes a point-mass gravitational potential. Mass = gravity_params(1)

Cartesian geometry: position given by gravity_params(2:4)

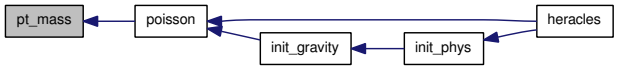
Cylindrical & Spherical geometry: position at grid center

Definition at line 73 of file poisson.f90.

References geom::Cartesien, geom::Cylindrique, unites::G, gravity::gravity_params, const::half, parameters::ndim, parameters::nxmax, parameters::nxmin, gravity::phi, geom::Spherique, and geom::x.

Referenced by poisson().

Here is the caller graph for this function:



9.16 gravity/update_gravity.f90 File Reference

Contains subroutine update_gravity().

Functions/Subroutines

- subroutine update_gravity

  This routine includes the gravity source term in the rhou and E hydro variables.

9.16.1 Detailed Description

Contains subroutine update_gravity().

Definition in file update_gravity.f90.

9.16.2 Function Documentation

9.16.2.1 subroutine update_gravity ()

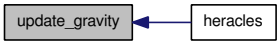
This routine includes the gravity source term in the rhou and E hydro variables.

Definition at line 15 of file update_gravity.f90.

References divers::dt, var::E, geom::geom_dir, const::half, parameters::ndim, parameters::nx, gravity::phi, gravity::phiold, var::rho, varold::rhoold, var::rhou, varold::rhouold, geom::Spherique, and geom::x.

Referenced by heracles().

Here is the caller graph for this function:



9.17 hdf/film_hdf.f90 File Reference

Contains subroutine film_hdf().

Functions/Subroutines

- subroutine film_hdf (kk)

  Writes hdf5 frames to create a movie.

9.17.1 Detailed Description

Contains subroutine film_hdf().

Definition in file film_hdf.f90.

9.17.2 Function Documentation

9.17.2.1 subroutine film_hdf (integer *kk*)

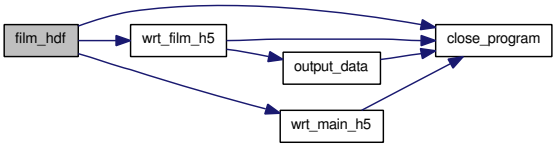
Writes hdf5 frames to create a movie.

Definition at line 14 of file film_hdf.f90.

References close_program(), mfilm::dt_film, mfilm::ifilm, para::mype, mfilm::t_film, divers::temps, divers::verbose, wrt_film_h5(), and wrt_main_h5().

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.18 hdf/rd_restart_h5.f90 File Reference

Contains subroutine rd_restart_h5().

Functions/Subroutines

- subroutine rd_restart_h5 (k)

  Reads a hdf5 output file to perform a simulation restart.

9.18.1 Detailed Description

Contains subroutine rd_restart_h5().

Definition in file rd_restart_h5.f90.

9.18.2 Function Documentation

9.18.2.1 subroutine rd_restart_h5 (integer *k*)

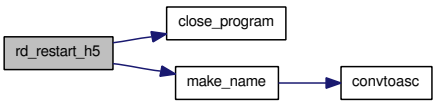
Reads a hdf5 output file to perform a simulation restart.

Definition at line 15 of file rd_restart_h5.f90.

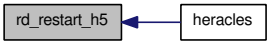
References var::B, close_program(), divers::CONDUC, divers::DIFFU, divers::dt, divers::dtold, var::E, gammas::gamma, geom::geometrie, divers::GRAV, divers::i_restart, make_name(), divers::mu, para::mype, parameters::N_vit, parameters::Nbuf, parameters::ndim, geom::nshift_gr, parameters::nx, parameters::nx_glob, divers::RAY, var::rho, var::rhou, geom::shift_gr, divers::temps, unites_sortie::u_dens, unites_sortie::u_Fr, unites_sortie::u_L, unites_sortie::u_Mom, unites_sortie::u_T, geom::x, para::x_end, geom::x_glob, and para::x_start.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:



9.19 hdf/rdwrt_h5.f90 File Reference

Contains module rdwrt_h5 and subroutines dump_1d_array_int_h5(), dump_1d_array_h5(), dump_3d_-array_h5(), dump_4d_array_h5(), dump_1D_array_string_h5(), get_1d_array_int_h5(), get_1d_array_-h5(), get_3d_array_h5() and get_4d_array_h5(), get_1D_array_string_h5().

Modules

- module rdwrt_h5

  Contains reading and writing hdf5 subroutines.

9.19.1 Detailed Description

Contains module rdwrt_h5 and subroutines dump_1d_array_int_h5(), dump_1d_array_h5(), dump_3d_-array_h5(), dump_4d_array_h5(), dump_1D_array_string_h5(), get_1d_array_int_h5(), get_1d_array_-h5(), get_3d_array_h5() and get_4d_array_h5(), get_1D_array_string_h5().

Definition in file rdwrt_h5.f90.

9.20 hdf/utils_h5.f90 File Reference

Contains subroutines output_data(), get_filename(), get_dataexist() and get_nl().

Functions/Subroutines

- subroutine output_data (filename, dataexist)

  Contains reading and writing hdf5 subroutines.

9.20.1 Detailed Description

Contains subroutines output_data(), get_filename(), get_dataexist() and get_nl().

Definition in file utils_h5.f90.

9.20.2 Function Documentation

9.20.2.1 subroutine output_data (character(len=∗) *filename*, integer,dimension(8) *dataexist*)

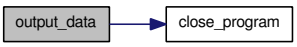
Contains reading and writing hdf5 subroutines.

Definition at line 373 of file utils_h5.f90.

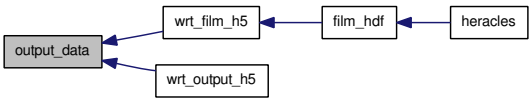
References close_program(), and para::mype.

Referenced by wrt_film_h5(), and wrt_output_h5().

Here is the call graph for this function:



Here is the caller graph for this function:



9.21 hdf/wrt_film_h5.f90 File Reference

Contains subroutine wrt_film_h5().

Functions/Subroutines

- subroutine wrt_film_h5 (k)

  Writes the hdf5 output frames to create a movie.

9.21.1 Detailed Description

Contains subroutine wrt_film_h5().

Definition in file wrt_film_h5.f90.

9.21.2 Function Documentation

9.21.2.1 subroutine wrt_film_h5 (integer *k*)

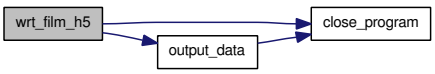
Writes the hdf5 output frames to create a movie.

Definition at line 14 of file wrt_film_h5.f90.

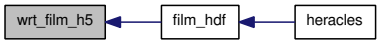
References close_program(), divers::datanames, para::mype, divers::ndata, output_data(), and mfilm::t_-film.

Referenced by film_hdf().

Here is the call graph for this function:



Here is the caller graph for this function:



9.22 hdf/wrt_main_h5.f90 File Reference

Contains subroutine wrt_main_h5().

Functions/Subroutines

- subroutine wrt_main_h5 (k)

  Create_main_hdf5_file create "myjob000_main_hdf5_file".

9.22.1 Detailed Description

Contains subroutine wrt_main_h5().

Definition in file wrt_main_h5.f90.

9.22.2 Function Documentation

9.22.2.1 subroutine wrt_main_h5 (integer *k*)

Create_main_hdf5_file create "myjob000_main_hdf5_file". This program creates a the main file of the job containing 11 datasets which are: 1 character∗20 name_of_job 2 character∗80 comments 3 integer, dimension(7) job_params 4 character∗80,dimension(ntotal) datanames 5 character∗80,dimension(ntotal) dataunits 6 integer , dimension(ntotal) fourdim 7 data_int list of integer parameters 8 data_real list of real parameters 9 xglob array 10 yglob array 11 zglob array 12 shift_gr array 13 nshift_gr array

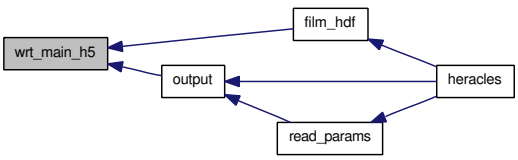
Definition at line 30 of file wrt_main_h5.f90.

References close_program(), divers::comments, divers::CONDUC, divers::datanames, divers::dataunits, divers::DIFFU, divers::dt, divers::dtold, divers::fourdim, gammas::gamma, geom::geometrie, divers::GRAV, divers::iout, divers::irepr, divers::mu, para::mype, parameters::N_vit, divers::name_of_job, parameters::Nbuf, para::ncpu, divers::ndata, parameters::ndim, divers::noutd, divers::nreprd, geom::nshift_gr, divers::nsupp, parameters::nx, parameters::nx_glob, divers::RAY, geom::shift_gr, divers::suppdim, divers::suppnames, divers::suppunits, divers::temps, unites_sortie::u_L, unites_sortie::u_T, para::x_end, geom::x_glob, and para::x_start.

Referenced by film_hdf(), and output().

Here is the call graph for this function:

Here is the caller graph for this function:



9.23 hdf/wrt_output_h5.f90 File Reference

Contains subroutine wrt_output_h5().

Functions/Subroutines

- subroutine wrt_output_h5 (k)

  Controls hdf5 outputs.

9.23.1 Detailed Description

Contains subroutine wrt_output_h5().

Definition in file wrt_output_h5.f90.

9.23.2 Function Documentation

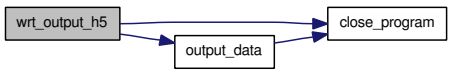
9.23.2.1 subroutine wrt_output_h5 (integer *k*)

Controls hdf5 outputs.

Definition at line 14 of file wrt_output_h5.f90.

References close_program(), divers::datanames, divers::iout, para::mype, divers::ndata, output_data(), and divers::tout.

Here is the call graph for this function:



9.24 hdf/wrt_restart_h5.f90 File Reference

Contains subroutine wrt_restart_h5().

Functions/Subroutines

- subroutine wrt_restart_h5 (k)

  Writes an hdf5 output file which can be used for performing a simulation restart.

9.24.1 Detailed Description

Contains subroutine wrt_restart_h5().

Definition in file wrt_restart_h5.f90.

9.24.2 Function Documentation

9.24.2.1 subroutine wrt_restart_h5 (integer *k*)

Writes an hdf5 output file which can be used for performing a simulation restart.

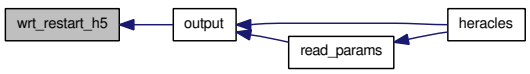
Definition at line 15 of file wrt_restart_h5.f90.

References var::B, close_program(), divers::CONDUC, divers::DIFFU, divers::dt, divers::dtold, var::E, gammas::gamma, geom::geometrie, divers::GRAV, divers::iout, divers::mu, para::mype, parameters::N_-vit, parameters::Nbuf, parameters::ndim, geom::nshift_gr, parameters::nx, parameters::nx_glob, divers::RAY, var::rho, var::rhou, geom::shift_gr, divers::temps, divers::tout, unites_sortie::u_dens, unites_sortie::u_Fr, unites_sortie::u_L, unites_sortie::u_Mom, unites_sortie::u_T, geom::x, para::x_end, geom::x_glob, and para::x_start.

Referenced by output().

Here is the call graph for this function:

Here is the caller graph for this function:



9.25 hydro/eos.f90 File Reference

Contains subroutines init_eos(), pressure(), temperature(), sound_speed(), energy() and heat_capacity().

Functions/Subroutines

- subroutine init_eos

  Initialises the gas equation of state.

- subroutine Pressure (rho, e, P, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Returns the pressure of an ideal gas array.

- subroutine Temperature (rho, e, T, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Returns the temperature of an ideal gas array.

- subroutine Sound_speed (rho, e, cs, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Returns the sound speed of an ideal gas array.

- subroutine Energy (rho, T, e, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Returns the thermal energy of an ideal gas array.

- subroutine heat_capacity (rho, T, cv, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Returns the volumic heat capacity of an ideal gas array.

9.25.1 Detailed Description

Contains subroutines init_eos(), pressure(), temperature(), sound_speed(), energy() and heat_capacity().

Definition in file eos.f90.

9.25.2 Function Documentation

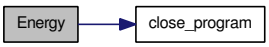
9.25.2.1 subroutine Energy (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *T*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *e*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

Returns the thermal energy of an ideal gas array.

Definition at line 129 of file eos.f90.

References close_program(), gammas::gamma, divers::mu, para::mype, and unites::uma.

Here is the call graph for this function:



9.25.2.2 subroutine heat_capacity (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *T*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *cv*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

Returns the volumic heat capacity of an ideal gas array.

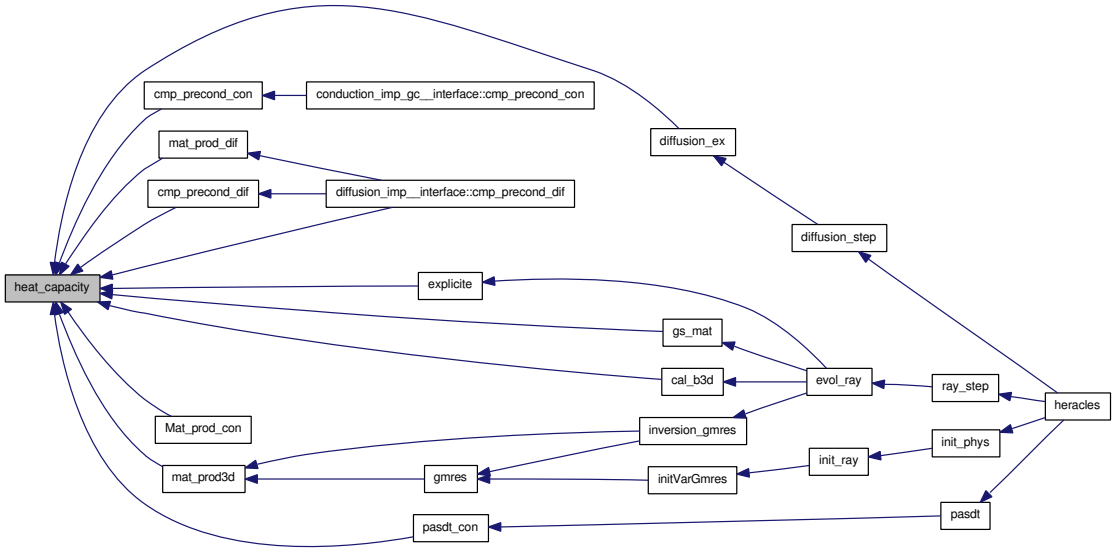
Definition at line 160 of file eos.f90.

References close_program(), gammas::gamma, divers::mu, para::mype, and unites::uma.

Referenced by cal_b3d(), cmp_precond_con(), cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), diffusion_ex(), explicite(), gs_mat(), mat_prod3d(), Mat_prod_con(), mat_prod_dif(), and pasdt_con().

Here is the call graph for this function:

Here is the caller graph for this function:



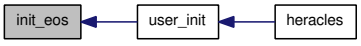
9.25.2.3 subroutine init_eos ()

Initialises the gas equation of state.

Definition at line 16 of file eos.f90.

Referenced by user_init().

Here is the caller graph for this function:



9.25.2.4 subroutine Pressure (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *e*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *P*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

Returns the pressure of an ideal gas array.

Definition at line 40 of file eos.f90.

References gammas::gamma.

Referenced by evol_hydro().

Here is the caller graph for this function:

9.25.2.5 subroutine Sound_speed (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *e*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *cs*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

Returns the sound speed of an ideal gas array.

Definition at line 100 of file eos.f90.

References gammas::gamma.

Referenced by evol_hydro().

Here is the caller graph for this function:

9.25.2.6 subroutine Temperature (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *e*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *T*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

Returns the temperature of an ideal gas array.

Definition at line 69 of file eos.f90.

References gammas::gamma, divers::mu, param_ini::Temperature_isotherme, and unites::uma.

9.26 hydro/evol_hydro.f90 File Reference

Contains subroutines evol_hydro() and pente_minmod().

Functions/Subroutines

- subroutine evol_hydro (imin, imax, jmin, jmax, kmin, kmax)

  Performs the evolution of the hydrodynamic variables over one timestep.

9.26.1 Detailed Description

Contains subroutines evol_hydro() and pente_minmod().

Definition in file evol_hydro.f90.

9.26.2 Function Documentation

9.26.2.1 subroutine evol_hydro (integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,intent(in) *kmax*)

Performs the evolution of the hydrodynamic variables over one timestep. The update is done over the range (imin:imax,jmin:jmax,kmin:kmax) which is not necessarily the entire range of cells which the cpu holds.

The routine:

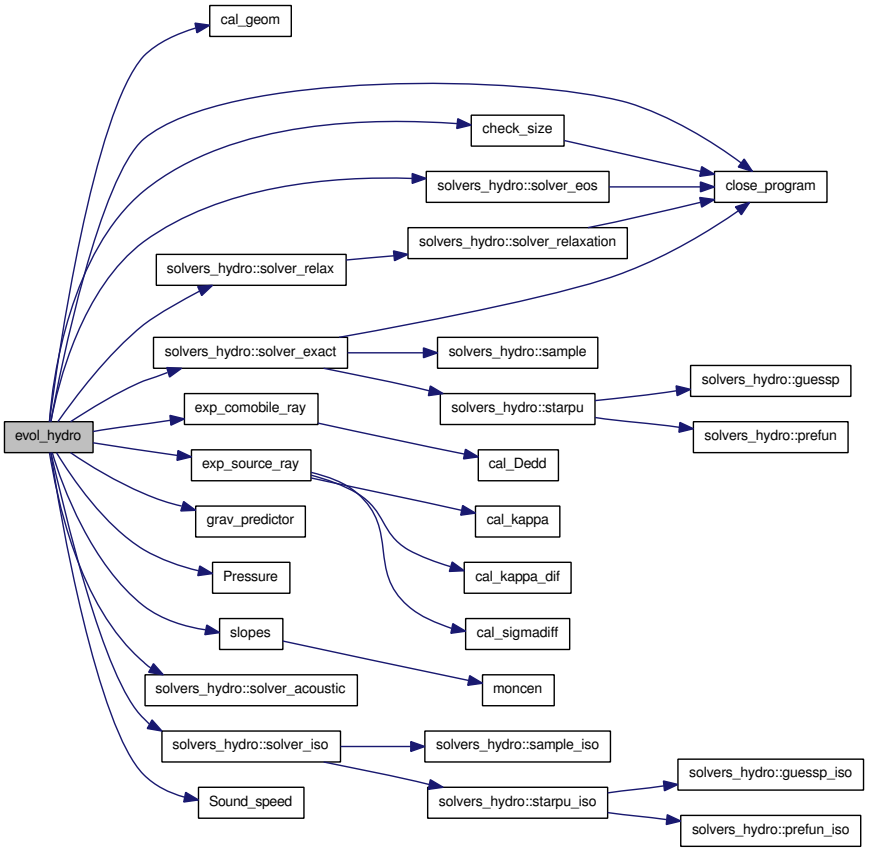
- computes slopes and interface values
- performs a half-timestep prediction
- computes Riemann fluxes using the Riemann solver
- updates the variables using computed fluxes
- eventually computes explicit comoving terms for radiation coupling using computed interface velocity values.

Definition at line 27 of file evol_hydro.f90.

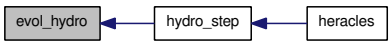
References cal_geom(), geom::Cartesien, check_size(), close_program(), gammas::Cs_iso2, divers::DIFFU, geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, var_loc::Einloc, var_loc::Eloc, varold::Eold, exp_comobile_ray(), exp_source_ray(), var_loc::Fxloc, varold::Fxold, gammas::gamma, geom::geom_dir, divers::GRAV, grav_predictor(), cl::icl, para::mype, parameters::ndim, parameters::nFx, parameters::nx, gravity::phi, var_loc::philoc, var_loc::Ploc, Pressure(), divers::RAY, var::rho, var_loc::rholoc, varold::rhoold, var::rhou, var_loc::rhouloc, varold::rhouold, parameters::slope_type, slopes(), divers::Solver, solvers_hydro::solver_acoustic(), solvers_hydro::solver_eos(), solvers_hydro::solver_exact(), solvers_hydro::solver_iso(), solvers_hydro::solver_relax(), Sound_speed(), var_loc::uloc, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by hydro_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.27 hydro/fill_new_arrays.f90 File Reference

Contains subroutine fill_new_arrays().

Functions/Subroutines

- subroutine fill_new_arrays

  Fill additional arrays by user if needed.

9.27.1 Detailed Description

Contains subroutine fill_new_arrays().

Definition in file fill_new_arrays.f90.

9.27.2 Function Documentation

9.27.2.1 subroutine fill_new_arrays ()

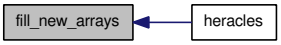
Fill additional arrays by user if needed. Arrays are names supp1 to supp5 and declared in modules.f90. They should be given names in namelist 'output' using the variable suppnames.

Definition at line 17 of file fill_new_arrays.f90.

References divers::nsupp, var::rho, var::supp1, and var::supp2.

Referenced by heracles().

Here is the caller graph for this function:



9.28 hydro/histogramme.f90 File Reference

Contains module histo and subroutines init_histo() and cal_histogramme().

Modules

- module histo

  Contains variables for histogram computations.

Functions/Subroutines

- subroutine init_histo

  Initialises histogram parameters.

- subroutine cal_histogramme

  Computes histogram statistics.

Variables

- integer histo::Npoint

  Npoint.

- real, dimension(:), allocatable histo::hP

  hp

- real, dimension(:), allocatable histo::hN

  hn

- real, dimension(:), allocatable histo::hT

  ht

- real, dimension(:), allocatable histo::P

  p

- real, dimension(:), allocatable histo::N

  n

- real, dimension(:), allocatable histo::T

  t

- real, dimension(:), allocatable histo::hPg

  hPg

- real, dimension(:), allocatable histo::hNg

  hNg

- real, dimension(:), allocatable histo::hTg

  hTg

- real histo::Pmin

  Pmin.

- real histo::Pmax

  Pmax.

- real histo::r_P

  r_P

- real histo::lr_P

  lr_P

- real histo::Nmin

  Nmin.

- real histo::Nmax

  Nmax.

- real histo::r_N

  r_N

- real histo::lr_N

  lr_N

- real histo::Tmin

  Tmin.

- real histo::Tmax

  Tmax.

- real histo::r_T

  r_T

- real histo::lr_T

  lr_T

9.28.1 Detailed Description

Contains module histo and subroutines init_histo() and cal_histogramme(). This file contains the routines to compute histogram statistics on hydro variables.

Definition in file histogramme.f90.

9.28.2 Function Documentation

9.28.2.1 subroutine cal_histogramme ()

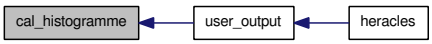
Computes histogram statistics.

Definition at line 121 of file histogramme.f90.

References unites::an, unites::cm3, var::E, unites::erg, gammas::gamma, unites::gramme, histo::hN, histo::hNg, histo::hP, histo::hPg, histo::hT, histo::hTg, histo::lr_N, histo::lr_P, histo::lr_T, divers::mu, para::mype, parameters::ndim, histo::Nmin, histo::Npoint, parameters::nx, histo::Pmin, var::rho, var::rhou, divers::temps, and histo::Tmin.

Referenced by user_output().

Here is the caller graph for this function:



9.28.2.2 subroutine init_histo ()

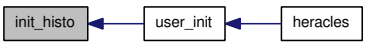
Initialises histogram parameters.

Definition at line 52 of file histogramme.f90.

References histo::hN, histo::hNg, histo::hP, histo::hPg, histo::hT, histo::hTg, histo::lr_N, histo::lr_P, histo::lr_T, para::mype, histo::N, histo::Nmax, histo::Nmin, histo::Npoint, histo::P, histo::Pmax, histo::Pmin, histo::r_N, histo::r_P, histo::r_T, histo::T, histo::Tmax, and histo::Tmin.

Referenced by user_init().

Here is the caller graph for this function:



9.29 hydro/hydro_step.f90 File Reference

Contains subroutine hydro_step().

Functions/Subroutines

- subroutine hydro_step (t_hydro, monit)

  Determines the size of the block which will be treated in the evol_hydro and evol_mhd steps.

9.29.1 Detailed Description

Contains subroutine hydro_step().

Definition in file hydro_step.f90.

9.29.2 Function Documentation

9.29.2.1 subroutine hydro_step (real *t_hydro*, logical *monit*)

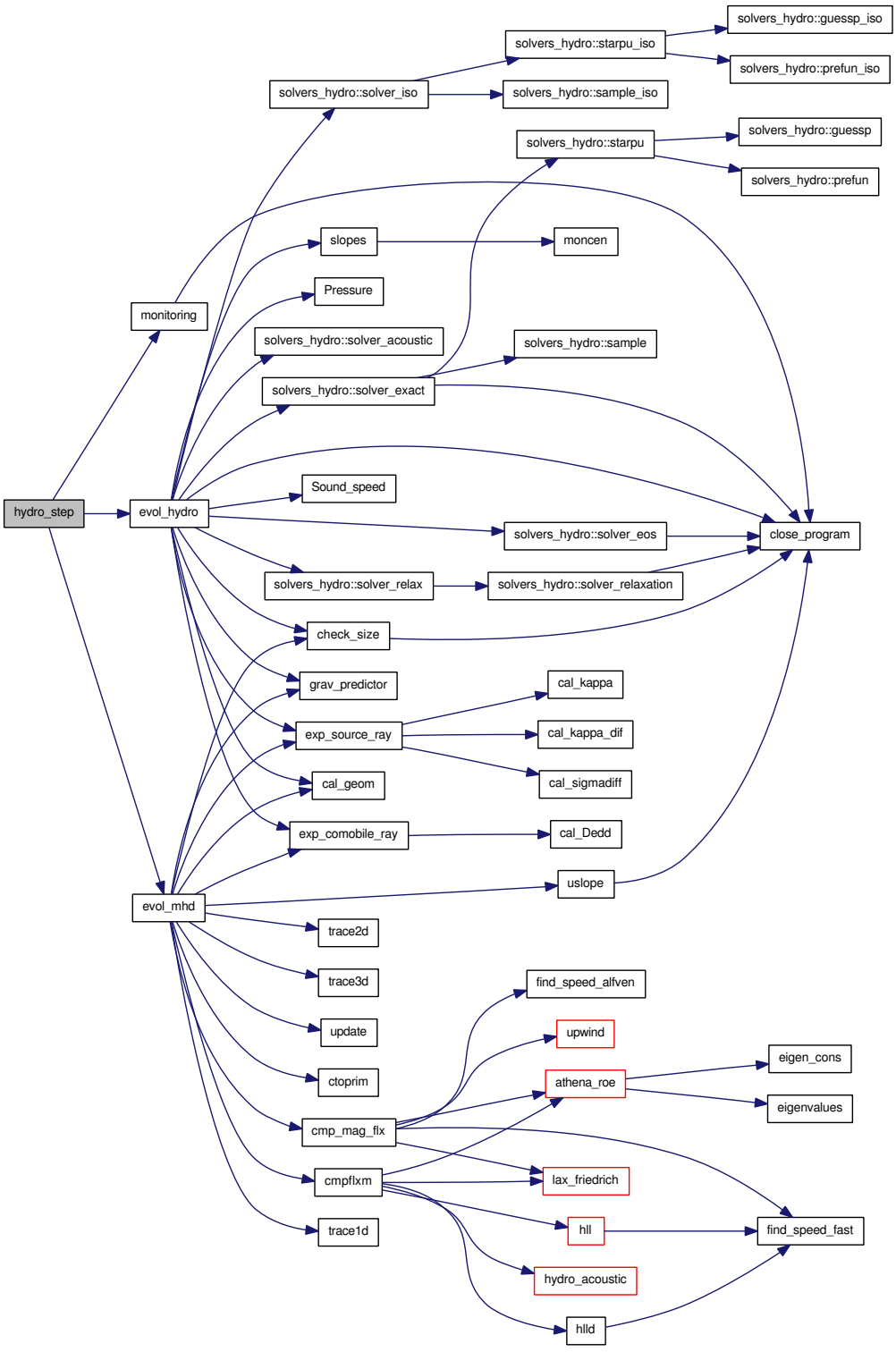
Determines the size of the block which will be treated in the evol_hydro and evol_mhd steps.

Definition at line 15 of file hydro_step.f90.

References evol_hydro(), evol_mhd(), monitoring(), divers::nsx, divers::nsy, divers::nsz, parameters::nx, and divers::verbose.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.30 hydro/monitoring.f90 File Reference

Contains subroutine monitoring().

Functions/Subroutines

- subroutine monitoring (Comment)

  Monitors the minimum, maximum and average values of the gas density, pressure, sound speed, Mach number and temperature in the entire grid.

9.30.1 Detailed Description

Contains subroutine monitoring().

Definition in file monitoring.f90.

9.30.2 Function Documentation

9.30.2.1 subroutine monitoring (character(len=50) *Comment*)

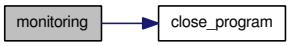
Monitors the minimum, maximum and average values of the gas density, pressure, sound speed, Mach number and temperature in the entire grid. The status of each variable is printed to screen.

Definition at line 17 of file monitoring.f90.

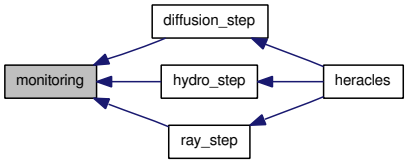
References var::B, close_program(), var::E, para::mype, parameters::nx, parameters::nx_glob, parameters::nxmax, parameters::nxmin, var::rho, and var::rhou.

Referenced by diffusion_step(), hydro_step(), and ray_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.31 hydro/pasdt_hydro.f90 File Reference

Contains subroutine pasdt_hydro().

Functions/Subroutines

- subroutine pasdt_hydro

  Computes the maximum CFL limited timestep in the grid.

9.31.1 Detailed Description

Contains subroutine pasdt_hydro().

Definition in file pasdt_hydro.f90.

9.31.2 Function Documentation

9.31.2.1 subroutine pasdt_hydro ()

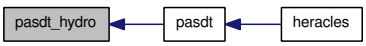
Computes the maximum CFL limited timestep in the grid.

Definition at line 14 of file pasdt_hydro.f90.

References unites::an, var::B, divers::dt_cfl, var::E, geom::geom_dir, parameters::ndim, parameters::nx, parameters::nxmax, parameters::nxmin, var::rho, var::rhou, divers::verbose, and geom::x.

Referenced by pasdt().

Here is the caller graph for this function:



9.32 hydro/rh.f90 File Reference

Contains subroutine rh() and function f().

Functions/Subroutines

- subroutine rh (rho1, u1, T1, rho2, u2, T2)
  Computes Rankine-Hugoniot jump conditions.
- real f (PP, r, r0, alpha1)
  Find the zero of f to find PP (fixed r).

9.32.1 Detailed Description

Contains subroutine rh() and function f().

Definition in file rh.f90.

9.32.2 Function Documentation

9.32.2.1 real f (real *PP*, real *r*, real *r0*, real *alpha1*)

Find the zero of f to find PP (fixed r).

Definition at line 213 of file rh.f90.

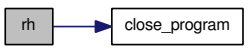
9.32.2.2 subroutine rh (real *rho1*, real *u1*, real *T1*, real *rho2*, real *u2*, real *T2*)

Computes Rankine-Hugoniot jump conditions.

Definition at line 14 of file rh.f90.

References close_program(), unites::cm3, unites::degre, gammas::gamma, unites::gramme, unites::kms, divers::mu, para::mype, unites::Pascal, divers::RAY, and unites::uma.

Here is the call graph for this function:



9.33 hydro/solvers.f90 File Reference

Contains subroutines solver_relax(), solver_relaxation(), solver_exact(), solver_cc(), solver_acoustic(), solver_eos(), solver_g(), solver_iso(), starpu(), guessp(), prefun(), sample(), starpu_iso(), guessp_iso(), prefun_iso(), sample_iso() and cal_gamma().

Modules

- module solvers_hydro

Functions/Subroutines

- subroutine solvers_hydro::solver_relax (dll, ull, pll, cll, ell, fll, drr, urr, prr, crr, err, frr, dss, uss, pss, ess, fss, vit, idim, N)

  Relaxation Riemann solver provided by Benjamin Braconnier (26/06/2008).

- subroutine solvers_hydro::solver_relaxation (dl, ul, el, pl, cl, dr, ur, er, pr, cr, ds, us, es, ps, ndim, vno)

  Relaxation Riemann solver provided by Benjamin Braconnier (26/06/2008).

- subroutine solvers_hydro::solver_exact (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Exact Riemann solver, taken from 'Riemann Solvers and Numerical Methods for Fluid Dynamics', E.

- subroutine solvers_hydro::solver_cc (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Two shock Riemann solver.

- subroutine solvers_hydro::solver_acoustic (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Acoustic (linearised) Riemann solver.

- subroutine solvers_hydro::solver_eos (dll, ull, pll, cll, ell, fll, drr, urr, prr, crr, err, frr, dss, uss, pss, ess, fss, vit, idim, N)

  Riemann solver from the FLASH paper.

- subroutine solvers_hydro::solver_g (dll, ull, pll, fll, drr, urr, prr, frr, ds, us, ps, fs, vit, idim, n)

  Variable gamma Riemann solver.

- subroutine solvers_hydro::solver_iso (dll, ull, fll, drr, urr, frr, ds, us, fs, Vit, idim, N)

  Isothermal Riemann solver.

- subroutine solvers_hydro::starpu (p, u, mpa, dl, ul, pl, cl, dr, ur, pr, cr)

  Computes the pressure and velocity in the star region.

- subroutine solvers_hydro::guessp (pm, dl, ul, pl, cl, dr, ur, pr, cr)

  Computes an intelligent first guess for the pressure.

- subroutine solvers_hydro::prefun (f, fd, p, dk, pk, ck)

  Evaluate pressure functions FL and FR.

- subroutine solvers_hydro::sample (pm, um, s, d, u, p, dl, ul, pl, cl, dr, ur, pr, cr)

  Search through the different waves to find the correct solution.

- subroutine solvers_hydro::starpu_iso (d, u, mpa, dl, ul, dr, ur, cs, cs2)

  Computes the pressure and velocity in the star region.

- subroutine solvers_hydro::guessp_iso (dm, dl, ul, dr, ur, cs, cs2)

  Computes an intelligent 'first guess' for the pressure.

- subroutine solvers_hydro::prefun_iso (f, fd, d, dk, cs)

  Computes pressure functions FL and FR.

- subroutine solvers_hydro::sample_iso (dm, um, s, d, u, dl, ul, dr, ur, cs, cs2)

  Find solution looking through the different waves.

- subroutine solvers_hydro::cal_gamma (y)

  Computes the value of the specific heats ratio γ according to the mass fraction y.

9.33.1 Detailed Description

Contains subroutines solver_relax(), solver_relaxation(), solver_exact(), solver_cc(), solver_acoustic(), solver_eos(), solver_g(), solver_iso(), starpu(), guessp(), prefun(), sample(), starpu_iso(), guessp_iso(), prefun_iso(), sample_iso() and cal_gamma().

Definition in file solvers.f90.

9.34 init/init.f90 File Reference

Contains subroutines init() and dimensions().

Functions/Subroutines

- subroutine init

  Initialises the simulation.

- subroutine dimensions

  Defines the physical units and constants in the code.

9.34.1 Detailed Description

Contains subroutines init() and dimensions(). This is essentially a dummy file. The init files specific to each run are usually linked from their own directories.

Definition in file init.f90.

9.34.2 Function Documentation

9.34.2.1 subroutine dimensions ()

Defines the physical units and constants in the code.

Definition at line 159 of file init.f90.

References unites::a_R, unites::an, unites::c, unites::c2, unites::centimetre, unites::cm2, unites::cm3, unites::degre, unites::dyne, unites::erg, unites::eV, unites::G, gammas::g1, gammas::g2, gammas::g3, gammas::g4, gammas::g5, gammas::g6, gammas::g7, gammas::g8, gammas::gamma, unites::Gans, unites::gramme, unites::H0, unites::hplanck, unites::joule, unites::Kelvin, unites::kg, unites::kms, unites::kpc, unites::Lsol, unites::Mans, unites::metre, unites::Mpc, unites::ms, unites::Msol, unites::Pascal, unites::pc, unites::Rsol, unites::seconde, and unites::uma.

9.34.2.2 subroutine init ()

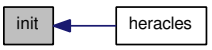
Initialises the simulation. The namelist initial conditions are read in. The grid coordinates are set, the variables inside the conservative arrays and the boundary conditions are initialised.

Definition at line 21 of file init.f90.

References unites::centimetre, var::E, gammas::gamma, cl::icl, divers::mu, parameters::nx, parameters::nx_glob, var::rho, var::rhou, divers::temps, geom::x, geom::x_glob, and para::x_start.

Referenced by heracles().

Here is the caller graph for this function:



9.35 init/init_out.f90 File Reference

Contains subroutine init_out().

Functions/Subroutines

- subroutine init_out

  Initialises the output parameters of the simulation.

9.35.1 Detailed Description

Contains subroutine init_out(). This is essentially a dummy file. The init_out files specific to each run are usually linked from their own directories.

Definition in file init_out.f90.

9.35.2 Function Documentation

9.35.2.1 subroutine init_out ()

Initialises the output parameters of the simulation.

Definition at line 18 of file init_out.f90.

References divers::iout, para::mype, divers::nout, divers::temps, divers::tend, and divers::tout.

Referenced by heracles().

Here is the caller graph for this function:

9.36 main/aff.f90 File Reference

Contains subroutines aff_new(), make_name(), mk_dir_out(), convtoasc() and sortie_ecran().

Functions/Subroutines

- subroutine aff_new (k)

  This routine writes binary data cubes which will be used for data analysis and for simulation restart.

- subroutine make_name (liste, mype, iout)

  Creates a name for the ouput binary file.

- subroutine mk_dir_out (i1, i2)

  Creates a new directory for outputting simulation data.

- subroutine convtoasc (number, sstring, ndigits)

  Converts an integer smaller than 999999 to a ndigits characters string.

- subroutine sortie_ecran

  Writes regular outputs to the terminal.

- subroutine print_configuration (format_sortie)

  Prints the initial configuration of the simulation: size of grid, whether gravity, MHD, rad transfer or MPI are being used, etc.

9.36.1 Detailed Description

Contains subroutines aff_new(), make_name(), mk_dir_out(), convtoasc() and sortie_ecran(). This file contains all the routines used for writing output files and printing outputs to the terminal.

Definition in file aff.f90.

9.36.2 Function Documentation

9.36.2.1 subroutine aff_new (integer *k*)

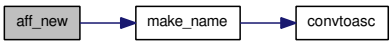
This routine writes binary data cubes which will be used for data analysis and for simulation restart.

Definition at line 20 of file aff.f90.

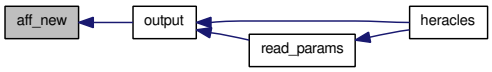
References unites::an, var::B, divers::DIFFU, divers::dt, divers::dtold, var::E, diffusion::Er_dif, varray::Eray, varray::Fray, gammas::gamma, geom::geometrie, divers::iout, make_name(), divers::mu, para::mype, para::ncpu, parameters::ndim, geom::nshift_gr, parameters::nx, parameters::nx_glob, divers::RAY, var::rho, var::rhou, geom::shift_gr, divers::temps, divers::tout, unites_sortie::u_dens, unites_sortie::u_Fr, unites_sortie::u_L, unites_sortie::u_Mom, unites_sortie::u_T, geom::x, para::x_end, geom::x_glob, and para::x_start.

Referenced by output().

Here is the call graph for this function:



Here is the caller graph for this function:



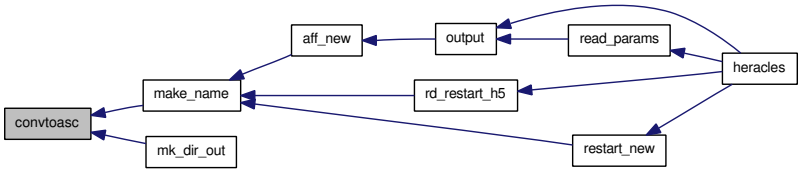
9.36.2.2 subroutine convtoasc (integer *number*, character(len=ndigits) *sstring*, integer *ndigits*)

Converts an integer smaller than 999999 to a ndigits characters string.

Definition at line 213 of file aff.f90.

Referenced by make_name(), and mk_dir_out().

Here is the caller graph for this function:



9.36.2.3 subroutine make_name (character(len=23) *liste*, integer,intent(in) *mype*, integer,intent(in) *iout*)

Creates a name for the ouput binary file.

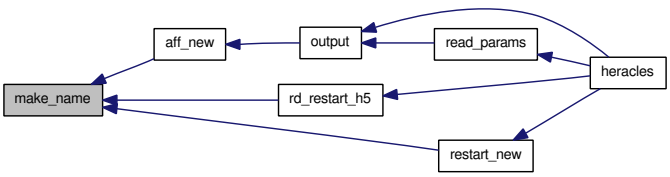
Definition at line 134 of file aff.f90.

References convtoasc().

Referenced by aff_new(), rd_restart_h5(), and restart_new().

Here is the call graph for this function:

Here is the caller graph for this function:



9.36.2.4 subroutine mk_dir_out (integer *i1*, integer *i2*)

Creates a new directory for outputting simulation data.

Definition at line 167 of file aff.f90.

References convtoasc(), and para::mype.

Here is the call graph for this function:

9.36.2.5 subroutine print_configuration (character(len=3) *format_sortie*)

Prints the initial configuration of the simulation: size of grid, whether gravity, MHD, rad transfer or MPI are being used, etc. ..

Definition at line 334 of file aff.f90.

References geom::Cartesien, unites::centimetre, divers::CONDUC, geom::Cylindrique, unites::degre, divers::DIFFU, divers::dt_max, geom::geom_dir, geom::geometrie, unites::gramme, divers::GRAV, divers::HYDRO, geom::Lbox, para::ncpu, parameters::ndim, divers::nout, divers::nsx, divers::nsy, divers::nsz, parameters::nx_glob, divers_ray::rad_trans_model, divers::RAY, divers::reprise, unites::seconde, parameters::slope_type, divers_ray::slope_type_ray, geom::Spherique, divers::tend, divers_ray::varrel_ray, and geom::x_glob.

Referenced by heracles().

Here is the caller graph for this function:

9.36.2.6 subroutine sortie_ecran ()

Writes regular outputs to the terminal.

Definition at line 243 of file aff.f90.

References unites::centimetre, unites::cm3, var::E, unites::erg, gammas::gamma, unites::gramme, parameters::nx, var::rho, var::rhou, unites::seconde, parameters::slope_type, and geom::x.

Referenced by heracles().

Here is the caller graph for this function:

9.37 main/allocate_array.f90 File Reference

Contains subroutine allocate_array().

Functions/Subroutines

- subroutine allocate_array

  Allocates all the main hydro and MHD variable arrays.

9.37.1 Detailed Description

Contains subroutine allocate_array().

Definition in file allocate_array.f90.

9.37.2 Function Documentation

9.37.2.1 subroutine allocate_array ()

Allocates all the main hydro and MHD variable arrays.

Definition at line 14 of file allocate_array.f90.

References var::B, cl::B_l, varold::Bold, var::E, cl::E_l, varold::Eold, var::Fx, cl::fx_l, varold::Fxold, parameters::N_vit, parameters::Nbuf, parameters::ndim, parameters::nFx, geom::nshift_gr, divers::nsupp, parameters::nx, parameters::nx_glob, parameters::nx_glob_max, parameters::nx_max, parameters::nxmax, parameters::nxmin, cl::P_l, var::rho, cl::rho_l, varold::rhoold, var::rhou, cl::rhou_l, varold::rhouold, geom::shift_gr, var::supp1, var::supp2, var::supp3, var::supp4, var::supp5, divers::suppdim, cl::T_l, cl::u_l, geom::x, and geom::x_glob.

Referenced by heracles().

Here is the caller graph for this function:

9.38 main/check_flags.f90 File Reference

Contains subroutine check_flags().

Functions/Subroutines

- subroutine check_flags (format_sortie)

  Performs a safety check on all the major flags (GRAV, RAY, MHD, etc.

9.38.1 Detailed Description

Contains subroutine check_flags().

Definition in file check_flags.f90.

9.38.2 Function Documentation

9.38.2.1 subroutine check_flags (character(len=∗) *format_sortie*)

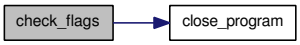
Performs a safety check on all the major flags (GRAV, RAY, MHD, etc. ..)

Definition at line 15 of file check_flags.f90.

References geom::Cartesien, close_program(), divers::CONDUC, geom::Cylindrique, divers::DIFFU, geom::geom_dir, geom::geometrie, divers::GRAV, divers::HYDRO, cl::icl_glob, cl_ray::icl_ray_glob, para::mype, parameters::N_vit, parameters::Nbuf, parameters::ndim, parameters::nx_glob, divers::RAY, and geom::Spherique.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.39 main/check_size.f90 File Reference

Contains subroutine check_size().

Functions/Subroutines

- subroutine check_size (imin, imax, jmin, jmax, kmin, kmax)

  Performs a safety check on the sub-arrays sizes.

9.39.1 Detailed Description

Contains subroutine check_size().

Definition in file check_size.f90.

9.39.2 Function Documentation

9.39.2.1 subroutine check_size (integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,intent(in) *kmax*)

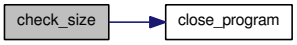
Performs a safety check on the sub-arrays sizes.

Definition at line 14 of file check_size.f90.

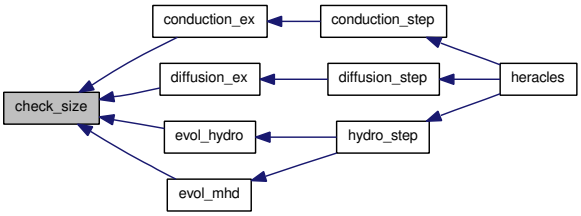
References close_program(), para::mype, and parameters::nx.

Referenced by conduction_ex(), diffusion_ex(), evol_hydro(), and evol_mhd().

Here is the call graph for this function:



Here is the caller graph for this function:



9.40 main/cl_ana.f90 File Reference

Contains subroutines cl_ana() and cl_ana_gen().

Functions/Subroutines

- subroutine cl_ana (iface)

  Dummy analytical boundary conditions.

- subroutine cl_ana_gen (iface, Tab, Tab_l, n)

  Dummy generic analytical boundary conditions.

9.40.1 Detailed Description

Contains subroutines cl_ana() and cl_ana_gen().

Definition in file cl_ana.f90.

9.40.2 Function Documentation

9.40.2.1 subroutine cl_ana (integer *iface*)

Dummy analytical boundary conditions.

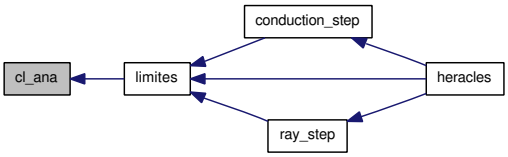
Definition at line 14 of file cl_ana.f90.

References close_program(), and para::mype.

Referenced by limites().

Here is the call graph for this function:

Here is the caller graph for this function:



9.40.2.2 subroutine cl_ana_gen (integer *iface*, real *Tab*, real *Tab_l*, integer *n*)

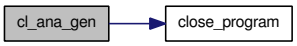
Dummy generic analytical boundary conditions.

Definition at line 36 of file cl_ana.f90.

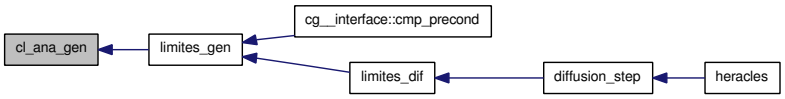
References close_program(), and para::mype.

Referenced by limites_gen().

Here is the call graph for this function:



Here is the caller graph for this function:



9.41 main/cpu.f90 File Reference

Contains subroutine cpu_().

Functions/Subroutines

- subroutine cpu_

  Performs the domain decomposition among the different cpus.

9.41.1 Detailed Description

Contains subroutine cpu_().

Definition in file cpu.f90.

9.41.2 Function Documentation

9.41.2.1 subroutine cpu_ ()

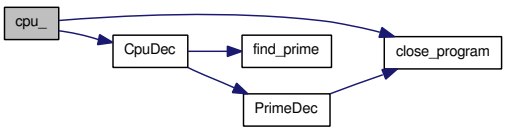
Performs the domain decomposition among the different cpus.

Definition at line 15 of file cpu.f90.

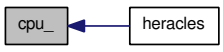
References close_program(), CpuDec(), para::mype, para::ncpu, para::ncpu_x, para::ncpu_y, para::ncpu_z, parameters::ndim, and parameters::nx_glob.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:



9.42 main/film.f90 File Reference

Contains module mfilm and subroutines film_() and init_film().

Modules

- module mfilm

  Contains variables used to create movies.

Functions/Subroutines

- subroutine film_

  Writes many density of velocity slices to create a movie.

- subroutine init_film

  Initialisation of variables for creation of movies.

Variables

- integer mfilm::ifilm

  Current number of film output.

- real mfilm::dt_film

  Timestep between film outputs.

- real mfilm::t_film

  t_film

- real mfilm::t_start

  t_start

- real mfilm::t_end

  t_end

- logical mfilm::FILM

  Creates movie outputs if .true.

- integer mfilm::nl1

  nl1

- integer mfilm::nl2

  nl2

- integer mfilm::nl3

  nl3

9.42.1 Detailed Description

Contains module mfilm and subroutines film_() and init_film(). This file contains the routines and variables used to create movies.

Definition in file film.f90.

9.42.2 Function Documentation

9.42.2.1 subroutine film_ ()

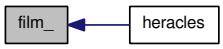
Writes many density of velocity slices to create a movie.

Definition at line 39 of file film.f90.

References unites::cm3, mfilm::dt_film, unites::gramme, mfilm::ifilm, para::mype, para::ncpu, parameters::nx, parameters::nx_cpu, parameters::nx_glob, mfilm::t_film, divers::temps, divers::verbose, and para::x_start_cpu.

Referenced by heracles().

Here is the caller graph for this function:



9.42.2.2 subroutine init_film ()

Initialisation of variables for creation of movies.

Definition at line 217 of file film.f90.

References unites::an, mfilm::dt_film, parameters::ndim, mfilm::nl1, mfilm::nl2, mfilm::nl3, mfilm::t_end, mfilm::t_film, mfilm::t_start, divers::temps, and divers::verbose.

Referenced by heracles().

Here is the caller graph for this function:

9.43 main/geom.f90 File Reference

Contains subroutines def_geom() and cal_geom().

Functions/Subroutines

- subroutine def_geom

  Definition of the grid geometry by computing the cell interface coordinates.

- subroutine cal_geom (nnx, nny, nnz)

  Computes the cell sizes, face areas and volumes.

9.43.1 Detailed Description

Contains subroutines def_geom() and cal_geom(). This file contains the routines which define the geometry of the grid (cartesian, spherical or cylindrical). It computes the cell coordinates, sizes, ares and volumes.

Definition in file geom.f90.

9.43.2 Function Documentation

9.43.2.1 subroutine cal_geom (integer *nnx*, integer *nny*, integer *nnz*)

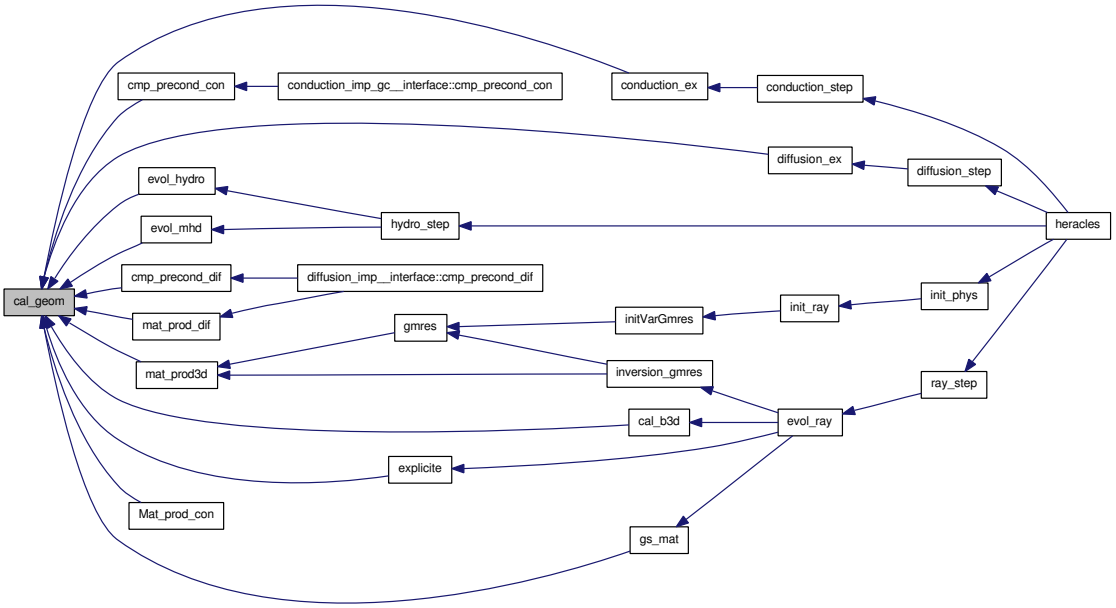
Computes the cell sizes, face areas and volumes.

Definition at line 71 of file geom.f90.

References geom::Cartesien, geom::Cylindrique, geom::ds, geom::dv, geom::dx, geom::dxc, geom::geometrie, parameters::ndim, geom::Spherique, geom::xcloc, and var_loc::xloc.

Referenced by cal_b3d(), cmp_precond_con(), cmp_precond_dif(), conduction_ex(), diffusion_ex(), evol_hydro(), evol_mhd(), explicite(), gs_mat(), mat_prod3d(), Mat_prod_con(), and mat_prod_dif().

Here is the caller graph for this function:



9.43.2.2 subroutine def_geom ()

Definition of the grid geometry by computing the cell interface coordinates.

Definition at line 20 of file geom.f90.

References cl::icl, parameters::ndim, parameters::nx, and geom::x.

Referenced by heracles().

Here is the caller graph for this function:

9.44 main/init_phys.f90 File Reference

Contains subroutine init_phys().

Functions/Subroutines

- subroutine init_phys

  Initialises the physical modules required by the simulation.

9.44.1 Detailed Description

Contains subroutine init_phys().

Definition in file init_phys.f90.

9.44.2 Function Documentation

9.44.2.1 subroutine init_phys ()

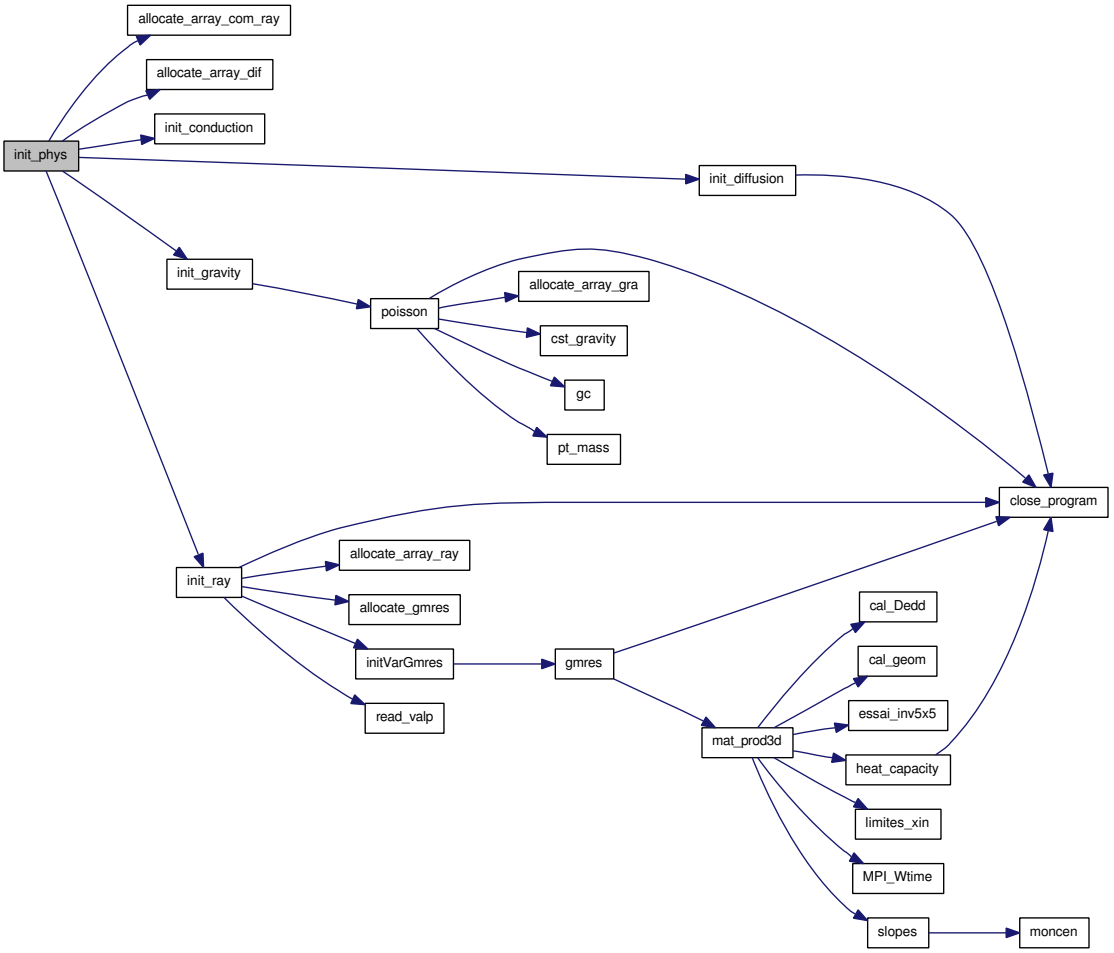
Initialises the physical modules required by the simulation.

Definition at line 15 of file init_phys.f90.

References allocate_array_com_ray(), allocate_array_dif(), init_conduction(), init_diffusion(), init_gravity(), init_ray(), and para::mype.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.45 main/limites.f90 File Reference

Contains subroutine limites().

Functions/Subroutines

- subroutine limites

  This routine fills the ghost cells at the domain boundaries with the appropriate values according to the type of boudary condition.

9.45.1 Detailed Description

Contains subroutine limites().

Definition in file limites.f90.

9.45.2 Function Documentation

9.45.2.1 subroutine limites ()

This routine fills the ghost cells at the domain boundaries with the appropriate values according to the type of boudary condition. Type of boundaries:

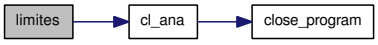
- 1 : free-flow
- 2 : periodic
- 3 : reflexive
- 4 : imposed values
- 5 : analytical

Definition at line 23 of file limites.f90.

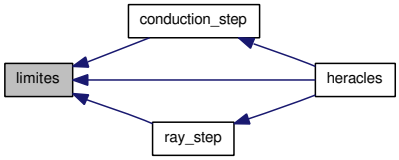
References var::B, cl::B_l, cl_ana(), var::E, cl::E_l, cl::fx_l, parameters::N_vit, parameters::Nbuf, parameters::ndim, parameters::nx, parameters::nxmax, parameters::nxmin, var::rho, cl::rho_l, var::rhou, cl::rhou_l, and divers::verbose.

Referenced by conduction_step(), heracles(), and ray_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.46 main/limites_generique.f90 File Reference

Contains subroutine limites_gen().

Functions/Subroutines

- subroutine limites_gen (Tab, icl, Tab_l, n)

  This routine is a generic form of the limites() subroutine.

9.46.1 Detailed Description

Contains subroutine limites_gen().

Definition in file limites_generique.f90.

9.46.2 Function Documentation

9.46.2.1 subroutine limites_gen (Real,dimension(nxmin(1) *Tab*, integer,dimension(1:6) *icl*, real,dimension(n,1:2∗ndim) *Tab_l*, integer,dimension(nxmin(1) *n*)

This routine is a generic form of the limites() subroutine. Type of boundaries:

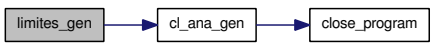
- 1 : free-flow
- 2 : periodic
- 3 : reflexive
- 4 : imposed values
- 5 : analytical

Definition at line 22 of file limites_generique.f90.

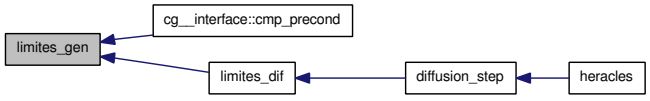
References cl_ana_gen(), parameters::Nbuf, parameters::ndim, parameters::nx, and divers::verbose.

Referenced by cg__interface::cmp_precond(), and limites_dif().

Here is the call graph for this function:



Here is the caller graph for this function:



9.47 main/main.f90 File Reference

Contains main program HERACLES and subroutines compute_cputime(), output() and close_program() and functions Arret(), MPI_Wtime(), tremain().

Functions/Subroutines

- program heracles

  Main program.

- real MPI_Wtime (ierr)

  Computes the MPI World time.

- integer tremain ()

  ?????

- integer Arret ()

  This routine is used to cleanly exit the program.

- subroutine compute_cputime (mype)

  This routine calculates the fractions of the computation time spent in the different parts of the code.

- subroutine output (kk, output_format, t_io, mype)

  This routine calls the relevant output routines to write binary or HDF5 files.

- subroutine close_program (mype, message)

  This routine closes the HERACLES program cleanly taking into account whether it is using MPI or not.

9.47.1 Detailed Description

Contains main program HERACLES and subroutines compute_cputime(), output() and close_program() and functions Arret(), MPI_Wtime(), tremain(). This file contains the main program and the routines used to allocate and deallocate the variable arrays, and open and close the log files.

Definition in file main.f90.

9.47.2 Function Documentation

9.47.2.1 integer Arret ()

This routine is used to cleanly exit the program.

Definition at line 406 of file main.f90.

References para::mype, divers::tcpu_last, and divers::tcpu_max.

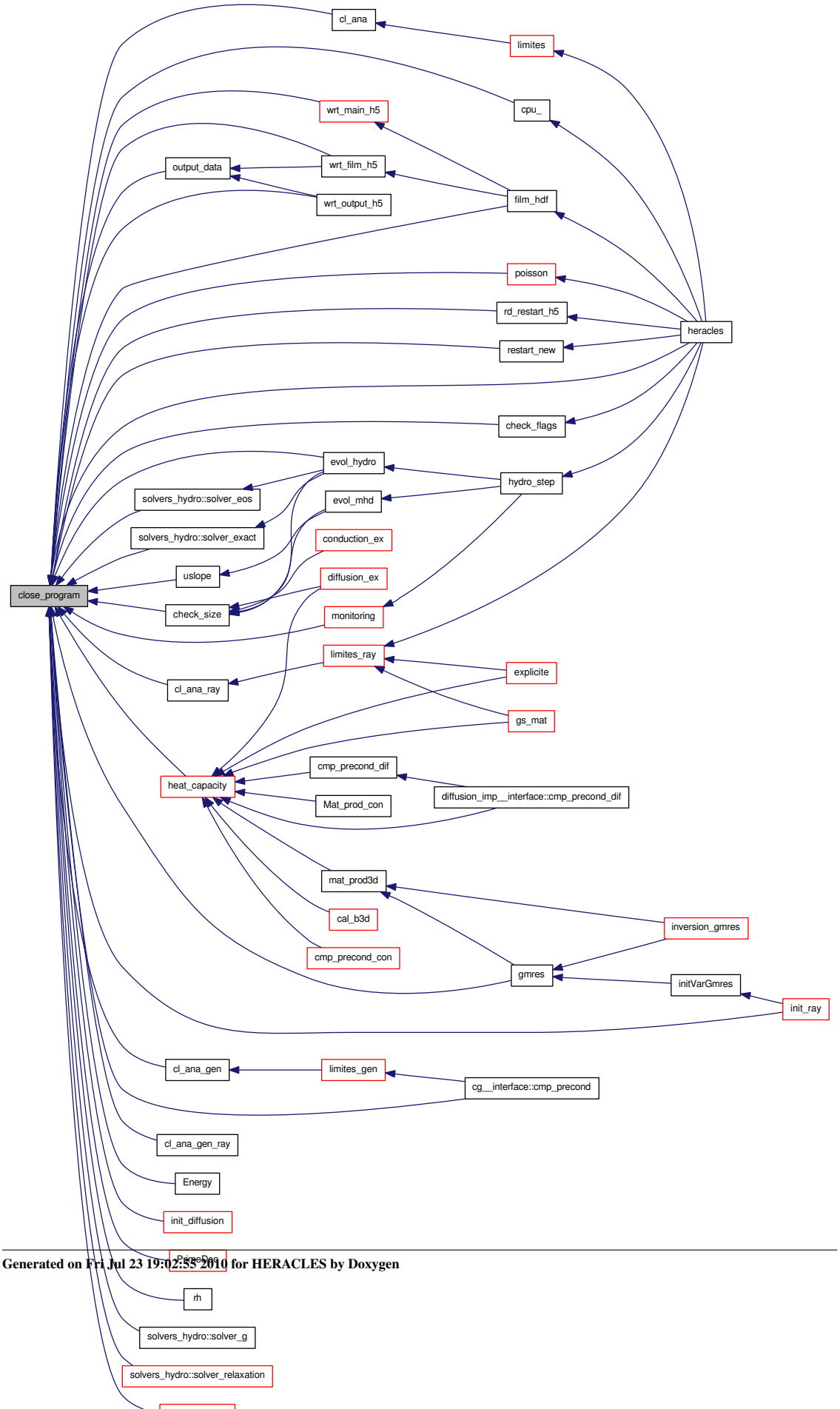
9.47.2.2 subroutine close_program (integer *mype*, character (len=∗) *message*)

This routine closes the HERACLES program cleanly taking into account whether it is using MPI or not. It is also used to close any opened monitoring files if the program exits in the middle of a simulation. This routine should replace every call to 'stop' in the entire source.

Definition at line 597 of file main.f90.

Referenced by check_flags(), check_size(), cl_ana(), cl_ana_gen(), cl_ana_gen_ray(), cl_ana_ray(), cg_-_interface::cmp_precond(), cpu_(), Energy(), evol_hydro(), film_hdf(), gmres(), heat_capacity(), heracles(), init_diffusion(), init_ray(), monitoring(), output_data(), poisson(), PrimeDec(), rd_restart_h5(), restart_new(), rh(), solvers_hydro::solver_eos(), solvers_hydro::solver_exact(), solvers_hydro::solver_g(), solvers_hydro::solver_relaxation(), uslope(), wrt_film_h5(), wrt_main_h5(), wrt_output_h5(), and wrt_-restart_h5().

Here is the caller graph for this function:



9.47.2.3 subroutine compute_cputime (integer *mype*)

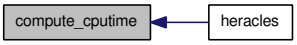
This routine calculates the fractions of the computation time spent in the different parts of the code.

Definition at line 475 of file main.f90.

References cputime::temps1, cputime::temps_comm, cputime::temps_conduc, cputime::temps_dif, cputime::temps_hydro, cputime::temps_io, cputime::temps_ray, and cputime::temps_ref.

Referenced by heracles().

Here is the caller graph for this function:



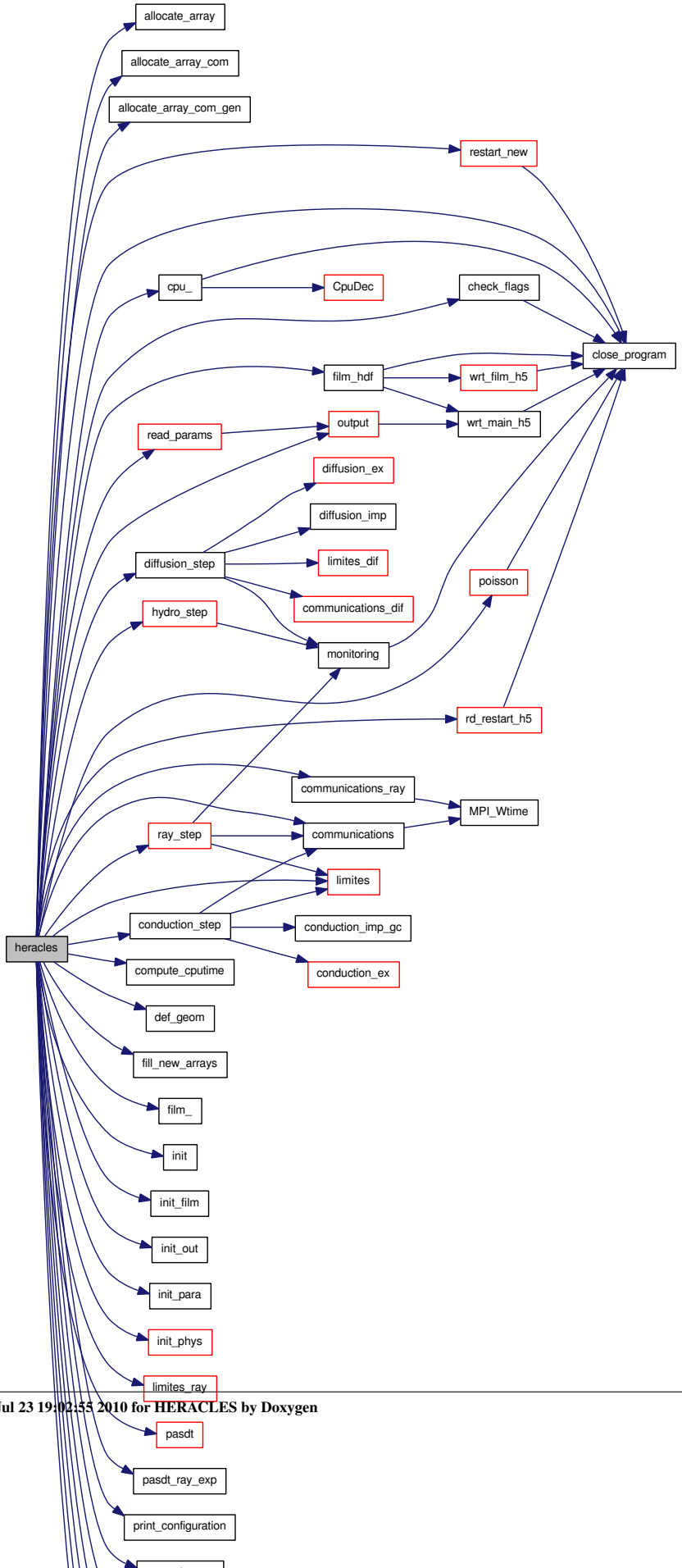
9.47.2.4 program heracles ()

Main program. 3D MPI Eulerian second-order Godunov hydrodynamic simulation code in cartesian, spherical and cylindrical coordinates, including MHD, radiative transfer and gravity.

Definition at line 41 of file main.f90.

References allocate_array(), allocate_array_com(), allocate_array_com_gen(), unites::an, var::B, varold::Bold, check_flags(), close_program(), communications(), communications_ray(), compute_cputime(), divers::CONDUC, conduction_step(), cpu_(), def_geom(), divers::DIFFU, diffusion_step(), divers::dt, divers::dt_max, divers_ray::dt_ray, divers_ray::dt_ray_exp, var::E, varold::Eold, varray::Eray, varray::Erayold, fill_new_arrays(), mfilm::FILM, film_(), film_hdf(), varray::Fray, varray::Frayold, divers::GRAV, hydro_step(), init(), init_film(), init_out(), init_para(), init_phys(), divers::iout, limites(), limites_ray(), para::mype, para::ncpu, parameters::ndim, divers::Nimp, output(), pasdt(), pasdt_ray_exp(), poisson(), print_configuration(), divers::RAY, ray_step(), rd_restart_h5(), read_params(), divers::reprise, restart_new(), var::rho, varold::rhoold, var::rhou, varold::rhouold, unites::seconde, set_units_out(), sortie_ecran(), divers::tcpu_ini, divers::temps, cputime::temps1, cputime::temps_comm, cputime::temps_conduc, cputime::temps_dif, cputime::temps_hydro, cputime::temps_io, cputime::temps_ray, divers::tend, var::Tgaz, varold::Tgazold, divers::tout, update_gravity(), user_init(), user_output(), user_step(), and divers::verbose.

Here is the call graph for this function:



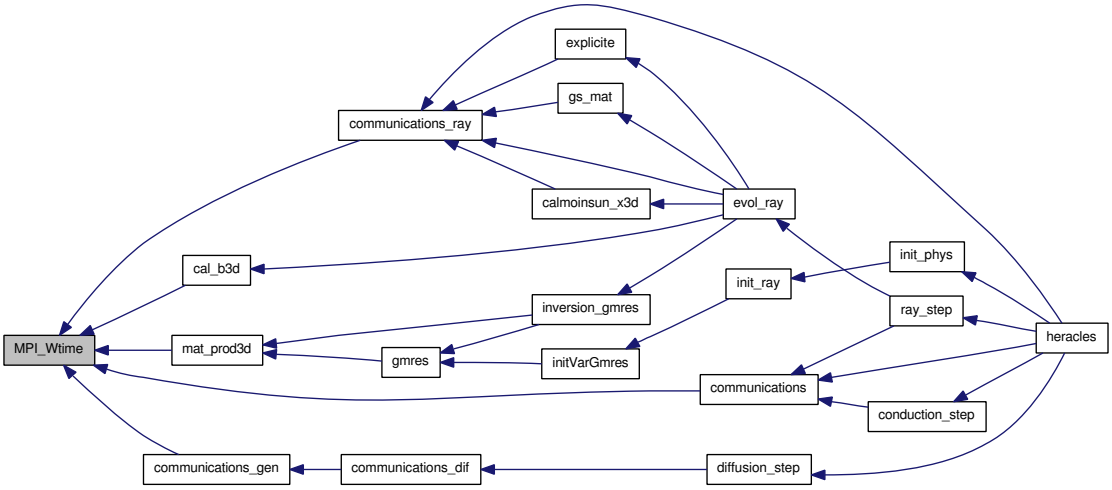
9.47.2.5 real MPI_Wtime (integer *ierr*)

Computes the MPI World time.

Definition at line 350 of file main.f90.

Referenced by cal_b3d(), communications(), communications_gen(), communications_ray(), and mat_-prod3d().

Here is the caller graph for this function:



9.47.2.6 subroutine output (integer *kk*, character(len=3) *output_format*, real *t_io*, integer *mype*)

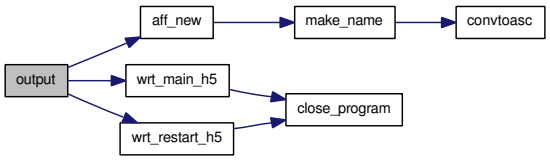
This routine calls the relevant output routines to write binary or HDF5 files.

Definition at line 553 of file main.f90.

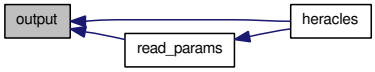
References aff_new(), wrt_main_h5(), and wrt_restart_h5().

Referenced by heracles(), and read_params().

Here is the call graph for this function:



Here is the caller graph for this function:



9.47.2.7 integer tremain ()

?????

Definition at line 388 of file main.f90.

9.48 main/modules.f90 File Reference

Contains modules parameters, var_loc, geom, var, varold, etat_gds, unites, unites_sortie, divers, param_ini, cl, gammas, cellules, const and cputime.

Modules

- module parameters

  Holds all the general simulation parameters.

- module var_loc

  Contains the local variables.

- module geom

  Contains all the grid variables and coordinates.

- module var

  Contains all the main variable arrays.

- module varold

  Contains all the old (non-updated) variable arrays.

- module Etat_GDS

  Contains the interface values arrays.

- module unites

  This routine calls the relevant output routines to write binary or HDF5 files.

- module unites_sortie

  Contains the units for output file dumping.

- module divers

  Contains various variables.

- module param_ini

  Contains variables used for initialisation of the simulation.

- module cl

  Contains the variables used to define the boundary conditions.

- module gammas

  Contains the variables determining the heat capacity ratio and thermodynamic relations.

- module cellules

  Contains ?

- module const

  Contains useful constants.

- module cputime

  Contains variables used to compute the cpu time spent is each part of the code.

Variables

- integer, dimension(:,:), allocatable parameters::nx_cpu

  Number of cells in the x,y,z directions per cpu.

- integer, dimension(3) parameters::nx_glob

  Global simulation dimensions.

- integer, dimension(3) parameters::nx

  Local simulation dimensions.

- integer, dimension(3) parameters::nxmin

  Lower limit for arrays.

- integer, dimension(3) parameters::nxmax

  Upper limit for arrays.

- integer parameters::ndim

  Number of dimensions.

- integer parameters::nx_max

  Highest nx.

- integer parameters::nx_glob_max

  Global highest nx.

- integer parameters::Nbuf

  Number of buffer/ghost cells.

- integer parameters::N_vit

  Number of velocity components.

- integer parameters::nvar

  Number of hydro/MHD variables.

- integer parameters::nFx

  Number of passive scalars.

- integer parameters::nvar_ray

  Total number of radiative variables.

- integer parameters::slope_type

  Type of slope limiter for Hydro and MHD.

- real, parameter parameters::Pi = 3.1415926535898

π

- real, parameter parameters::quatre_Pi = 4.*pi
  4π
- real, parameter parameters::smallc = 1.d-30
  Dimensioned small constant.
- real, parameter parameters::smallr = 1.d-40
  Dimensioned small real number.
- character(len=10) parameters::riemann
  Type of Riemann solver.
- character(len=10) parameters::riemann2d
  Type of Riemann solver 2.
- real, dimension(:,:,:,:), allocatable var_loc::rhouloc
  Local momentum array.
- real, dimension(:,:,:,:), allocatable var_loc::uloc
  Local velocity array.
- real, dimension(:,:,:,:), allocatable var_loc::Fxloc
  Local passive scalar array.
- real, dimension(:,:,:), allocatable var_loc::rholoc
  Local density array.
- real, dimension(:,:,:), allocatable var_loc::Ploc
  Local pressure array.
- real, dimension(:,:,:), allocatable var_loc::Eloc
  Local total gas energy array.
- real, dimension(:,:,:), allocatable var_loc::Tloc
  Local temperature array.
- real, dimension(:,:,:), allocatable var_loc::Einloc
  Local internal gas energy array.
- real, dimension(:,:,:), allocatable var_loc::philoc
  Local gravitational potential array.
- real, dimension(:,:), allocatable var_loc::xloc
  Local coordinate array.
- integer, dimension(3) geom::nshift_gr
  nshift

- real, dimension(:,:,:,:), allocatable geom::ds
  Surface element.
- real, dimension(:,:,:), allocatable geom::dv
  Volume element.
- real, dimension(:,:), allocatable geom::x
  Local cell interface coordinate array.
- real, dimension(:,:), allocatable geom::x_glob
  Global cell interface coordinate array.
- real, dimension(:,:), allocatable geom::dx
  Line element.
- real, dimension(:,:), allocatable geom::dxc
  Cell centred line element.
- real, dimension(:,:), allocatable geom::xcloc
  Local cell centre coordinate array.
- real, dimension(3) geom::shift_gr
  shiftgr
- real, dimension(3) geom::Lbox
  Size of computational box.
- real, dimension(3) geom::box_min
  Negative coordinate offset.
- real, dimension(3) geom::box_max
  Positive coordinate offset.
- character(len=5), dimension(3) geom::geom_dir
  Geometry directions.
- character(len=5) geom::geometrie
  Grid geometry.
- logical geom::Cartesien
  Cartesian grid if .true.
- logical geom::Cylindrique
  Cylindrical grid if .true.
- logical geom::Spherique
  Spherical grid if .true.
- real, dimension(:,:,:,:), allocatable var::rhou
  Gas momentum array $\rho(u_x, u_y, u_z)$.

- real, dimension(:,:,:,:), allocatable var::B
  Magnetic field.
- real, dimension(:,:,:,:), allocatable var::Fx
  Passive scalar array.
- real, dimension(:,:,:,:), allocatable var::supp1
  Additional array 1.
- real, dimension(:,:,:,:), allocatable var::supp2
  Additional array 2.
- real, dimension(:,:,:,:), allocatable var::supp3
  Additional array 3.
- real, dimension(:,:,:,:), allocatable var::supp4
  Additional array 4.
- real, dimension(:,:,:,:), allocatable var::supp5
  Additional array 5.
- real, dimension(:,:,:), allocatable var::rho
  Gas density array ρ.
- real, dimension(:,:,:), allocatable var::E
  Total gas energy $E = \frac{1}{2}\rho u^2 + \frac{P}{\gamma - 1}$.
- real, dimension(:,:,:), allocatable var::Tgaz
  Gas temperature array.
- real, dimension(:,:,:,:), allocatable varold::rhouold
  Old gas momentum.
- real, dimension(:,:,:,:), allocatable varold::Bold
  Old magnetic field.
- real, dimension(:,:,:,:), allocatable varold::Fxold
  Old passive scalar.
- real, dimension(:,:,:), allocatable varold::rhoold
  Old gas density.
- real, dimension(:,:,:), allocatable varold::Eold
  Old total gas energy.
- real, dimension(:,:,:), allocatable varold::Tgazold
  Old gas temperature.
- real, dimension(:,:,:,:,:), allocatable Etat_GDS::uS

Value of the gas velocity at the cell interface.

- real, dimension(:,:,:,:), allocatable Etat_GDS::FxS
  Value of the passive scalar at the cell interface.
- real, dimension(:,:,:,:), allocatable Etat_GDS::rhoS
  Value of the gas momentum at the cell interface.
- real, dimension(:,:,:,:), allocatable Etat_GDS::pS
  Value of the gas pressure at the cell interface.
- real, dimension(:,:,:,:), allocatable Etat_GDS::eS
  Value of the total gas energy at the cell interface.
- real unites::gramme
  Gram unit.
- real unites::centimetre
  Centimetre unit.
- real unites::seconde
  Second unit.
- real unites::dyne
  Dyne unit.
- real unites::erg
  Erg unit.
- real unites::degre
  Temperature unit.
- real unites::unitm
  Mass unit.
- real unites::unitl
  Length unit.
- real unites::unitt
  Time unit.
- real unites::unitf
  Force unit.
- real unites::unite
  Energy unit.
- real unites::unitd
  Temperature unit.

- real unites::unitmom
  Momentum unit.
- real unites::cm2
  cm^2
- real unites::cm3
  cm^3
- real unites::ms
  Millisecond.
- real unites::kms
  Kilometer.
- real unites::an
  Year.
- real unites::Mans
  Million years.
- real unites::Gans
  Billion years.
- real unites::Msol
  Solar mass $M_\odot$.
- real unites::Rsol
  Solar radius $R_\odot$.
- real unites::Lsol
  Solar luminosity $L_\odot$.
- real unites::pc
  Parsec.
- real unites::kpc
  Kiloparsec.
- real unites::Mpc
  Megaparsec.
- real unites::Kelvin
  Kelvin.
- real unites::c
  Speed of light c.
- real unites::c2
  Square of the speed of light c^2.

- real unites::eV
  Electron volt eV.
- real unites::G
  Gravitational constant G.
- real unites::hplanck
  Planck constant h.
- real unites::kB
  Boltzmann constant k_B.
- real unites::uma
  Atomic mass unit u.
- real unites::H0
  Hubble's constant H_0.
- real unites::a_R
  Stefan-Boltzmann constant a_R.
- real unites::Pascal
  Pascal.
- real unites::bar
  Bar.
- real unites::kg
  Kilogram.
- real unites::metre
  Metre.
- real unites::m2
  m^2
- real unites::m3
  m^3
- real unites::joule
  Joule.
- real unites::ns
  Nanosecond.
- real unites::micron
  Micron.
- real unites::Gauss

Gauss.

- real unites::mu0

  Reduced atomic mass.

- real unites_sortie::u_M

  Output mass unit.

- real unites_sortie::u_L

  Output length unit.

- real unites_sortie::u_T

  Output time unit.

- real unites_sortie::u_E

  Output energy unit.

- real unites_sortie::u_Vol

  Output volume unit.

- real unites_sortie::u_dens

  Output density unit.

- real unites_sortie::u_evol

  Output volumic energy unit.

- real unites_sortie::u_Fr

  Output force unit.

- real unites_sortie::u_Vit

  Output velocity unit.

- real unites_sortie::u_Mom

  Output momentum unit.

- integer, dimension(20) divers::fourdim

  f

- integer, dimension(5) divers::suppdim

  f

- integer divers::Nimp

  Write 000 output file every Nimp timesteps.

- integer divers::i_restart

  File number to restart from.

- integer divers::nsx

  x domain subdivision factor

- integer divers::nsy
  y domain subdivision factor
- integer divers::nsz
  z domain subdivision factor
- integer divers::nrepr
  nr
- integer divers::irepr
  ir
- integer divers::nreprd
  nr
- integer divers::ndata
  nd
- integer divers::nsupp
  Number of additional arrays required.
- integer divers::nout
  Total number of outputs.
- integer divers::iout
  iout
- integer divers::noutd
  no
- real, dimension(1000) divers::tout
  Output time.
- real, dimension(1000) divers::trepr
  Restart time.
- real divers::temps
  Time.
- real divers::dt
  Timestep.
- real divers::dtold
  Old timestep.
- real divers::tend
  End time.
- real divers::alpha
  α

- real divers::dt_max
  Maximum timestep.
- real divers::dt_glob
  Global timestep.
- real divers::dt_mhd
  MHD timestep.
- real divers::dt_cfl
  CFL limited timestep.
- real divers::dt_ff
  Gravitational free-fall limited timestep.
- real divers::tcpu_min
  tcpu_min
- real divers::tcpu_max
  tcpu_max
- real divers::tcpu_ini
  tcpu_ini
- real divers::tcpu_last
  tcpu_last en seconde
- real divers::mu
  μ
- character(len=80), dimension(20) divers::datanames
  datanames
- character(len=80), dimension(20) divers::dataunits
  dataunits
- character(len=80), dimension(5) divers::suppnames
  Names of additional arrays.
- character(len=80), dimension(5) divers::suppunits
  Units of additional arrays.
- character(len=80) divers::comments
  comments
- character(len=10) divers::Solver
  Type of Hydro/MHD Riemann solver.
- character(len=5) divers::name_of_job

Name of job.

- character(len=3) divers::sortie

  Output.

- logical divers::reprise

  Perform restart from previous run if .true.

- logical divers::COM

  com

- logical divers::subcycle

  subcycle

- logical divers::limdx

  limdx

- logical divers::limnite

  limnite

- logical divers::limfr

  limfr

- logical divers::REF

  Include cooling if .true.

- logical divers::DETO

  Include detonation if .true.

- logical divers::CONDUC

  Include thermal conduction if .true.

- logical divers::DIFFU

  Include flux limited diffusion if .true.

- logical divers::RAY

  Include radiative transfer if .true.

- logical divers::HYDRO

  Include hydrodynamics if .true.

- logical divers::GRAV

  Include gravity if .true.

- logical divers::HISTOGRAMME

  Computes histograms if .true.

- logical divers::ISOTHERME

  Uses isothermal equation of state if .true.

- logical divers::STIR
  Include stirring if .true.
- logical divers::verbose
  Prints out extra information for debugging if .true.
- logical divers::swip
  Include swip if .true.
- logical divers::rankine
  Include Rankine if .true.
- real param_ini::rho1
  ρ_1
- real param_ini::rho2
  ρ_2
- real param_ini::rho3
  ρ_3
- real param_ini::rho4
  ρ_4
- real param_ini::p1
  p_1
- real param_ini::p2
  p_2
- real param_ini::p3
  p_3
- real param_ini::p4
  p_4
- real param_ini::T1
  T_1
- real param_ini::T2
  T_2
- real param_ini::T3
  T_3
- real param_ini::T4
  T_4
- real param_ini::E1
  E_1

- real param_ini::E2
  E_2
- real param_ini::E3
  E_3
- real param_ini::E4
  E_4
- real param_ini::u1
  u_1
- real param_ini::u2
  u_2
- real param_ini::u3
  u_3
- real param_ini::u4
  u_4
- real param_ini::v1
  v_1
- real param_ini::v2
  v_2
- real param_ini::v3
  v_3
- real param_ini::v4
  v_4
- real param_ini::cs1
  cs_1
- real param_ini::x1
  x_1
- real param_ini::x2
  x_2
- real param_ini::d1
  d_1
- real param_ini::d2
  d_1
- real param_ini::ww

?

- real param_ini::B0

  B_0

- real param_ini::Temperature_isotherme

  Isothermal temperature.

- integer, dimension(6) cl::icl_glob

  Global boundaries types for faces 1 to 6.

- integer, dimension(6) cl::icl

  Local boundaries types for faces 1 to 6.

- integer, dimension(6) cl::icl0

  icl

- real, dimension(:,:), allocatable cl::rhou_l

  Imposed value for the gas momentum inside the ghost cells.

- real, dimension(:,:), allocatable cl::u_l

  Imposed value for the gas velocity inside the ghost cells.

- real, dimension(:), allocatable cl::rho_l

  Imposed value for the gas density inside the ghost cells.

- real, dimension(:), allocatable cl::E_l

  Imposed value for the gas energy inside the ghost cells.

- real, dimension(:), allocatable cl::P_l

  Imposed value for the gas pressure inside the ghost cells.

- real, dimension(:), allocatable cl::T_l

  Imposed value for the gas temperature inside the ghost cells.

- real, dimension(:,:), allocatable cl::fx_l

  Imposed value for the passive scalar inside the ghost cells.

- real, dimension(:,:), allocatable cl::B_l

  Imposed value for the magnetic field inside the ghost cells.

- real gammas::mu_reac

  μ_{reac}

- real gammas::gamma

  Ratio of specific heats γ.

- real gammas::gamma1

  γ_1

- real gammas::gamma2
  γ_1
- real gammas::g1
  $\frac{\gamma-1}{2\gamma}$
- real gammas::g2
  $\frac{\gamma+1}{2\gamma}$
- real gammas::g3
  $\frac{2\gamma}{\gamma-1}$
- real gammas::g4
  $\frac{2}{\gamma-1}$
- real gammas::g5
  $\frac{2}{\gamma+1}$
- real gammas::g6
  $\frac{\gamma-1}{\gamma+1}$
- real gammas::g7
  $\frac{\gamma-1}{2}$
- real gammas::g8
  $\gamma-1$
- real gammas::Cs_iso
  Isothermal sound speed.
- real gammas::Cs_iso2
  Isothermal sound speed 2.
- integer, dimension(:,:), allocatable cellules::Imax
  Imax.
- integer, dimension(:,:), allocatable cellules::Imax2
  Imax2.
- integer, dimension(:), allocatable cellules::fff_t
  fff_t
- integer cellules::N_cell
  N_cell.
- integer cellules::fff
  fff

- integer cellules::ioffset
  ioffset
- integer cellules::ifich
  ifich
- integer cellules::size_offset
  size_offset
- integer cellules::ifilm = 0
  ifilm
- real, dimension(:,:,:), allocatable cellules::Pcell
  Pcell.
- real, dimension(:,:), allocatable cellules::Pmax
  Pmax.
- real, dimension(:,:), allocatable cellules::Pmax2
  Pmax2.
- real cellules::dx_cell
  dx_cell
- real cellules::t_film
  t_film
- real cellules::dt_film
  dt_film
- real, parameter const::bigreal = 1.0e+30
  A big real number.
- real, parameter const::zero = 0.0
  0
- real, parameter const::one = 1.0
  1
- real, parameter const::two = 2.0
  2
- real, parameter const::three = 3.0
  3
- real, parameter const::four = 4.0
  4
- real, parameter const::two3rd = 2.0/3.0
  2/3

- real, parameter const::half = 0.5
  1/2
- real, parameter const::third = 1.0/3.0
  1/3
- real, parameter const::forth = 0.25
  1/4
- real, parameter const::sixth = 1.0/6.0
  1/6
- real cputime::temps1
  Time 1.
- real cputime::temps_ref
  Time spent computing cooling.
- real cputime::temps_conduc
  Time spent computing thermal conduction.
- real cputime::temps_ray
  Time spent computing radiative transfer.
- real cputime::temps_hydro
  Time spent computing hydrodynamics.
- real cputime::temps_io
  Time spent performing outputs.
- real cputime::temps_comm
  Time spent performing communications.
- real cputime::temps_dif
  Time spent computing flux limited diffusion.

9.48.1 Detailed Description

Contains modules parameters, var_loc, geom, var, varold, etat_gds, unites, unites_sortie, divers, param_ini, cl, gammas, cellules, const and cputime. This file contains all the modules which hold all the global variables.

Definition in file modules.f90.

9.49 main/para.f90 File Reference

Contains modules communication and para and subroutines init_para(), allocate_array_com(), communications(), com_1D(), com_2D() and com_3D().

Modules

- module communication

  Contains variables arrays used to communicate ghost cell data across cpus.

- module para

  Contains variables defining the MPI parameters of the simulation.

Functions/Subroutines

- subroutine init_para

  This subroutine initialises the parallelisation of the simualtion.

- subroutine allocate_array_com

  Allocates the arrays used in the communication process.

- subroutine communications (tcomm)

  Perform communications and return cputime used.

Variables

- real, dimension(:,:,:,:), allocatable communication::face1_e

  Face 1 e.

- real, dimension(:,:,:,:), allocatable communication::face2_e

  Face 2 e.

- real, dimension(:,:,:,:), allocatable communication::face3_e

  Face 3 e.

- real, dimension(:,:,:,:), allocatable communication::face4_e

  Face 4 e.

- real, dimension(:,:,:,:), allocatable communication::face5_e

  Face 5 e.

- real, dimension(:,:,:,:), allocatable communication::face6_e

  Face 6 e.

- real, dimension(:,:,:,:), allocatable communication::face1_r

  Face 1 r.

- real, dimension(:,:,:,:), allocatable communication::face2_r
  Face 2 r.
- real, dimension(:,:,:,:), allocatable communication::face3_r
  Face 3 r.
- real, dimension(:,:,:,:), allocatable communication::face4_r
  Face 4 r.
- real, dimension(:,:,:,:), allocatable communication::face5_r
  Face 5 r.
- real, dimension(:,:,:,:), allocatable communication::face6_r
  Face 6 r.
- real, dimension(:,:,:,:), allocatable communication::ar13_e
  Edge 1 e.
- real, dimension(:,:,:,:), allocatable communication::ar14_e
  Edge 2 e.
- real, dimension(:,:,:,:), allocatable communication::ar23_e
  Edge 3 e.
- real, dimension(:,:,:,:), allocatable communication::ar24_e
  Edge 4 e.
- real, dimension(:,:,:,:), allocatable communication::ar15_e
  Edge 5 e.
- real, dimension(:,:,:,:), allocatable communication::ar25_e
  Edge 6 e.
- real, dimension(:,:,:,:), allocatable communication::ar35_e
  Edge 7 e.
- real, dimension(:,:,:,:), allocatable communication::ar45_e
  Edge 8 e.
- real, dimension(:,:,:,:), allocatable communication::ar16_e
  Edge 9 e.
- real, dimension(:,:,:,:), allocatable communication::ar26_e
  Edge 10 e.
- real, dimension(:,:,:,:), allocatable communication::ar36_e
  Edge 11 e.
- real, dimension(:,:,:,:), allocatable communication::ar46_e
  Edge 12 e.

- real, dimension(:,:,:,:), allocatable communication::ar13_r
  Edge 1 r.
- real, dimension(:,:,:,:), allocatable communication::ar14_r
  Edge 2 r.
- real, dimension(:,:,:,:), allocatable communication::ar23_r
  Edge 3 r.
- real, dimension(:,:,:,:), allocatable communication::ar24_r
  Edge 4 r.
- real, dimension(:,:,:,:), allocatable communication::ar15_r
  Edge 5 r.
- real, dimension(:,:,:,:), allocatable communication::ar25_r
  Edge 6 r.
- real, dimension(:,:,:,:), allocatable communication::ar35_r
  Edge 7 r.
- real, dimension(:,:,:,:), allocatable communication::ar45_r
  Edge 8 r.
- real, dimension(:,:,:,:), allocatable communication::ar16_r
  Edge 9 r.
- real, dimension(:,:,:,:), allocatable communication::ar26_r
  Edge 10 r.
- real, dimension(:,:,:,:), allocatable communication::ar36_r
  Edge 11 r.
- real, dimension(:,:,:,:), allocatable communication::ar46_r
  Edge 12 r.
- real, dimension(:,:,:,:), allocatable communication::coin13_e
  Corner 2D 1 e.
- real, dimension(:,:,:,:), allocatable communication::coin14_e
  Corner 2D 2 e.
- real, dimension(:,:,:,:), allocatable communication::coin23_e
  Corner 2D 3 e.
- real, dimension(:,:,:,:), allocatable communication::coin24_e
  Corner 2D 4 e.
- real, dimension(:,:,:,:), allocatable communication::coin13_r

Corner 2D 1 r.

- real, dimension(:,:,:,:), allocatable communication::coin14_r
  Corner 2D 2 r.
- real, dimension(:,:,:,:), allocatable communication::coin23_r
  Corner 2D 3 r.
- real, dimension(:,:,:,:), allocatable communication::coin24_r
  Corner 2D 4 r.
- real, dimension(:,:,:,:), allocatable communication::coin135_e
  Corner 3D 1 e.
- real, dimension(:,:,:,:), allocatable communication::coin136_e
  Corner 3D 2 e.
- real, dimension(:,:,:,:), allocatable communication::coin145_e
  Corner 3D 3 e.
- real, dimension(:,:,:,:), allocatable communication::coin146_e
  Corner 3D 4 e.
- real, dimension(:,:,:,:), allocatable communication::coin235_e
  Corner 3D 5 e.
- real, dimension(:,:,:,:), allocatable communication::coin236_e
  Corner 3D 6 e.
- real, dimension(:,:,:,:), allocatable communication::coin245_e
  Corner 3D 7 e.
- real, dimension(:,:,:,:), allocatable communication::coin246_e
  Corner 3D 8 e.
- real, dimension(:,:,:,:), allocatable communication::coin135_r
  Corner 3D 1 r.
- real, dimension(:,:,:,:), allocatable communication::coin136_r
  Corner 3D 2 r.
- real, dimension(:,:,:,:), allocatable communication::coin145_r
  Corner 3D 3 r.
- real, dimension(:,:,:,:), allocatable communication::coin146_r
  Corner 3D 4 r.
- real, dimension(:,:,:,:), allocatable communication::coin235_r
  Corner 3D 5 r.

- real, dimension(:,:,:,:), allocatable communication::coin236_r
  Corner 3D 6 r.
- real, dimension(:,:,:,:), allocatable communication::coin245_r
  Corner 3D 7 r.
- real, dimension(:,:,:,:), allocatable communication::coin246_r
  Corner 3D 8 r.
- integer, dimension(:,:,:), allocatable para::grid_cpu
  Number of cpus in the x,y,z, directions.
- integer, dimension(:,:), allocatable para::x_start_cpu
  Starting x,y,z coordinates which cpu holds.
- integer, dimension(:,:), allocatable para::x_end_cpu
  Ending x,y,z coordinates which cpu holds.
- integer, dimension(:), allocatable para::lim_x
  lim_x
- integer, dimension(:), allocatable para::lim_y
  lim_y
- integer, dimension(:), allocatable para::lim_z
  lim_z
- integer, dimension(-1:1,-1:1,-1:1) para::voisin
  Cpu neighbours.
- integer, dimension(3) para::grid_pos
  grid_pos
- integer, dimension(3) para::x_start
  x_start
- integer, dimension(3) para::x_end
  x_end
- integer para::ncpu
  Total number of cpus.
- integer para::mype
  Cpu rank.
- integer para::ncpu_x
  Number of cpus in the x direction.
- integer para::ncpu_y
  Number of cpus in the y direction.

- integer para::ncpu_z

  Number of cpus in the z direction.

9.49.1 Detailed Description

Contains modules communication and para and subroutines init_para(), allocate_array_com(), communications(), com_1D(), com_2D() and com_3D(). This file contains all the variables and routines necessary to the communications between cpus.

Definition in file para.f90.

9.49.2 Function Documentation

9.49.2.1 subroutine allocate_array_com ()

Allocates the arrays used in the communication process.

Definition at line 303 of file para.f90.

References communication::ar13_e, communication::ar13_r, communication::ar14_e, communication::ar14_r, communication::ar15_e, communication::ar15_r, communication::ar16_e, communication::ar16_r, communication::ar23_e, communication::ar23_r, communication::ar24_e, communication::ar24_r, communication::ar25_e, communication::ar25_r, communication::ar26_e, communication::ar26_r, communication::ar35_e, communication::ar35_r, communication::ar36_e, communication::ar36_r, communication::ar45_e, communication::ar45_r, communication::ar46_e, communication::ar46_r, communication::coin135_e, communication::coin135_r, communication::coin136_e, communication::coin136_r, communication::coin13_e, communication::coin13_r, communication::coin145_e, communication::coin145_r, communication::coin146_e, communication::coin146_r, communication::coin14_e, communication::coin14_r, communication::coin235_e, communication::coin235_r, communication::coin236_e, communication::coin236_r, communication::coin23_e, communication::coin23_r, communication::coin245_e, communication::coin245_r, communication::coin246_e, communication::coin246_r, communication::coin24_e, communication::coin24_r, communication::face1_e, communication::face1_r, communication::face2_e, communication::face2_r, communication::face3_e, communication::face3_r, communication::face4_e, communication::face4_r, communication::face5_e, communication::face5_r, communication::face6_e, communication::face6_r, parameters::Nbuf, parameters::ndim, parameters::nvar, and parameters::nx.

Referenced by heracles().

Here is the caller graph for this function:

9.49.2.2 subroutine communications (real *tcomm*)

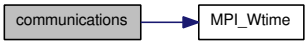
Perform communications and return cputime used.

Definition at line 370 of file para.f90.

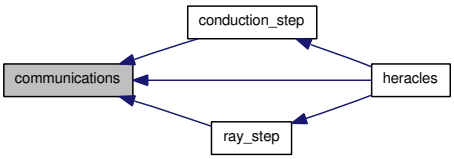
References communication::ar13_e, communication::ar13_r, communication::ar14_e, communication::ar14_r, communication::ar15_e, communication::ar15_r, communication::ar16_e, communication::ar16_r, communication::ar23_e, communication::ar23_r, communication::ar24_e, communication::ar24_r, communication::ar25_e, communication::ar25_r, communication::ar26_e, communication::ar26_r, communication::ar35_e, communication::ar35_r, communication::ar36_e, communication::ar36_r, communication::ar45_e, communication::ar45_r, communication::ar46_e, communication::ar46_r, var::B, communication::coin135_e, communication::coin135_r, communication::coin136_e, communication::coin136_r, communication::coin13_e, communication::coin13_r, communication::coin145_e, communication::coin145_r, communication::coin146_e, communication::coin146_r, communication::coin14_e, communication::coin14_r, communication::coin235_e, communication::coin235_r, communication::coin236_e, communication::coin236_r, communication::coin23_e, communication::coin23_r, communication::coin245_e, communication::coin245_r, communication::coin246_e, communication::coin246_r, communication::coin24_e, communication::coin24_r, var::E, communication::face1_e, communication::face1_r, communication::face2_e, communication::face2_r, communication::face3_e, communication::face3_r, communication::face4_e, communication::face4_r, communication::face5_e, communication::face5_r, communication::face6_e, communication::face6_r, MPI_Wtime(), parameters::N_vit, parameters::Nbuf, parameters::ndim, parameters::nvar, parameters::nx, var::rho, var::rhou, divers::verbose, and para::voisin.

Referenced by conduction_step(), heracles(), and ray_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.49.2.3 subroutine init_para ()

This subroutine initialises the parallelisation of the simualtion. It decomposes the compuational domain by distributing it evenly among the cpus.

Definition at line 124 of file para.f90.

References para::grid_cpu, para::grid_pos, cl::icl, cl::icl0, cl_con::icl_con, cl_con::icl_con_glob, cl_dif::icl_dif, cl_dif::icl_dif_glob, cl::icl_glob, cl_ray::icl_ray, cl_ray::icl_ray_glob, para::lim_x, para::lim_y, para::lim_z, para::mype, parameters::N_vit, para::ncpu, para::ncpu_x, para::ncpu_y, para::ncpu_z, parameters::ndim, parameters::nvar, parameters::nvar_ray, parameters::nx, parameters::nx_cpu, parameters::nx_glob, para::voisin, para::x_end, para::x_end_cpu, para::x_start, and para::x_start_cpu.

Referenced by heracles().

Here is the caller graph for this function:

9.50 main/para_generique.f90 File Reference

Contains module communication_gen and subroutines allocate_array_com_gen(), communications_gen(), com_1D_gen(), com_2D_gen() and com_3D().

Modules

- module communication_gen

  Contains variables arrays used to communicate ghost cell data across cpus.

Functions/Subroutines

- subroutine allocate_array_com_gen

  Allocates the arrays used in the communication process.

- subroutine communications_gen (Tab, n, tcomm)

  Perform communications and return cputime used.

Variables

- real, dimension(:,:,:,:), allocatable communication_gen::face1_e

  Face 1 e.

- real, dimension(:,:,:,:), allocatable communication_gen::face2_e

  Face 2 e.

- real, dimension(:,:,:,:), allocatable communication_gen::face3_e

  Face 3 e.

- real, dimension(:,:,:,:), allocatable communication_gen::face4_e

  Face 4 e.

- real, dimension(:,:,:,:), allocatable communication_gen::face5_e

  Face 5 e.

- real, dimension(:,:,:,:), allocatable communication_gen::face6_e

  Face 6 e.

- real, dimension(:,:,:,:), allocatable communication_gen::face1_r

  Face 1 r.

- real, dimension(:,:,:,:), allocatable communication_gen::face2_r

  Face 2 r.

- real, dimension(:,:,:,:), allocatable communication_gen::face3_r

  Face 3 r.

- real, dimension(:,:,:,:), allocatable communication_gen::face4_r
  Face 4 r.
- real, dimension(:,:,:,:), allocatable communication_gen::face5_r
  Face 5 r.
- real, dimension(:,:,:,:), allocatable communication_gen::face6_r
  Face 6 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar13_e
  Edge 1 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar14_e
  Edge 2 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar23_e
  Edge 3 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar24_e
  Edge 4 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar15_e
  Edge 5 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar25_e
  Edge 6 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar35_e
  Edge 7 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar45_e
  Edge 8 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar16_e
  Edge 9 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar26_e
  Edge 10 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar36_e
  Edge 11 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar46_e
  Edge 12 e.
- real, dimension(:,:,:,:), allocatable communication_gen::ar13_r
  Edge 1 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar14_r
  Edge 2 r.

- real, dimension(:,:,:,:), allocatable communication_gen::ar23_r
  Edge 3 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar24_r
  Edge 4 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar15_r
  Edge 5 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar25_r
  Edge 6 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar35_r
  Edge 7 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar45_r
  Edge 8 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar16_r
  Edge 9 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar26_r
  Edge 10 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar36_r
  Edge 11 r.
- real, dimension(:,:,:,:), allocatable communication_gen::ar46_r
  Edge 12 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin13_e
  Corner 2D 1 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin14_e
  Corner 2D 2 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin23_e
  Corner 2D 3 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin24_e
  Corner 2D 4 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin13_r
  Corner 2D 1 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin14_r
  Corner 2D 2 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin23_r

Corner 2D 3 r.

- real, dimension(:,:,:,:), allocatable communication_gen::coin24_r
  Corner 2D 4 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin135_e
  Corner 3D 1 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin136_e
  Corner 3D 2 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin145_e
  Corner 3D 3 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin146_e
  Corner 3D 4 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin235_e
  Corner 3D 5 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin236_e
  Corner 3D 6 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin245_e
  Corner 3D 7 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin246_e
  Corner 3D 8 e.
- real, dimension(:,:,:,:), allocatable communication_gen::coin135_r
  Corner 3D 1 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin136_r
  Corner 3D 2 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin145_r
  Corner 3D 3 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin146_r
  Corner 3D 4 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin235_r
  Corner 3D 5 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin236_r
  Corner 3D 6 r.
- real, dimension(:,:,:,:), allocatable communication_gen::coin245_r
  Corner 3D 7 r.

- real, dimension(:,:,:,:), allocatable communication_gen::coin246_r

  Corner 3D 8 r.

- integer, parameter communication_gen::Nmax = 2

  nmax

9.50.1 Detailed Description

Contains module communication_gen and subroutines allocate_array_com_gen(), communications_gen(), com_1D_gen(), com_2D_gen() and com_3D(). This file contains all the variables and routines necessary to the generic communications between cpus.

Definition in file para_generique.f90.

9.50.2 Function Documentation

9.50.2.1 subroutine allocate_array_com_gen ()

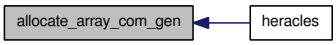
Allocates the arrays used in the communication process.

Definition at line 94 of file para_generique.f90.

References communication_gen::ar13_e, communication_gen::ar13_r, communication_gen::ar14_e, communication_gen::ar14_r, communication_gen::ar15_e, communication_gen::ar15_r, communication_gen::ar16_e, communication_gen::ar16_r, communication_gen::ar23_e, communication_gen::ar23_r, communication_gen::ar24_e, communication_gen::ar24_r, communication_gen::ar25_e, communication_gen::ar25_r, communication_gen::ar26_e, communication_gen::ar26_r, communication_gen::ar35_e, communication_gen::ar35_r, communication_gen::ar36_e, communication_gen::ar36_r, communication_gen::ar45_e, communication_gen::ar45_r, communication_gen::ar46_e, communication_gen::ar46_r, communication_gen::coin135_e, communication_gen::coin135_r, communication_gen::coin136_e, communication_gen::coin136_r, communication_gen::coin13_e, communication_gen::coin13_r, communication_gen::coin145_e, communication_gen::coin145_r, communication_gen::coin146_e, communication_gen::coin146_r, communication_gen::coin14_e, communication_gen::coin14_r, communication_gen::coin235_e, communication_gen::coin235_r, communication_gen::coin236_e, communication_gen::coin236_r, communication_gen::coin23_e, communication_gen::coin23_r, communication_gen::coin245_e, communication_gen::coin245_r, communication_gen::coin246_e, communication_gen::coin246_r, communication_gen::coin24_e, communication_gen::coin24_r, communication_gen::face1_e, communication_gen::face1_r, communication_gen::face2_e, communication_gen::face2_r, communication_gen::face3_e, communication_gen::face3_r, communication_gen::face4_e, communication_gen::face4_r, communication_gen::face5_e, communication_gen::face5_r, communication_gen::face6_e, communication_gen::face6_r, parameters::Nbuf, parameters::ndim, communication_gen::Nmax, and parameters::nx.

Referenced by heracles().

Here is the caller graph for this function:



9.50.2.2 subroutine communications_gen (real,dimension(nxmin(1) *Tab*, integer,dimension(nxmin(1) *n*, real *tcomm*)

Perform communications and return cputime used.

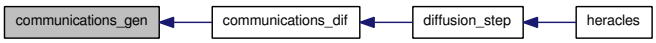
Definition at line 162 of file para_generique.f90.

References communication_gen::ar13_e, communication_gen::ar13_r, communication_gen::ar14_e, communication_gen::ar14_r, communication_gen::ar15_e, communication_gen::ar15_r, communication_gen::ar16_e, communication_gen::ar16_r, communication_gen::ar23_e, communication_gen::ar23_r, communication_gen::ar24_e, communication_gen::ar24_r, communication_gen::ar25_e, communication_gen::ar25_r, communication_gen::ar26_e, communication_gen::ar26_r, communication_gen::ar35_e, communication_gen::ar35_r, communication_gen::ar36_e, communication_gen::ar36_r, communication_gen::ar45_e, communication_gen::ar45_r, communication_gen::ar46_e, communication_gen::ar46_r, communication_gen::coin135_e, communication_gen::coin135_r, communication_gen::coin136_e, communication_gen::coin136_r, communication_gen::coin13_e, communication_gen::coin13_r, communication_gen::coin145_e, communication_gen::coin145_r, communication_gen::coin146_e, communication_gen::coin146_r, communication_gen::coin14_e, communication_gen::coin14_r, communication_gen::coin235_e, communication_gen::coin235_r, communication_gen::coin236_e, communication_gen::coin236_r, communication_gen::coin23_e, communication_gen::coin23_r, communication_gen::coin245_e, communication_gen::coin245_r, communication_gen::coin246_e, communication_gen::coin246_r, communication_gen::coin24_e, communication_gen::coin24_r, communication_gen::face1_e, communication_gen::face1_r, communication_gen::face2_e, communication_gen::face2_r, communication_gen::face3_e, communication_gen::face3_r, communication_gen::face4_e, communication_gen::face4_r, communication_gen::face5_e, communication_gen::face5_r, communication_gen::face6_e, communication_gen::face6_r, MPI_Wtime(), parameters::Nbuf, parameters::ndim, parameters::nx, divers::verbose, and para::voisin.

Referenced by communications_dif().

Here is the call graph for this function:

Here is the caller graph for this function:



9.51 main/pasdt.f90 File Reference

Contains subroutine pasdt().

Functions/Subroutines

- subroutine pasdt

  Compute maximum allowed timestep from various stability criteria (CFL, MHD, RHD, gravity, etc.

9.51.1 Detailed Description

Contains subroutine pasdt().

Definition in file pasdt.f90.

9.51.2 Function Documentation

9.51.2.1 subroutine pasdt ()

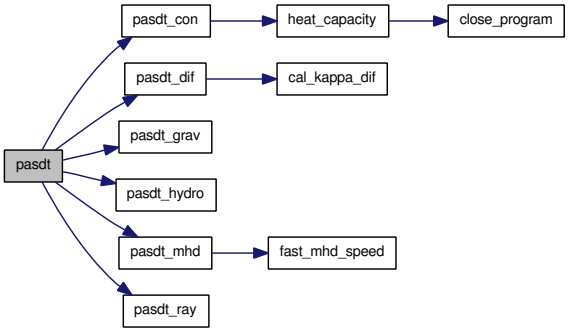
Compute maximum allowed timestep from various stability criteria (CFL, MHD, RHD, gravity, etc. ..)

Definition at line 15 of file pasdt.f90.

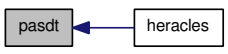
References unites::an, divers::CONDUC, divers::DIFFU, divers::dt, divers::dt_cfl, divers::dt_ff, divers::dt_max, divers::dt_mhd, divers_ray::dt_ray, divers::dtold, divers::GRAV, divers::HYDRO, para::mype, pasdt_con(), pasdt_dif(), pasdt_grav(), pasdt_hydro(), pasdt_mhd(), pasdt_ray(), divers::RAY, unites::seconde, and divers::verbose.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:



9.52 main/prime.f90 File Reference

Contains module prime and subroutines find_prime(), primedec() and cpudec().

Modules

- module prime

  Contains variables used for prime number computations.

Functions/Subroutines

- subroutine find_prime

  Decomposes Number into prime factors.

- subroutine PrimeDec (n, Dec, Np)

  Decomposes n into prime factors.

- subroutine CpuDec (Ncpu, Cpu_dim, NN)

  Decomposes total number of cpus into prime factors.

Variables

- integer, parameter prime::Pmax = 50000
- integer, dimension(pmax) prime::PrimeNumber
- integer, dimension(pmax) prime::Number
- integer, parameter prime::Pdmax = 200
- integer prime::Nprime

9.52.1 Detailed Description

Contains module prime and subroutines find_prime(), primedec() and cpudec(). This file contains the subroutines which are used to decompose the computational volume and split it evenly between the cpus.

Definition in file prime.f90.

9.52.2 Function Documentation

9.52.2.1 subroutine CpuDec (integer *Ncpu*, integer,dimension(ndim) *Cpu_Dim*, integer,dimension(ndim) *NN*)

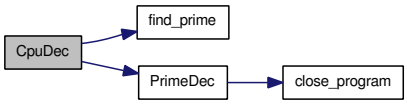
Decomposes total number of cpus into prime factors.

Definition at line 126 of file prime.f90.

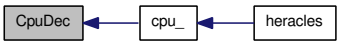
References find_prime(), and PrimeDec().

Referenced by cpu_().

Here is the call graph for this function:



Here is the caller graph for this function:



9.52.2.2 subroutine find_prime ()

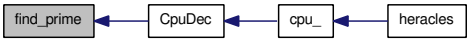
Decomposes Number into prime factors.

Definition at line 37 of file prime.f90.

References prime::Nprime, prime::Number, prime::Pmax, and prime::PrimeNumber.

Referenced by CpuDec().

Here is the caller graph for this function:



9.52.2.3 subroutine PrimeDec (integer *n*, integer,dimension(pdmax) *Dec*, integer *Np*)

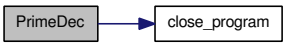
Decomposes n into prime factors.

Definition at line 79 of file prime.f90.

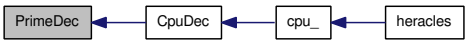
References close_program(), para::mype, prime::Pmax, and prime::PrimeNumber.

Referenced by CpuDec().

Here is the call graph for this function:



Here is the caller graph for this function:



9.53 main/read_params.f90 File Reference

Contains subroutines read_params() and default_params().

Functions/Subroutines

- subroutine read_params (kmax, format_sortie)

  Reads in all the simulation parameters from the namelist.

- subroutine default_params (kmax, format_sortie)

  Sets the simulation parameters to default values in case they are not set in the namelist.

9.53.1 Detailed Description

Contains subroutines read_params() and default_params().

Definition in file read_params.f90.

9.53.2 Function Documentation

9.53.2.1 subroutine default_params (integer *kmax*, character(len=3) *format_sortie*)

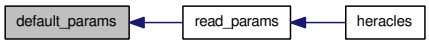
Sets the simulation parameters to default values in case they are not set in the namelist.

Definition at line 107 of file read_params.f90.

References divers::comments, divers::CONDUC, divers::datanames, divers::dataunits, divers::DETO, divers::DIFFU, divers::dt_max, mfilm::FILM, divers::fourdim, divers::GRAV, divers::HISTOGRAMME, divers::HYDRO, divers::i_restart, cl::icl, cl_con::icl_con, cl_con::icl_con_glob, cl_dif::icl_dif_glob, cl::icl_glob, cl_ray::icl_ray_glob, divers::ISOTHERME, divers::name_of_job, parameters::Nbuf, para::ncpu_x, para::ncpu_y, para::ncpu_z, divers::ndata, parameters::ndim, divers::Nimp, divers::noutd, divers::nrepr, divers::nreprd, divers::nsupp, divers::nsx, divers::nsy, divers::nsz, parameters::nx_glob, divers::rankine, divers::RAY, divers::REF, divers::reprise, parameters::riemann, parameters::riemann2d, parameters::slope_type, divers::Solver, divers::STIR, divers::suppdim, divers::suppnames, divers::suppunits, divers::tend, divers::tout, divers::trepr, and divers::verbose.

Referenced by read_params().

Here is the caller graph for this function:



9.53.2.2 subroutine read_params (integer *kmax*, character(len=3) *format_sortie*)

Reads in all the simulation parameters from the namelist.

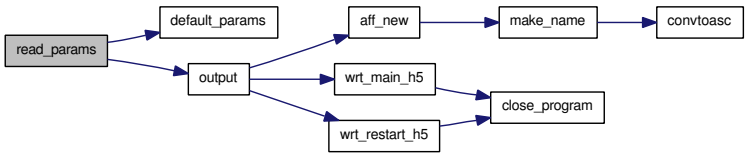
Definition at line 15 of file read_params.f90.

References divers::comments, divers::CONDUC, divers::datanames, divers::dataunits, default_params(), divers::DETO, divers::DIFFU, divers::dt_max, mfilm::FILM, divers::GRAV, divers::HISTOGRAMME,

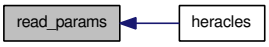
divers::HYDRO, divers::i_restart, cl_dif::icl_dif_glob, cl::icl_glob, cl_ray::icl_ray_glob, divers::ISOTHERME, para::mype, divers::name_of_job, parameters::Nbuf, para::ncpu_x, para::ncpu_y, para::ncpu_z, divers::ndata, parameters::ndim, divers::Nimp, divers::nrepr, divers::nsupp, divers::nsx, divers::nsy, divers::nsz, parameters::nx_glob, output(), divers::rankine, divers::RAY, divers::REF, divers::reprise, parameters::riemann, parameters::riemann2d, parameters::slope_type, divers::Solver, divers::STIR, divers::suppdim, divers::suppnames, divers::suppunits, cputime::temps_comm, cputime::temps_conduc, cputime::temps_dif, cputime::temps_hydro, cputime::temps_io, cputime::temps_ray, cputime::temps_ref, divers::tend, divers::tout, divers::trepr, and divers::verbose.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:



9.54 main/restart.f90 File Reference

Contains subroutine restart_new().

Functions/Subroutines

- subroutine restart_new (k)

  Performs simulation restart.

9.54.1 Detailed Description

Contains subroutine restart_new().

Definition in file restart.f90.

9.54.2 Function Documentation

9.54.2.1 subroutine restart_new (integer *k*)

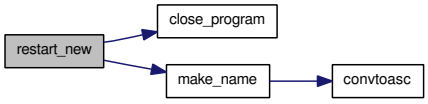
Performs simulation restart. The data cube is read in and the variables are set to the values from the previous run.

Definition at line 16 of file restart.f90.

References var::B, close_program(), divers::DIFFU, divers::dt, divers::dtold, var::E, diffusion::Er_dif, varray::Eray, varray::Fray, gammas::gamma, geom::geometrie, divers::i_restart, make_name(), unites::Mans, divers::mu, para::mype, para::ncpu, parameters::ndim, geom::nshift_gr, parameters::nx, parameters::nx_glob, divers::RAY, var::rho, var::rhou, geom::shift_gr, divers::temps, unites_sortie::u_dens, unites_sortie::u_Fr, unites_sortie::u_L, unites_sortie::u_Mom, unites_sortie::u_T, geom::x, para::x_end, geom::x_glob, and para::x_start.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.55 main/set_units_out.f90 File Reference

Contains subroutine set_units_out().

Functions/Subroutines

- subroutine set_units_out (tend, dt_max)

  Sets the physical units for the datacube outputs.

9.55.1 Detailed Description

Contains subroutine set_units_out().

Definition in file set_units_out.f90.

9.55.2 Function Documentation

9.55.2.1 subroutine set_units_out (real *tend*, real *dt_max*)

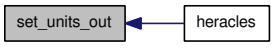
Sets the physical units for the datacube outputs.

Definition at line 14 of file set_units_out.f90.

References unites::centimetre, unites::gramme, para::mype, unites::seconde, unites_sortie::u_dens, unites_sortie::u_E, unites_sortie::u_Fr, unites_sortie::u_L, unites_sortie::u_M, unites_sortie::u_Mom, unites_sortie::u_T, unites_sortie::u_Vit, and unites_sortie::u_Vol.

Referenced by heracles().

Here is the caller graph for this function:



9.56 main/slopes.f90 File Reference

Contains subroutine slopes() and functions minmod(), moncen(), moyharm() and vavl().

Functions/Subroutines

- subroutine slopes (xx, pxxG, pxxD, nxmin, nxmax, nymin, nymax, nzmin, nzmax, slope_type)

  Computes slopes for hydro and radiative transfer variables in 3 dimensions.

- real minmod (pa, pb)

  Minmod averaging function.

- real moncen (pa, pb, pc)

  Monotonised centred averaging function.

- real moyharm (pa, pb)

  Harmonic averaging function.

- real vavl (pa, pb)

  Van Albada & Van Leer averaging function.

9.56.1 Detailed Description

Contains subroutine slopes() and functions minmod(), moncen(), moyharm() and vavl().
Definition in file slopes.f90.

9.56.2 Function Documentation

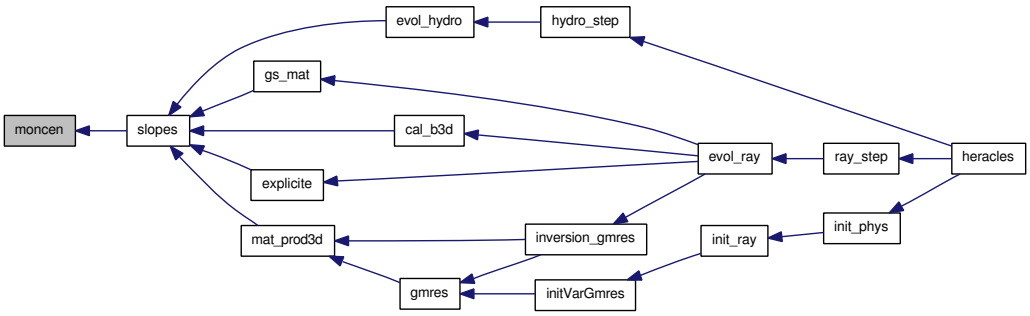
9.56.2.1 real slopes::minmod (real,intent(in) *pa*, real,intent(in) *pb*)

Minmod averaging function.
Definition at line 355 of file slopes.f90.

9.56.2.2 real slopes::moncen (real,intent(in) *pa*, real,intent(in) *pb*, real,intent(in) *pc*)

Monotonised centred averaging function.
Definition at line 377 of file slopes.f90.
Referenced by slopes().

Here is the caller graph for this function:



9.56.2.3 real slopes::moyharm (real,intent(in) *pa*, real,intent(in) *pb*)

Harmonic averaging function.

Definition at line 409 of file slopes.f90.

9.56.2.4 subroutine slopes (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *xx*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax,1:ndim) *pxxG*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax,1:ndim) *pxxD*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*, integer *slope_type*)

Computes slopes for hydro and radiative transfer variables in 3 dimensions. The different slope limiters are:

- 1 : Minmod
- 2 : Moncen
- 3 : Positivity preserving
- 4 : Harmonic average
- 5 : Van Albada & Van Leer
- 6 : Superbee

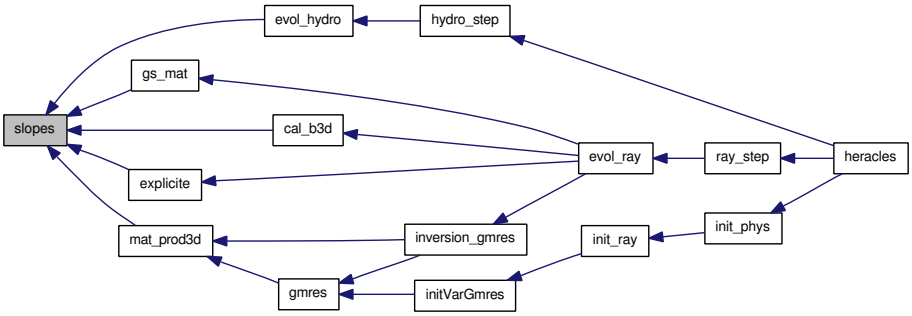
Definition at line 23 of file slopes.f90.

References geom::dxc, moncen(), parameters::ndim, and const::two.

Referenced by cal_b3d(), evol_hydro(), explicite(), gs_mat(), and mat_prod3d().

Here is the call graph for this function:

Here is the caller graph for this function:



9.56.2.5 real slopes::vavl (real,intent(in) *pa*, real,intent(in) *pb*)

Van Albada & Van Leer averaging function.

Definition at line 435 of file slopes.f90.

9.57 main/user_init.f90 File Reference

Contains subroutine user_init().

Functions/Subroutines

- subroutine user_init

  Initialises any extra variables which are needed by the user for a specific simulation.

9.57.1 Detailed Description

Contains subroutine user_init().

Definition in file user_init.f90.

9.57.2 Function Documentation

9.57.2.1 subroutine user_init ()

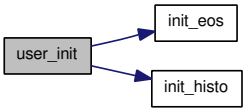
Initialises any extra variables which are needed by the user for a specific simulation.

Definition at line 15 of file user_init.f90.

References init_eos(), and init_histo().

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.58 main/user_output.f90 File Reference

Contains subroutine user_output().

Functions/Subroutines

- subroutine user_output (kk)

  Will be used to create any user-specified outputs.

9.58.1 Detailed Description

Contains subroutine user_output().

Definition in file user_output.f90.

9.58.2 Function Documentation

9.58.2.1 subroutine user_output (integer *kk*)

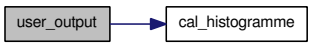
Will be used to create any user-specified outputs.

Definition at line 14 of file user_output.f90.

References cal_histogramme(), and divers::HISTOGRAMME.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.59 main/user_step.f90 File Reference

Contains subroutine user_step().

Functions/Subroutines

- subroutine user_step

  Will be called at every timestep.

9.59.1 Detailed Description

Contains subroutine user_step().

Definition in file user_step.f90.

9.59.2 Function Documentation

9.59.2.1 subroutine user_step ()

Will be called at every timestep. This is where any user-specified functions or modifications should be included for a specific simulation. By default it is empty.

Definition at line 17 of file user_step.f90.

Referenced by heracles().

Here is the caller graph for this function:

9.60 mhd/ctoprim.f90 File Reference

Contains subroutine ctoprim().

Functions/Subroutines

- subroutine ctoprim (uin, q, bf, gravin, dt, iu1, iu2, ju1, ju2, ku1, ku2, nvar, ndim)

  Converts conservative variables to primitive in mhd.

9.60.1 Detailed Description

Contains subroutine ctoprim().

Definition in file ctoprim.f90.

9.60.2 Function Documentation

9.60.2.1 subroutine ctoprim (real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar+3) *uin*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar) *q*, real,dimension(iu1:iu2+1,ju1:ju2+1,ku1:ku2+1,1:3) *bf*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:ndim) *gravin*, real *dt*, integer *iu1*, integer *iu2*, integer *ju1*, integer *ju2*, integer *ku1*, integer *ku2*, integer *nvar*, integer *ndim*)

Converts conservative variables to primitive in mhd.

Definition at line 14 of file ctoprim.f90.

References gammas::gamma, const::half, const::one, parameters::smallc, parameters::smallr, and const::zero.

Referenced by evol_mhd().

Here is the caller graph for this function:

9.61 mhd/evol_mhd.f90 File Reference

Contains subroutine evol_mhd().

Functions/Subroutines

- subroutine evol_mhd (imin, imax, jmin, jmax, kmin, kmax)

  Performs a single timestep evolution of mhd.

9.61.1 Detailed Description

Contains subroutine evol_mhd().

Definition in file evol_mhd.f90.

9.61.2 Function Documentation

9.61.2.1 subroutine evol_mhd (integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,intent(in) *kmax*)

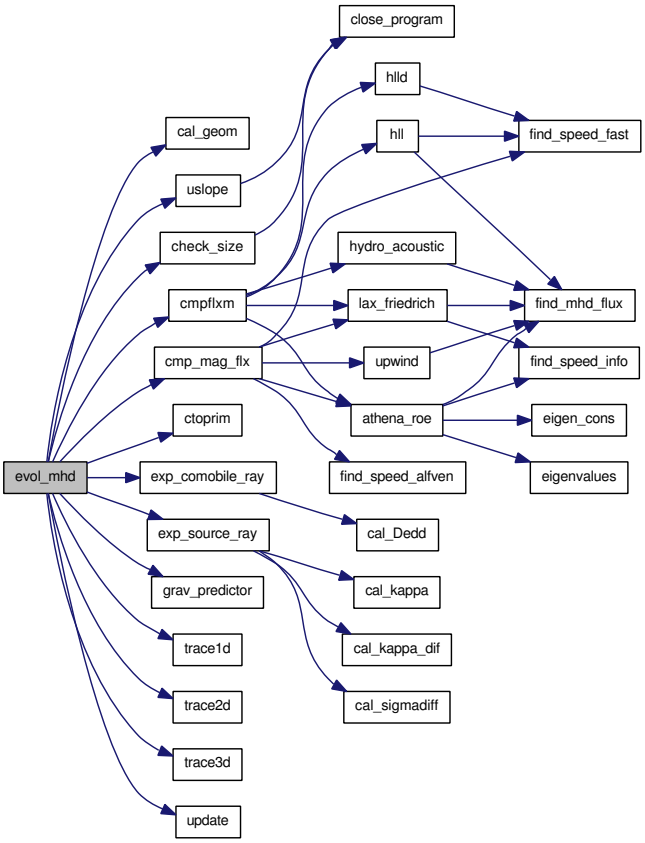
Performs a single timestep evolution of mhd.

Definition at line 14 of file evol_mhd.f90.

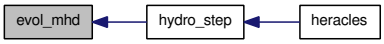
References var::B, varold::Bold, cal_geom(), check_size(), cmp_mag_flx(), cmpflxm(), ctoprim(), divers::DIFFU, geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, var_loc::Einloc, varold::Eold, exp_comobile_ray(), exp_source_ray(), gammas::gamma, geom::geom_dir, divers::GRAV, grav_predictor(), cl::icl, parameters::ndim, parameters::nvar, parameters::nx, gravity::phi, var_loc::philoc, divers::RAY, var::rho, varold::rhoold, var::rhou, varold::rhouold, trace1d(), trace2d(), trace3d(), update(), Etat_GDS::uS, uslope(), divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by hydro_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.62 mhd/godunov_utils.f90 File Reference

contains subroutines upwind(), lax_friedrich(), hll(), hlld(), find_mhd_flux(), find_mhd_flux2(), find_speed_info(), find_speed_fast(), find_speed_alfven(), hydro_acoustic(), athena_roe(), eigenvalues() and eigen_cons()

Functions/Subroutines

- subroutine upwind (qleft, qright, fgdnv, zero_flux)

  1D upwind Riemann solver.

- subroutine lax_friedrich (qleft, qright, fmean, zero_flux)

  Lax -Friedrich Riemann solver.

- subroutine hll (qleft, qright, fgdnv)

  HLL Riemann solver.

- subroutine hlld (qleft, qright, fgdnv, ro, uo, vo, wo, bo, co, ptoto, nvar)

  HLLD Riemann solver.

- subroutine find_mhd_flux (qvar, cvar, ff)

  Compute the 1d mhd fluxes from the conservative variables.

- subroutine find_mhd_flux2 (qvar, cvar, ff)

  Compute the 1d mhd fluxes from the conservative variables.

- subroutine find_speed_info (qvar, vel_info)

  Calculate the fastest velocity at which information is exchanged at the interface.

- subroutine find_speed_fast (qvar, vel_info, nvar)

  Calculate the fast magnetosonic velocity.

- subroutine find_speed_alfven (qvar, vel_info, nvar)

  Calculate the Alfven velocity.

- subroutine hydro_acoustic (qleft, qright, fgdnv)

  Acoustic Riemann solver.

- subroutine athena_roe (qleft, qright, fmean, zero_flux)

  Athena Roe Riemann solver.

- subroutine eigenvalues (d, vx, vy, vz, p, bx, by, bz, lambda)

  Computes mhd adiabatic eigenvalues.

- subroutine eigen_cons (d, vx, vy, vz, h, bx, by, bz, xfac, yfac, lambda, rem, lem)

  Computes Roe eigenvalues.

9.62.1 Detailed Description

contains subroutines upwind(), lax_friedrich(), hll(), hlld(), find_mhd_flux(), find_mhd_flux2(), find_speed_info(), find_speed_fast(), find_speed_alfven(), hydro_acoustic(), athena_roe(), eigenvalues() and eigen_cons() This file contains all the subroutines used to compute mhd fluxes.

Definition in file godunov_utils.f90.

9.62.2 Function Documentation

9.62.2.1 subroutine athena_roe (real,dimension(1:nvar) *qleft*, real,dimension(1:nvar) *qright*, real,dimension(1:nvar) *fmean*, real *zero_flux*)

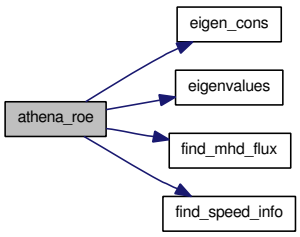
Athena Roe Riemann solver.

Definition at line 760 of file godunov_utils.f90.

References eigen_cons(), eigenvalues(), find_mhd_flux(), find_speed_info(), const::half, parameters::nvar, const::two, and const::zero.

Referenced by cmp_mag_flx(), and cmpflxm().

Here is the call graph for this function:



Here is the caller graph for this function:

9.62.2.2 subroutine eigen_cons (real *d*, real *vx*, real *vy*, real *vz*, real *h*, real *bx*, real *by*, real *bz*, real *xfac*, real *yfac*, real,dimension(7) *lambda*, real,dimension(7,7) *rem*, real,dimension(7,7) *lem*)

Computes Roe eigenvalues. Input arguments: d = roe density vx = roe x velocity vy = roe y velocity vz = roe z velocity bx = magnetic field in sweep direction by = roe y magnetic field bz = roe z magnetic field h = roe enthalpy xfac = ((by^2-byl∗byr)+bz^2-bzl∗bzr))/(2∗rho) yfac = (rho_l+rho_r)/(2∗rho)

output arguments:

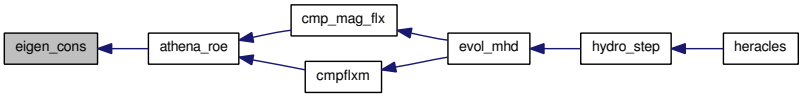
lambda = eigenvalues rem = right eigenmatrix lem = left eigenmatrix

Definition at line 1058 of file godunov_utils.f90.

References gammas::gamma, const::one, and parameters::smallc.

Referenced by athena_roe().

Here is the caller graph for this function:



9.62.2.3 subroutine eigenvalues (real,intent(in) *d*, real,intent(in) *vx*, real,intent(in) *vy*, real,intent(in) *vz*, real,intent(in) *p*, real,intent(in) *bx*, real,intent(in) *by*, real,intent(in) *bz*, real,dimension(1:7),intent(out) *lambda*)

Computes mhd adiabatic eigenvalues. Input arguments: bx = magnetic field in sweep direction d = density vx = x velocity vy = y velocity vz = z velocity by = y magnetic field bz = z magnetic field p = thermal pressure

Output arguments:

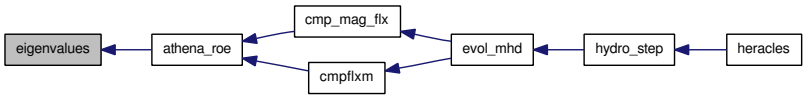
lambda = eigenvalues

Definition at line 987 of file godunov_utils.f90.

References gammas::gamma, and parameters::smallc.

Referenced by athena_roe().

Here is the caller graph for this function:



9.62.2.4 subroutine find_mhd_flux (real,dimension(1:nvar) *qvar*, real,dimension(1:nvar) *cvar*, real,dimension(1:nvar) *ff*)

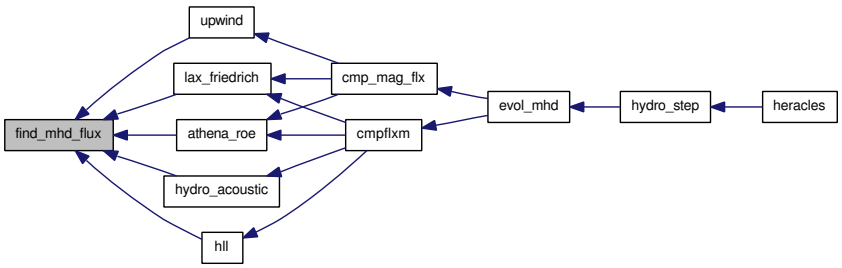
Compute the 1d mhd fluxes from the conservative variables. The structure of qvar is : rho, pressure, vnormal, bnormal, vtransverse1, btransverse1, vtransverse2, btransverse2.

Definition at line 372 of file godunov_utils.f90.

References gammas::gamma, const::half, parameters::nvar, const::one, and const::zero.

Referenced by athena_roe(), hll(), hydro_acoustic(), lax_friedrich(), and upwind().

Here is the caller graph for this function:



9.62.2.5 subroutine find_mhd_flux2 (real,dimension(1:nvar) *qvar*, real,dimension(1:nvar) *cvar*, real,dimension(1:nvar) *ff*)

Compute the 1d mhd fluxes from the conservative variables. qvar: rho,p,vn,bn,vt,bt1,vt2,bt2

Definition at line 439 of file godunov_utils.f90.

References gammas::gamma, const::half, parameters::nvar, and const::one.

9.62.2.6 subroutine find_speed_alfven (real,dimension(1:nvar) *qvar*, real *vel_info*, integer *nvar*)

Calculate the Alfven velocity. The structure of qvar is : rho, pressure, vnormal, bnormal, vtransverse1, btransverse1, vtransverse2, btransverse2

Definition at line 608 of file godunov_utils.f90.

Referenced by cmp_mag_flx().

Here is the caller graph for this function:

9.62.2.7 subroutine find_speed_fast (real,dimension(1:nvar) *qvar*, real *vel_info*, integer *nvar*)

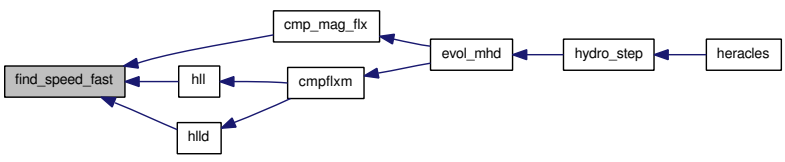
Calculate the fast magnetosonic velocity. The structure of qvar is : rho, pressure, vnormal, bnormal, vtransverse1, btransverse1, vtransverse2, btransverse2

Definition at line 572 of file godunov_utils.f90.

References gammas::gamma, and const::half.

Referenced by cmp_mag_flx(), hll(), and hlld().

Here is the caller graph for this function:



9.62.2.8 subroutine find_speed_info (real,dimension(1:nvar) *qvar*, real *vel_info*)

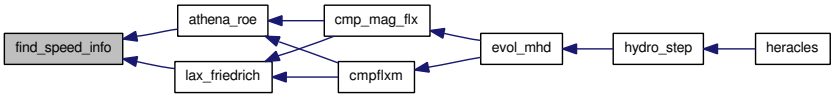
Calculate the fastest velocity at which information is exchanged at the interface. The structure of qvar is : rho, pressure, vnormal, bnormal, vtransverse1, btransverse1, vtransverse2, btransverse2

Definition at line 535 of file godunov_utils.f90.

References gammas::gamma, and const::half.

Referenced by athena_roe(), and lax_friedrich().

Here is the caller graph for this function:



9.62.2.9 subroutine hll (real,dimension(1:nvar) *qleft*, real,dimension(1:nvar) *qright*, real,dimension(1:nvar) *fgdnv*)

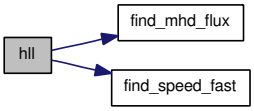
HLL Riemann solver.

Definition at line 114 of file godunov_utils.f90.

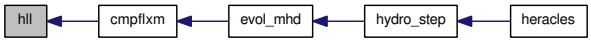
References find_mhd_flux(), find_speed_fast(), const::half, parameters::nvar, and const::zero.

Referenced by cmpflxm().

Here is the call graph for this function:



Here is the caller graph for this function:



9.62.2.10 subroutine hlld (real,dimension(1:nvar) *qleft*, real,dimension(1:nvar) *qright*, real,dimension(1:nvar) *fgdnv*, real *ro*, real *uo*, real *vo*, real *wo*, real *bo*, real *co*, real *ptoto*, integer *nvar*)

HLLD Riemann solver. (Miyoshi & Kusano, 2005, JCP, 208, 315)

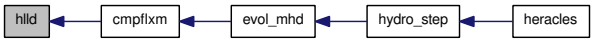
Definition at line 159 of file godunov_utils.f90.

References find_speed_fast(), gammas::gamma, const::half, const::one, and const::zero.

Referenced by cmpflxm().

Here is the call graph for this function:

Here is the caller graph for this function:



9.62.2.11 subroutine hydro_acoustic (real,dimension(1:nvar) *qleft*, real,dimension(1:nvar) *qright*, real,dimension(1:nvar) *fgdnv*)

Acoustic Riemann solver.

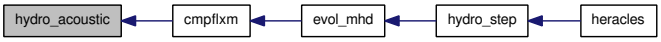
Definition at line 635 of file godunov_utils.f90.

References find_mhd_flux(), gammas::gamma, const::half, parameters::nvar, const::one, parameters::smallc, parameters::smallr, and const::zero.

Referenced by cmpflxm().

Here is the call graph for this function:

Here is the caller graph for this function:



9.62.2.12 subroutine lax_friedrich (real,dimension(1:nvar) *qleft*, real,dimension(1:nvar) *qright*, real,dimension(1:nvar) *fmean*, real *zero_flux*)

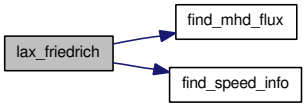
Lax -Friedrich Riemann solver.

Definition at line 68 of file godunov_utils.f90.

References find_mhd_flux(), find_speed_info(), and const::half.

Referenced by cmp_mag_flx(), and cmpflxm().

Here is the call graph for this function:



Here is the caller graph for this function:

9.62.2.13 subroutine upwind (real,dimension(1:nvar) *qleft*, real,dimension(1:nvar) *qright*, real,dimension(1:nvar) *fgdnv*, real *zero_flux*)

1D upwind Riemann solver.

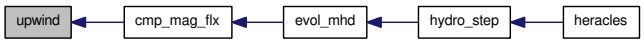
Definition at line 21 of file godunov_utils.f90.

References find_mhd_flux(), and const::half.

Referenced by cmp_mag_flx().

Here is the call graph for this function:

Here is the caller graph for this function:



9.63 mhd/pasdt_mhd.f90 File Reference

contains subroutine pasdt_mhd()

Functions/Subroutines

- subroutine pasdt_mhd

  Computes the mhd timestep.

9.63.1 Detailed Description

contains subroutine pasdt_mhd()

Definition in file pasdt_mhd.f90.

9.63.2 Function Documentation

9.63.2.1 subroutine pasdt_mhd ()

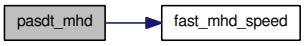
Computes the mhd timestep.

Definition at line 14 of file pasdt_mhd.f90.

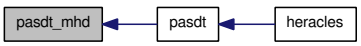
References unites::an, var::B, divers::dt_mhd, var::E, fast_mhd_speed(), gammas::gamma, geom::geom_-dir, const::half, parameters::N_vit, parameters::ndim, parameters::nx, var::rho, var::rhou, divers::verbose, and geom::x.

Referenced by pasdt().

Here is the call graph for this function:



Here is the caller graph for this function:



9.64 mhd/trace.f90 File Reference

contains subroutines trace1d(), trace2d() and trace3d()

Functions/Subroutines

- subroutine trace1d (q, dq, qm, qp, dt, iu1, iu2, ju1, ju2, ku1, ku2, nvar, ndim)

  This subroutine assumed that one considers a 1D problem with 3 velocities (vx,vy,vz) and 3 magnetic fields (bx,by,bz).

- subroutine trace2d (q, bf, dq, dbf, qm, qp, qRT, qRB, qLT, qLB, dt, iu1, iu2, ju1, ju2, ku1, ku2, nvar, ndim)

  Trace in 2 dimensions.

- subroutine trace3d (q, bf, dq, dbf, qm, qp, qRT, qRB, qLT, qLB, dt, iu1, iu2, ju1, ju2, ku1, ku2, nvar, ndim)

  Trace in 3 dimensions.

9.64.1 Detailed Description

contains subroutines trace1d(), trace2d() and trace3d()

Definition in file trace.f90.

9.64.2 Function Documentation

9.64.2.1 subroutine trace1d (real,dimension(iu1:iu2,ju1:ju2,ku1:ku2,1:nvar) *q*, real,dimension(iu1:iu2,ju1:ju2,ku1:ku2,1:nvar,1:ndim) *dq*, real,dimension(iu1:iu2,ju1:ju2,ku1:ku2,1:nvar,1:ndim) *qm*, real,dimension(iu1:iu2,ju1:ju2,ku1:ku2,1:nvar,1:ndim) *qp*, real *dt*, integer *iu1*, integer *iu2*, integer *ju1*, integer *ju2*, integer *ku1*, integer *ku2*, integer *nvar*, integer *ndim*)

This subroutine assumed that one considers a 1D problem with 3 velocities (vx,vy,vz) and 3 magnetic fields (bx,by,bz).

Definition at line 16 of file trace.f90.

References geom::Cartesien, geom::Cylindrique, geom::dx, gammas::gamma, const::half, parameters::smallr, geom::Spherique, const::two, and geom::xcloc.

Referenced by evol_mhd().

Here is the caller graph for this function:

9.64.2.2 subroutine trace2d (real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar) *q*, real,dimension(iu1:iu2+1,ju1:ju2+1,ku1:ku2+1,1:3) *bf*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:ndim) *dq*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:3 ,1:ndim) *dbf*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:ndim) *qm*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:ndim) *qp*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qRT*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qRB*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qLT*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qLB*, real *dt*, integer *iu1*, integer *iu2*, integer *ju1*, integer *ju2*, integer *ku1*, integer *ku2*, integer *nvar*, integer *ndim*)

Trace in 2 dimensions.

Definition at line 168 of file trace.f90.

References geom::Cartesien, geom::Cylindrique, geom::dx, gammas::gamma, const::half, parameters::smallc, parameters::smallr, geom::Spherique, const::two, geom::xcloc, and var_loc::xloc.

Referenced by evol_mhd().

Here is the caller graph for this function:

9.64.2.3 subroutine trace3d (real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar) *q*, real,dimension(iu1:iu2+1,ju1:ju2+1,ku1:ku2+1,1:3) *bf*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:ndim) *dq*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:3 ,1:ndim) *dbf*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:ndim) *qm*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:ndim) *qp*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qRT*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qRB*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qLT*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:3) *qLB*, real *dt*, integer *iu1*, integer *iu2*, integer *ju1*, integer *ju2*, integer *ku1*, integer *ku2*, integer *nvar*, integer *ndim*)

Trace in 3 dimensions.

Definition at line 514 of file trace.f90.

References geom::Cartesien, geom::Cylindrique, geom::dx, gammas::gamma, const::half, parameters::smallc, parameters::smallr, geom::Spherique, const::two, geom::xcloc, and var_loc::xloc.

Referenced by evol_mhd().

Here is the caller graph for this function:

9.65 mhd/umuscl.f90 File Reference

contains subroutines cmpflxm(), cmp_mag_flx(), fast_mhd_speed() and uslope()

Functions/Subroutines

- subroutine cmpflxm (qm, im1, im2, jm1, jm2, km1, km2, qp, ip1, ip2, jp1, jp2, kp1, kp2, ilo, ihi, jlo, jhi, klo, khi, ln, lt1, lt2, bn, bt1, bt2, flx, flx_pre_tot, vel_star, nvar, ndim)

  UNSPLIT Unsplit first (second) order Godunov integrator for polytropic magnetized gas dynamics using the Lax-Friedrish sheme The mesh for the magnetic field is staggered The sheme follows closely the paper by Londrillo & Del Zanna ApJ 2000, 530, 508,.

- subroutine cmp_mag_flx (qRT, irt1, irt2, jrt1, jrt2, krt1, krt2, qRB, irb1, irb2, jrb1, jrb2, krb1, krb2, qLT, ilt1, ilt2, jlt1, jlt2, klt1, klt2, qLB, ilb1, ilb2, jlb1, jlb2, klb1, klb2, ilo, ihi, jlo, jhi, klo, khi, lp1, lp2, lor, bp1, bp2, bor, emf, nvar, ndim)

  2D Riemann solver to compute EMF at cell edges.

- subroutine fast_mhd_speed (rho, p, vx, vy, vz, bx, by, bz, fast_speed)

  Computes the fast magnetosonic velocity.

- subroutine uslope (bf, q, dq, dbf, iu1, iu2, ju1, ju2, ku1, ku2, nvar, ndim)

  Computes mhd slopes.

9.65.1 Detailed Description

contains subroutines cmpflxm(), cmp_mag_flx(), fast_mhd_speed() and uslope()

Definition in file umuscl.f90.

9.65.2 Function Documentation

9.65.2.1 subroutine cmp_mag_flx (real,dimension(irt1:irt2,jrt1:jrt2,krt1:krt2,1:nvar,1:3) *qRT*, integer *irt1*, integer *irt2*, integer *jrt1*, integer *jrt2*, integer *krt1*, integer *krt2*, real,dimension(irb1:irb2,jrb1:jrb2,krb1:krb2,1:nvar,1:3) *qRB*, integer *irb1*, integer *irb2*, integer *jrb1*, integer *jrb2*, integer *krb1*, integer *krb2*, real,dimension(ilt1:ilt2,jlt1:jlt2,klt1:klt2,1:nvar,1:3) *qLT*, integer *ilt1*, integer *ilt2*, integer *jlt1*, integer *jlt2*, integer *klt1*, integer *klt2*, real,dimension(ilb1:ilb2,jlb1:jlb2,klb1:klb2,1:nvar,1:3) *qLB*, integer *ilb1*, integer *ilb2*, integer *jlb1*, integer *jlb2*, integer *klb1*, integer *klb2*, integer *ilo*, integer *ihi*, integer *jlo*, integer *jhi*, integer *klo*, integer *khi*, integer *lp1*, integer *lp2*, integer *lor*, integer *bp1*, integer *bp2*, integer *bor*, real,dimension(ilb1:ilb2,jlb1:jlb2,klb1:klb2) *emf*, integer *nvar*, integer *ndim*)

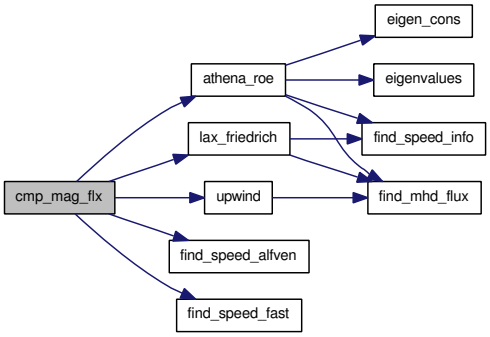
2D Riemann solver to compute EMF at cell edges. Indices of the 2 planar velocity lp1 and lp2 and the orthogonal one, lor and idem for the magnetic field.

Definition at line 280 of file umuscl.f90.

References athena_roe(), find_speed_alfven(), find_speed_fast(), const::forth, const::half, lax_friedrich(), parameters::riemann2d, parameters::smallc, upwind(), and const::zero.

Referenced by evol_mhd().

Here is the call graph for this function:



Here is the caller graph for this function:

9.65.2.2 subroutine cmpflxm (real,dimension(im1:im2,jm1:jm2,km1:km2,1:nvar,1:ndim) *qm*, integer *im1*, integer *im2*, integer *jm1*, integer *jm2*, integer *km1*, integer *km2*, real,dimension(ip1:ip2,jp1:jp2,kp1:kp2,1:nvar,1:ndim) *qp*, integer *ip1*, integer *ip2*, integer *jp1*, integer *jp2*, integer *kp1*, integer *kp2*, integer *ilo*, integer *ihi*, integer *jlo*, integer *jhi*, integer *klo*, integer *khi*, integer *ln*, integer *lt1*, integer *lt2*, integer *bn*, integer *bt1*, integer *bt2*, real,dimension(ip1:ip2,jp1:jp2,kp1:kp2,1:nvar) *flx*, real,dimension(ip1:ip2,jp1:jp2,kp1:kp2) *flx_pre_tot*, real,dimension(ip1:ip2,jp1:jp2,kp1:kp2,1:ndim,1:ndim) *vel_star*, integer *nvar*, integer *ndim*)

UNSPLIT Unsplit first (second) order Godunov integrator for polytropic magnetized gas dynamics using the Lax-Friedrish sheme The mesh for the magnetic field is staggered The sheme follows closely the paper by Londrillo & Del Zanna ApJ 2000, 530, 508,. Inputs/Outputs uin => (const) input state $u_{\rm in} = (\rho, \rho v_x, \rho v_y, \rho v_z, E_{\rm tot}, B_x, B_y, B_z)$ the hydro variable are centered whereas B_x, B_y, B_z are written on the face (staggered mesh) note that in MHD one can be in 1 or 2D and have 3 components for v and B therefore one has 2 variables for the dimension : "ndim" the spatial dimension and "n_vit" the number of velocity components

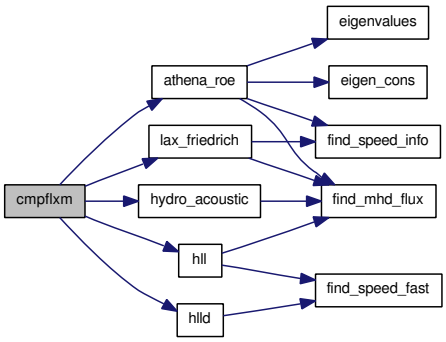
gravin => (const) input gravitational acceleration iu1,iu2 => (const) first and last index of input array, ju1,ju2 => (const) cell centered, ku1,ku2 => (const) including buffer cells. flux <= (modify) return fluxes in the 3 coord directions if1,if2 => (const) first and last index of output array, jf1,jf2 => (const) edge centered, kf1,kf2 => (const) for active cells only. dx,dy,dz => (const) (dx, dy, dz) dt => (const) time step ngrid => (const) number of sub-grids ndim => (const) number of dimensions n_vit => (const) number of velocity components

Definition at line 43 of file umuscl.f90.

References athena_roe(), geom::Cartesien, geom::Cylindrique, geom::ds, geom::dv, geom::dx, const::half, hll(), hlld(), hydro_acoustic(), lax_friedrich(), const::one, parameters::riemann, geom::Spherique, and geom::xcloc.

Referenced by evol_mhd().

Here is the call graph for this function:



Here is the caller graph for this function:

9.65.2.3 subroutine fast_mhd_speed (real *rho*, real *p*, real *vx*, real *vy*, real *vz*, real *bx*, real *by*, real *bz*, real *fast_speed*)

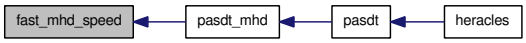
Computes the fast magnetosonic velocity.

Definition at line 717 of file umuscl.f90.

References const::four, gammas::gamma, const::half, and const::zero.

Referenced by pasdt_mhd().

Here is the caller graph for this function:



9.65.2.4 subroutine uslope (real,dimension(iu1:iu2+1,ju1:ju2+1,ku1:ku2+1,1:3) *bf*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar) *q*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:nvar,1:ndim) *dq*, real,dimension(iu1:iu2 ,ju1:ju2 ,ku1:ku2 ,1:3 ,1:ndim) *dbf*, integer *iu1*, integer *iu2*, integer *ju1*, integer *ju2*, integer *ku1*, integer *ku2*, integer *nvar*, integer *ndim*)

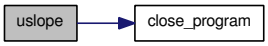
Computes mhd slopes.

Definition at line 751 of file umuscl.f90.

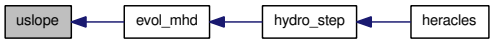
References close_program(), const::half, para::mype, const::one, parameters::slope_type, and const::zero.

Referenced by evol_mhd().

Here is the call graph for this function:



Here is the caller graph for this function:



9.66 mhd/update.f90 File Reference

contains subroutine update()

Functions/Subroutines

- subroutine update (uin, flux, flux_pre_tot, emfx, emfy, emfz, iu1, iu2, ju1, ju2, ku1, ku2, if1, if2, jf1, jf2, kf1, kf2, nvar, ndim)

  Updates the conservative variables after the mhd step using the computed mhd fluxes.

9.66.1 Detailed Description

contains subroutine update()

Definition in file update.f90.

9.66.2 Function Documentation

9.66.2.1 subroutine update (real,dimension(iu1:iu2,ju1:ju2,ku1:ku2,1:nvar+3) *uin*, real,dimension(if1:if2,jf1:jf2,kf1:kf2,1:nvar ,1:ndim) *flux*, real,dimension(if1:if2,jf1:jf2,kf1:kf2 ,1:ndim) *flux_pre_tot*, real,dimension(iu1:iu2,ju1:ju2,ku1:ku2) *emfx*, real,dimension(iu1:iu2,ju1:ju2,ku1:ku2) *emfy*, real,dimension(iu1:iu2,ju1:ju2,ku1:ku2) *emfz*, integer *iu1*, integer *iu2*, integer *ju1*, integer *ju2*, integer *ku1*, integer *ku2*, integer *if1*, integer *if2*, integer *jf1*, integer *jf2*, integer *kf1*, integer *kf2*, integer *nvar*, integer *ndim*)

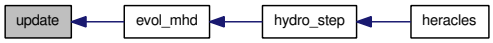
Updates the conservative variables after the mhd step using the computed mhd fluxes.

Definition at line 15 of file update.f90.

References geom::Cartesien, geom::Cylindrique, geom::dv, geom::dx, geom::Spherique, divers::verbose, geom::xcloc, and var_loc::xloc.

Referenced by evol_mhd().

Here is the caller graph for this function:



9.67 rad_transfer/exp_comobile_ray.f90 File Reference

Contains subroutine exp_comobile_ray().

Functions/Subroutines

- subroutine exp_comobile_ray (uS, imin, imax, jmin, jmax, kmin, kmax, nxmin, nxmax, nymin, nymax, nzmin, nzmax, radiation_model)

  This routine explicitly computes the comoving terms of radiation.

9.67.1 Detailed Description

Contains subroutine exp_comobile_ray().

Definition in file exp_comobile_ray.f90.

9.67.2 Function Documentation

9.67.2.1 subroutine exp_comobile_ray (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax,1:ndim,1:ndim),intent(in) *uS*, integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,intent(in) *kmax*, integer,intent(in) *nxmin*, integer,intent(in) *nxmax*, integer,intent(in) *nymin*, integer,intent(in) *nymax*, integer,intent(in) *nzmin*, integer,intent(in) *nzmax*, character(3),intent(in) *radiation_model*)

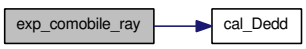
This routine explicitly computes the comoving terms of radiation.

Definition at line 15 of file exp_comobile_ray.f90.

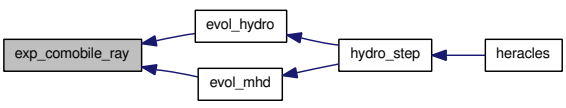
References cal_Dedd(), geom::ds, divers::dt, geom::dv, diffusion::Er_dif, varray::Eray, varray::Erayold, varray::Fray, varray::Frayold, geom::geom_dir, parameters::ndim, divers::verbose, geom::xcloc, and var_loc::xloc.

Referenced by evol_hydro(), and evol_mhd().

Here is the call graph for this function:



Here is the caller graph for this function:



9.68 rad_transfer/exp_source_ray.f90 File Reference

Contains subroutine exp_source_ray().

Functions/Subroutines

- subroutine exp_source_ray (rho, Ein, moment_ray, imin, imax, jmin, jmax, kmin, kmax, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  This routine explicitly computes the radiative source term (kappa rho+sigma)F/c in moment and energy hydro equations.

9.68.1 Detailed Description

Contains subroutine exp_source_ray().

Definition in file exp_source_ray.f90.

9.68.2 Function Documentation

9.68.2.1 subroutine exp_source_ray (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax),intent(in) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax),intent(in) *Ein*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax,1:ndim),intent(out) *moment_ray*, integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,intent(in) *kmax*, integer,intent(in) *nxmin*, integer,intent(in) *nxmax*, integer,intent(in) *nymin*, integer,intent(in) *nymax*, integer,intent(in) *nzmin*, integer,intent(in) *nzmax*)

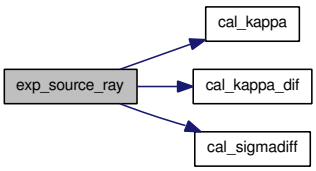
This routine explicitly computes the radiative source term (kappa rho+sigma)F/c in moment and energy hydro equations.

Definition at line 16 of file exp_source_ray.f90.

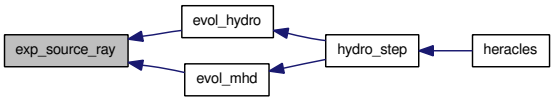
References unites::c, cal_kappa(), cal_kappa_dif(), cal_sigmadiff(), divers::DIFFU, diffusion::Er_dif, diffusion::fld_limiter, varray::Frayold, geom::geom_dir, parameters::ndim, divers::RAY, divers::verbose, geom::xcloc, and var_loc::xloc.

Referenced by evol_hydro(), and evol_mhd().

Here is the call graph for this function:



Here is the caller graph for this function:



9.69 rad_transfer/fld/diffusion.f90 File Reference

Contains subroutine diffusion_ex().

Functions/Subroutines

- subroutine diffusion_ex (imin, imax, jmin, jmax, kmin, kmax)

  This routine solves the FLD equations explicitly.

9.69.1 Detailed Description

Contains subroutine diffusion_ex().

Definition in file diffusion.f90.

9.69.2 Function Documentation

9.69.2.1 subroutine diffusion_ex (integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,intent(in) *kmax*)

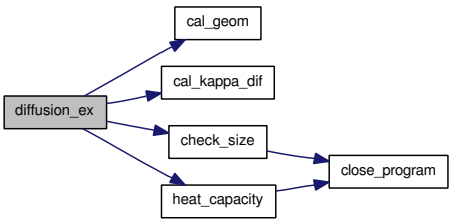
This routine solves the FLD equations explicitly.

Definition at line 14 of file diffusion.f90.

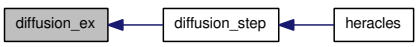
References var::B, unites::c, cal_geom(), cal_kappa_dif(), check_size(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, var_loc::Eloc, varold::Eold, diffusion::Er_dif, diffusion::fld_limiter, geom::geom_dir, heat_capacity(), cl::icl, parameters::ndim, parameters::nx, parameters::nxmax, parameters::nxmin, var_loc::rholoc, varold::rhoold, var_loc::rhouloc, varold::rhouold, var_loc::Tloc, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by diffusion_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.70 rad_transfer/fld/diffusion_imp_gc.f90 File Reference

Contains subroutines diffusion_imp(), mat_prod_dif() and cmp_precond_dif().

Data Types

- interface diffusion_imp__interface

Functions/Subroutines

- subroutine diffusion_imp

  This routine solves the FLD equations implicitely with a conjugate gradient method.

- subroutine mat_prod_dif (T, AT)

  This routine computes the matric product $A * X$.

- subroutine cmp_precond_dif (RR, ZZ)

  This routine computes the preconditioned matrix.

9.70.1 Detailed Description

Contains subroutines diffusion_imp(), mat_prod_dif() and cmp_precond_dif().

Definition in file diffusion_imp_gc.f90.

9.70.2 Function Documentation

9.70.2.1 subroutine cmp_precond_dif (real,dimension(nxmin(1) *RR*, real,dimension(nxmin(1) *ZZ*)

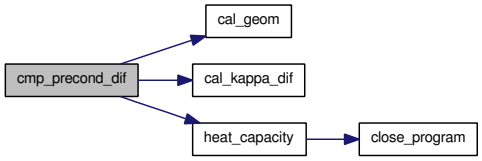
This routine computes the preconditioned matrix.

Definition at line 313 of file diffusion_imp_gc.f90.

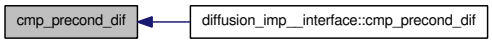
References var::B, unites::c, cal_geom(), cal_kappa_dif(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, diffusion::fld_limiter, geom::geom_dir, heat_capacity(), parameters::ndim, parameters::nx, var::rho, var::rhou, var_loc::Tloc, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by diffusion_imp__interface::cmp_precond_dif().

Here is the call graph for this function:



Here is the caller graph for this function:



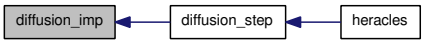
9.70.2.2 subroutine diffusion_imp ()

This routine solves the FLD equations implicitely with a conjugate gradient method.

Definition at line 16 of file diffusion_imp_gc.f90.

Referenced by diffusion_step().

Here is the caller graph for this function:



9.70.2.3 subroutine mat_prod_dif (real,dimension(nxmin(1) *T*, real,dimension(nxmin(1) *AT*)

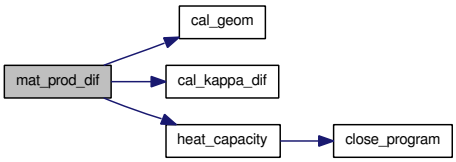
This routine computes the matric product A $*$ X.

Definition at line 153 of file diffusion_imp_gc.f90.

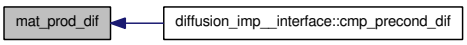
References var::B, unites::c, cal_geom(), cal_kappa_dif(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, var::E, diffusion::fld_limiter, geom::geom_dir, heat_capacity(), parameters::ndim, parameters::nx, var::rho, var::rhou, var_loc::Tloc, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by diffusion_imp__interface::cmp_precond_dif().

Here is the call graph for this function:



Here is the caller graph for this function:



9.71 rad_transfer/fld/diffusion_step.f90 File Reference

Contains subroutine diffusion_step().

Functions/Subroutines

- subroutine diffusion_step (t_fld, t_com, monit)

  This routine performs the evolution over one timestep of the radiative flux limited diffusion.

9.71.1 Detailed Description

Contains subroutine diffusion_step().

Definition in file diffusion_step.f90.

9.71.2 Function Documentation

9.71.2.1 subroutine diffusion_step (real *t_fld*, real *t_com*, logical *monit*)

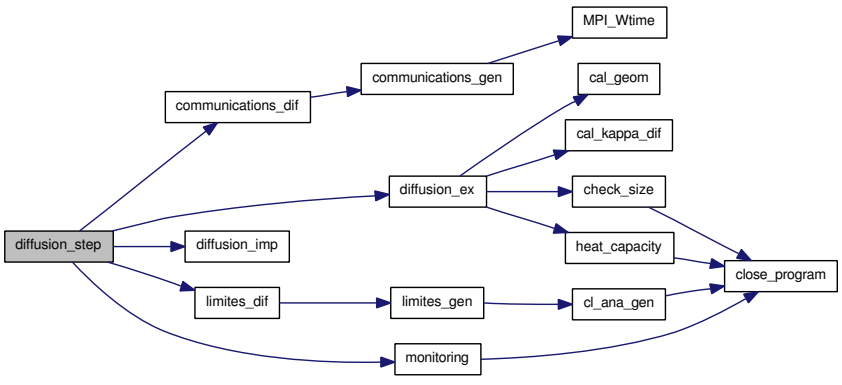
This routine performs the evolution over one timestep of the radiative flux limited diffusion.

Definition at line 15 of file diffusion_step.f90.

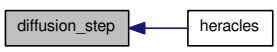
References communications_dif(), diffusion_ex(), diffusion_imp(), limites_dif(), monitoring(), divers::nsx, divers::nsy, divers::nsz, parameters::nx, and divers::verbose.

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:



9.72 rad_transfer/fld/init_diffusion.f90 File Reference

Contains subroutines init_diffusion() and allocate_array_dif() and functions source_dif() and derive_source_dif().

Functions/Subroutines

- subroutine init_diffusion

  Initialization for the FLD routines.

- subroutine allocate_array_dif

  Allocates the arrays used in the FLD calculations.

- real source_dif (Temp)

  Energy ource term for the FLD.

- real derive_source_dif (Temp)

  Derivative of the energy ource term for the FLD.

9.72.1 Detailed Description

Contains subroutines init_diffusion() and allocate_array_dif() and functions source_dif() and derive_source_dif().

Definition in file init_diffusion.f90.

9.72.2 Function Documentation

9.72.2.1 subroutine allocate_array_dif ()

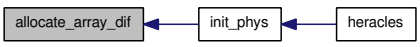
Allocates the arrays used in the FLD calculations.

Definition at line 64 of file init_diffusion.f90.

References diffusion::Er_dif, cl_dif::Er_dif_l, parameters::ndim, parameters::nxmax, and parameters::nxmin.

Referenced by init_phys().

Here is the caller graph for this function:



9.72.2.2 real derive_source_dif (real *Temp*)

Derivative of the energy ource term for the FLD.

Definition at line 109 of file init_diffusion.f90.

9.72.2.3 subroutine init_diffusion ()

Initialization for the FLD routines.

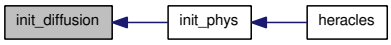
Definition at line 16 of file init_diffusion.f90.

References close_program(), diffusion::Dif_imp, diffusion::fld_limiter, para::mype, diffusion::nitetot_dif, diffusion::precond_dif, diffusion::vardt_dif, and diffusion::varrel_dif.

Referenced by init_phys().

Here is the call graph for this function:

Here is the caller graph for this function:



9.72.2.4 real source_dif (real *Temp*)

Energy ource term for the FLD.

Definition at line 88 of file init_diffusion.f90.

9.73 rad_transfer/fld/kappa_dif.f90 File Reference

Contains subroutine cal_kappa_dif().

Functions/Subroutines

- subroutine cal_kappa_dif (rho, T, kappa, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  This routine computes the FLD opacities.

9.73.1 Detailed Description

Contains subroutine cal_kappa_dif().

Definition in file kappa_dif.f90.

9.73.2 Function Documentation

9.73.2.1 subroutine cal_kappa_dif (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *T*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *kappa*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

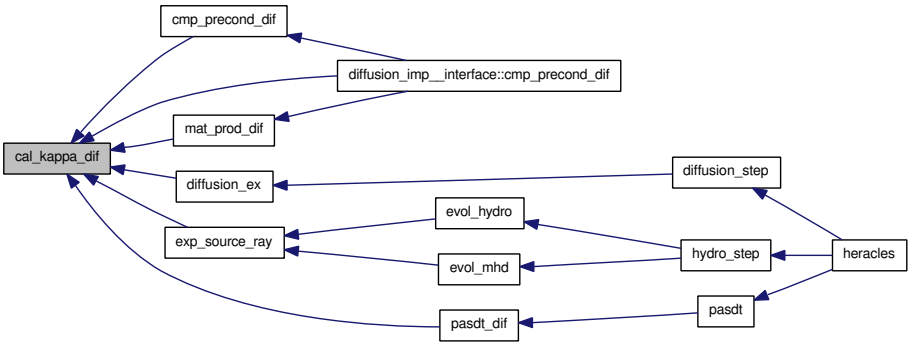
This routine computes the FLD opacities.

Definition at line 14 of file kappa_dif.f90.

References unites::metre.

Referenced by cmp_precond_dif(), diffusion_imp__interface::cmp_precond_dif(), diffusion_ex(), exp_source_ray(), mat_prod_dif(), and pasdt_dif().

Here is the caller graph for this function:



9.74 rad_transfer/fld/limites_dif.f90 File Reference

Contains subroutine limites_dif().

Functions/Subroutines

- subroutine limites_dif

  This routine fills the ghost cells at the domain boundaries using the subroutine limites_gen().

9.74.1 Detailed Description

Contains subroutine limites_dif().

Definition in file limites_dif.f90.

9.74.2 Function Documentation

9.74.2.1 subroutine limites_dif ()

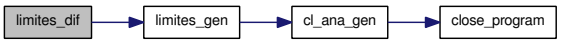
This routine fills the ghost cells at the domain boundaries using the subroutine limites_gen().

Definition at line 16 of file limites_dif.f90.

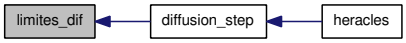
References diffusion::Er_dif, cl_dif::Er_dif_l, cl_dif::icl_dif, limites_gen(), and divers::verbose.

Referenced by diffusion_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.75 rad_transfer/fld/modules_dif.f90 File Reference

Contains modules cl_dif and diffusion.

Modules

- module cl_dif

  Contains the variables for the FLD boundary conditions.

- module diffusion

  Contains the variables for the FLD.

Variables

- integer, dimension(6) cl_dif::icl_dif

  Type of local FLD boundary conditions.

- integer, dimension(6) cl_dif::icl_dif_glob

  Type of global FLD boundary conditions.

- real, dimension(:), allocatable cl_dif::Er_dif_l

  Radiative energy in the boundaries for the FLD.

- real, dimension(:,:,:), allocatable diffusion::Er_dif

  FLD radiative energy.

- real diffusion::varrel_dif

  Maximum variable variations allowed.

- real diffusion::vardt_dif

  Maximum timestep variation allowed.

- real diffusion::dt_dif_exp

  FLD explicit timestep.

- real diffusion::dt_dif_imp

  FLD implicit timestep.

- real diffusion::dt_dif

  FLD timestep.

- character(len=9) diffusion::fld_limiter

  FLD flux limiter.

- logical diffusion::Dif_imp

  Compute FLD implicitely if .true.

- logical diffusion::precond_dif

Perform matrix preconditionning if .true.

- integer diffusion::nitetot_dif

  FLD total number of iterations.

9.75.1 Detailed Description

Contains modules cl_dif and diffusion.

Definition in file modules_dif.f90.

9.76 rad_transfer/fld/para_dif.f90 File Reference

Contains subroutine communications_dif().

Functions/Subroutines

- subroutine communications_dif (tcomm)

  Perform communications and return cputime used.

9.76.1 Detailed Description

Contains subroutine communications_dif().

Definition in file para_dif.f90.

9.76.2 Function Documentation

9.76.2.1 subroutine communications_dif (real *tcomm*)

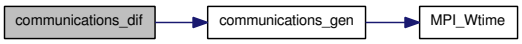
Perform communications and return cputime used.

Definition at line 15 of file para_dif.f90.

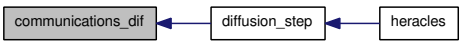
References communications_gen(), and diffusion::Er_dif.

Referenced by diffusion_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.77 rad_transfer/fld/pasdt_dif.f90 File Reference

Contains subroutine pasdt_dif().

Functions/Subroutines

- subroutine pasdt_dif

  This routine computes the FLD timestep.

9.77.1 Detailed Description

Contains subroutine pasdt_dif().

Definition in file pasdt_dif.f90.

9.77.2 Function Documentation

9.77.2.1 subroutine pasdt_dif ()

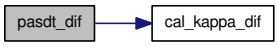
This routine computes the FLD timestep.

Definition at line 14 of file pasdt_dif.f90.

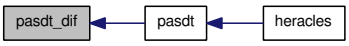
References unites::an, var::B, unites::c, cal_kappa_dif(), diffusion::Dif_imp, diffusion::dt_dif, diffusion::dt_dif_imp, divers::dt_max, var::E, geom::geom_dir, parameters::ndim, parameters::nx, var::rho, var::rhou, divers::verbose, and geom::x.

Referenced by pasdt().

Here is the call graph for this function:



Here is the caller graph for this function:



9.78 rad_transfer/m1/allocate_array_ray.f90 File Reference

Contains subroutine allocate_array_ray().

Functions/Subroutines

- subroutine allocate_array_ray

  Allocates the arrays used for the M1 radiative transfer.

9.78.1 Detailed Description

Contains subroutine allocate_array_ray().

Definition in file allocate_array_ray.f90.

9.78.2 Function Documentation

9.78.2.1 subroutine allocate_array_ray ()

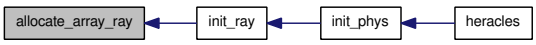
Allocates the arrays used for the M1 radiative transfer.

Definition at line 14 of file allocate_array_ray.f90.

References cl_ray::Er_l, varray::Eray, varray::Eray_t, varray::Erayold, varray::ff_t, cl_ray::Fr_l, varray::Fray, varray::Fray_t, varray::Frayold, varray::kappa_abs_t, parameters::ndim, parameters::nxmax, parameters::nxmin, varray::sigma_diff, varold::Tgazold, and cl_ray::Tr_l.

Referenced by init_ray().

Here is the caller graph for this function:



9.79 rad_transfer/m1/cal_b3d.f90 File Reference

Contains subroutine cal_b3d().

Functions/Subroutines

- subroutine cal_b3d

  Computes the B member in 3D for the GMRES inversion.

9.79.1 Detailed Description

Contains subroutine cal_b3d().

Definition in file cal_b3d.f90.

9.79.2 Function Documentation

9.79.2.1 subroutine cal_b3d ()

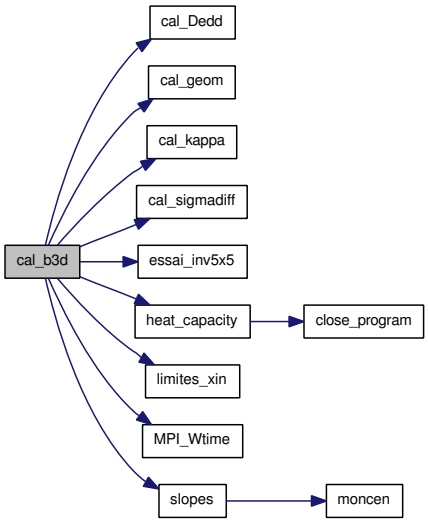
Computes the B member in 3D for the GMRES inversion. Note: equations are written for alpha = 1.

Definition at line 15 of file cal_b3d.f90.

References divers_ray::alpha, unites::c, unites::c2, cal_Dedd(), cal_geom(), cal_kappa(), cal_sigmadiff(), divers_ray::cal_valp, geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, essai_inv5x5(), geom::geom_dir, heat_capacity(), cl_ray::icl_ray, varray::kappa_abs_t, limites_xin(), MPI_Wtime(), parameters::nx, divers_ray::rad_trans_model, var::rho, varray::sigma_diff, divers_ray::slope_type_ray, slopes(), cputime::temps_comm, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by evol_ray().

Here is the call graph for this function:



Here is the caller graph for this function:

9.80 rad_transfer/m1/cal_x3d.f90 File Reference

Contains subroutines cal_x3d(), calmoinsun_x3d() and allocate_gmres().

Functions/Subroutines

- subroutine cal_x3d

  Computes the X member of the GMRES inversion algorithm.

- subroutine calmoinsun_x3d

  Computes the X member of the GMRES inversion algorithm.

- subroutine allocate_gmres

  Allocates the arrays required by the GMRES inversion algorithm.

9.80.1 Detailed Description

Contains subroutines cal_x3d(), calmoinsun_x3d() and allocate_gmres().

Definition in file cal_x3d.f90.

9.80.2 Function Documentation

9.80.2.1 subroutine allocate_gmres ()

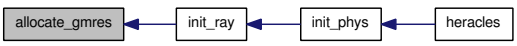
Allocates the arrays required by the GMRES inversion algorithm.

Definition at line 109 of file cal_x3d.f90.

References parameters::nvar_ray, and parameters::nx.

Referenced by init_ray().

Here is the caller graph for this function:



9.80.2.2 subroutine cal_x3d ()

Computes the X member of the GMRES inversion algorithm.

Definition at line 15 of file cal_x3d.f90.

References unites::c, varray::Eray, varray::Erayold, varray::Fray, varray::Frayold, parameters::ndim, parameters::nvar_ray, parameters::nx, var::Tgaz, and varold::Tgazold.

Referenced by evol_ray().

Here is the caller graph for this function:

9.80.2.3 subroutine calmoinsun_x3d ()

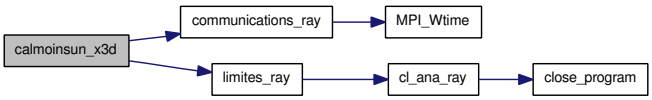
Computes the X member of the GMRES inversion algorithm.

Definition at line 57 of file cal_x3d.f90.

References unites::c, communications_ray(), varray::Eray, varray::Erayold, varray::Fray, varray::Frayold, limites_ray(), parameters::ndim, parameters::nvar_ray, parameters::nx, cputime::temps_comm, var::Tgaz, and varold::Tgazold.

Referenced by evol_ray().

Here is the call graph for this function:



Here is the caller graph for this function:

9.81 rad_transfer/m1/cl_ana_ray.f90 File Reference

Contains subroutines cl_ana_ray() and cl_ana_gen_ray().

Functions/Subroutines

- subroutine cl_ana_ray (iface)

  Analytical boundary conditions for M1 radiative transfer.

- subroutine cl_ana_gen_ray (iface, Tab, Tab_l, n)

  Generic analytical boundary conditions for M1 radiative transfer.

9.81.1 Detailed Description

Contains subroutines cl_ana_ray() and cl_ana_gen_ray().

Definition in file cl_ana_ray.f90.

9.81.2 Function Documentation

9.81.2.1 subroutine cl_ana_gen_ray (integer *iface*, real *Tab*, real *Tab_l*, integer *n*)

Generic analytical boundary conditions for M1 radiative transfer.

Definition at line 37 of file cl_ana_ray.f90.

References close_program(), and para::mype.

Here is the call graph for this function:

9.81.2.2 subroutine cl_ana_ray (integer *iface*)

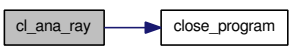
Analytical boundary conditions for M1 radiative transfer.

Definition at line 14 of file cl_ana_ray.f90.

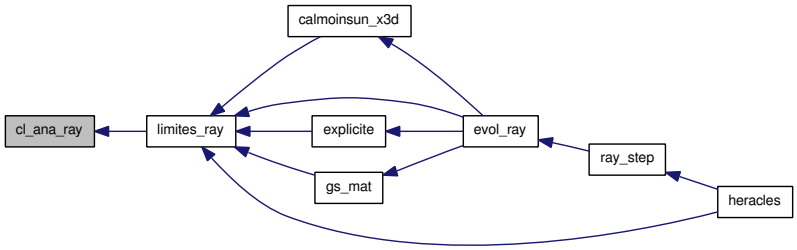
References close_program(), and para::mype.

Referenced by limites_ray().

Here is the call graph for this function:



Here is the caller graph for this function:



9.82 rad_transfer/m1/evol_ray.f90 File Reference

Contains subroutine evol_ray().

Functions/Subroutines

- subroutine evol_ray

  Performs the evolution over one timestep of the radiative energy and flux in the case of M1 radiative transfer.

- subroutine pente_minmod (xx, pxx, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Computes minmod averaged slopes for an arbitrary array of variables.

9.82.1 Detailed Description

Contains subroutine evol_ray().

Definition in file evol_ray.f90.

9.82.2 Function Documentation

9.82.2.1 subroutine evol_ray ()

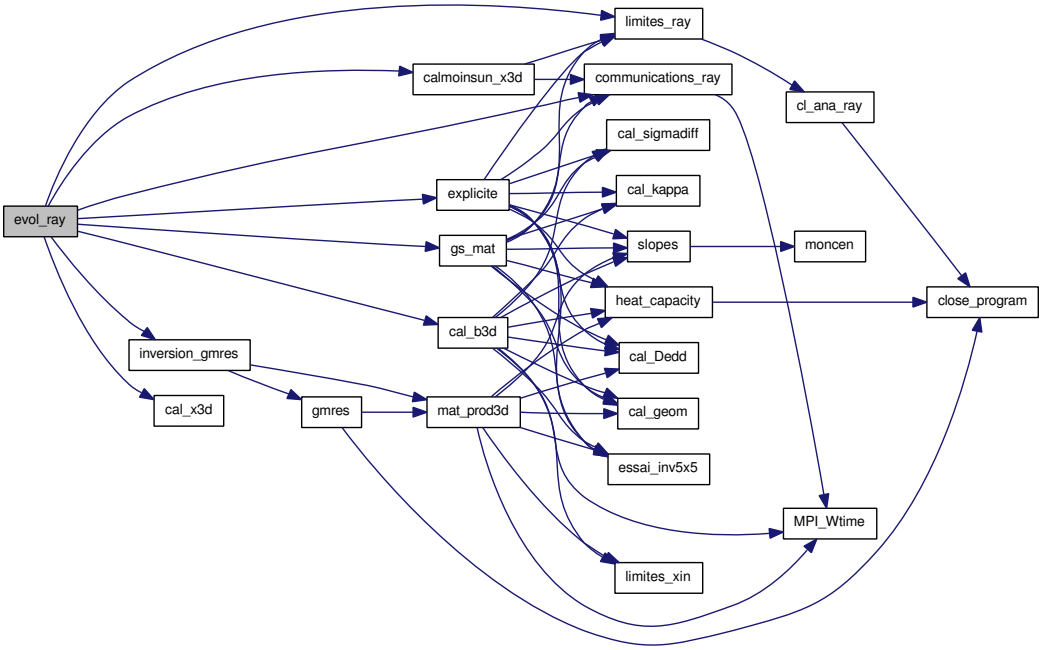
Performs the evolution over one timestep of the radiative energy and flux in the case of M1 radiative transfer. It computes the evolution using either an explicit method or an implicit method. The simulations will usually always start using the explicit method. Once the implicit timestep is a factor of 'alpha_explicite' (cf. namelist) greater than the explicit timstep, the code switches to implicit resolution.

Definition at line 21 of file evol_ray.f90.

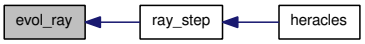
References var::B, cal_b3d(), cal_x3d(), calmoinsun_x3d(), communications_ray(), divers_ray::deja_converge, divers::dt, divers::dtold, var::E, varray::Eray, varray::Erayold, divers_ray::EXP_RAY, explicite(), varray::Fray, varray::Frayold, gs_mat(), inversion_gmres(), limites_ray(), para::mype, parameters::nxmax, parameters::nxmin, var::rho, var::rhou, cputime::temps_comm, var::Tgaz, and varold::Tgazold.

Referenced by ray_step().

Here is the call graph for this function:



Here is the caller graph for this function:



9.82.2.2 subroutine pente_minmod (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *xx*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax,1:ndim) *pxx*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

Computes minmod averaged slopes for an arbitrary array of variables.

Definition at line 137 of file evol_ray.f90.

References geom::dxc, and parameters::ndim.

9.83 rad_transfer/m1/explicite.f90 File Reference

Contains subroutine explicite().

Functions/Subroutines

- subroutine explicite

  Performs explicit resolution of one timestep for radiative transfer.

9.83.1 Detailed Description

Contains subroutine explicite().

Definition in file explicite.f90.

9.83.2 Function Documentation

9.83.2.1 subroutine explicite ()

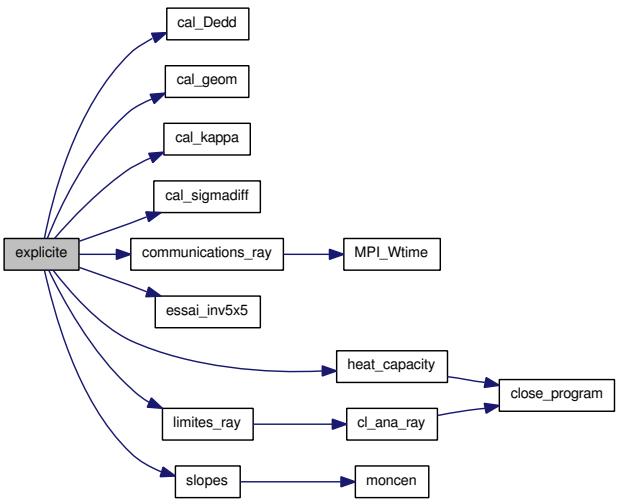
Performs explicit resolution of one timestep for radiative transfer.

Definition at line 15 of file explicite.f90.

References unites::c, unites::c2, cal_Dedd(), cal_geom(), cal_kappa(), cal_sigmadiff(), divers_ray::cal_valp, communications_ray(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, varray::Eray, varray::Erayold, essai_inv5x5(), varray::Fray, varray::Frayold, geom::geom_dir, heat_capacity(), varray::kappa_abs_t, limites_ray(), para::mype, parameters::ndim, parameters::nvar_ray, parameters::nx, parameters::nxmax, parameters::nxmin, divers_ray::rad_trans_model, var::rho, unites::seconde, varray::sigma_diff, divers_ray::slope_type_ray, slopes(), cputime::temps_comm, var::Tgaz, varold::Tgazold, divers::verbose, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by evol_ray().

Here is the call graph for this function:



Here is the caller graph for this function:

9.84 rad_transfer/m1/gmres.f90 File Reference

Contains subroutine gmres().

Functions/Subroutines

- subroutine gmres (n, m, nloc, work, nx, ny, nz, nvar_ray, imin, imax, jmin, jmax, kmin, kmax, workred, lwork, cntl, icntl, info)

  Performs gmres implicit resolution of one timestep for radiative transfer.

9.84.1 Detailed Description

Contains subroutine gmres().

Definition in file gmres.f90.

9.84.2 Function Documentation

9.84.2.1 subroutine gmres (integer,intent(in) *n*, integer,intent(in) *m*, integer,intent(in) *nloc*, real,dimension(lwork) *work*, integer,intent(in) *nx*, integer,intent(in) *ny*, integer,intent(in) *nz*, integer,dimension(-nvar_ray+1:(imax-imin+(jmax-jmin),intent(in) *nvar_ray*, integer,intent(in) *imin*, integer,intent(in) *imax*, integer,intent(in) *jmin*, integer,intent(in) *jmax*, integer,intent(in) *kmin*, integer,dimension(jmax-jmin+1),intent(in) *kmax*, real,dimension(-nvar_ray+1:(imax-imin+(jmax-jmin) *workred*, integer,intent(in) *lwork*, real,dimension(5) *cntl*, integer,dimension(7) *icntl*, integer,dimension(3) *info*)

Performs gmres implicit resolution of one timestep for radiative transfer. n : total problem size m : restart nloc : local problem size work (cf doc gmres) : workspace array for gmres routine work(1 : nloc) : solution or initial estimate work(1+nloc:2∗nloc) : second term

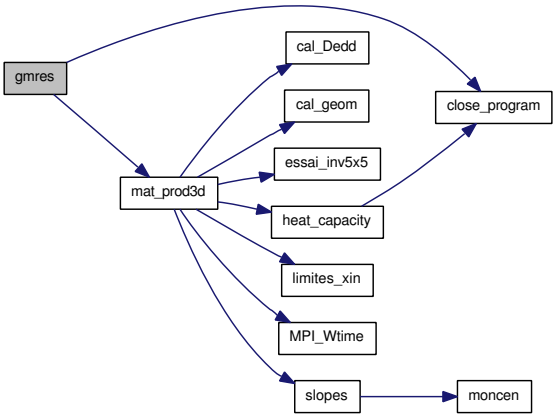
workred : workspace array with overlap zones lwork : size of work cntl, icntl : parameter arrays for gmres (cf doc gmres)

Definition at line 27 of file gmres.f90.

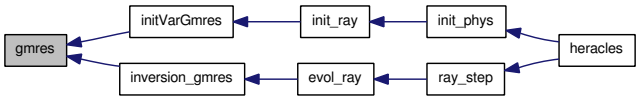
References close_program(), mat_prod3d(), and para::mype.

Referenced by initVarGmres(), and inversion_gmres().

Here is the call graph for this function:



Here is the caller graph for this function:



9.85 rad_transfer/m1/gs.f90 File Reference

Contains subroutines gs_mat(), cal_Dedd() and essai_inv5x5 and functions det_3x3() and det_4x4().

Functions/Subroutines

- subroutine gs_mat

  Performs implicit resolution of one timestep for radiative transfer using the Gauss-Seidel algorithm.

- subroutine cal_Dedd (E, F, Dedd, Dedd_dE, Dedd_dF, Derive_chi)

  Computes the Eddington tensor Dedd and its derivatives with respect to Er and Fr, Dedd_dE and Dedd_dF, respectively.

- subroutine essai_inv5x5 (mat, inv)

  Inverts a 5x5 matrix.

- real det_3x3 (a, b, c, d, e, f, g, h, i)

  Computes the determinant of a 3x3 matrix.

- real det_4x4 (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p)

  Computes the determinant of a 4x4 matrix.

9.85.1 Detailed Description

Contains subroutines gs_mat(), cal_Dedd() and essai_inv5x5 and functions det_3x3() and det_4x4().

Definition in file gs.f90.

9.85.2 Function Documentation

9.85.2.1 subroutine cal_Dedd (real *E*, real,dimension(1:ndim) *F*, real,dimension(3,3) *Dedd*, real,dimension(3,3) *Dedd_dE*, real,dimension(3,3,1:ndim) *Dedd_dF*, logical *Derive_chi*)

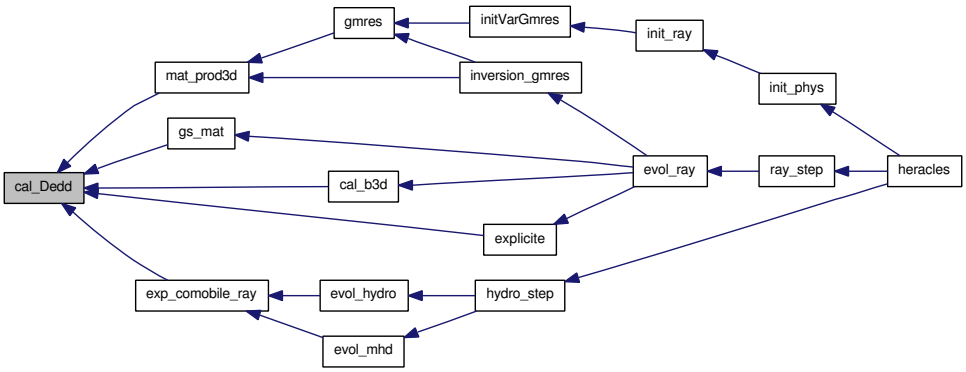
Computes the Eddington tensor Dedd and its derivatives with respect to Er and Fr, Dedd_dE and Dedd_dF, respectively.

Definition at line 1436 of file gs.f90.

References unites::c, parameters::ndim, and divers_ray::rad_trans_model.

Referenced by cal_b3d(), exp_comobile_ray(), explicite(), gs_mat(), and mat_prod3d().

Here is the caller graph for this function:



9.85.2.2 real det_3x3 (real *a*, real *b*, real *c*, real *d*, real *e*, real *f*, real *g*, real *h*, real *i*)

Computes the determinant of a 3x3 matrix.

Definition at line 1647 of file gs.f90.

9.85.2.3 real det_4x4 (real *a*, real *b*, real *c*, real *d*, real *e*, real *f*, real *g*, real *h*, real *i*, real *j*, real *k*, real *l*, real *m*, real *n*, real *o*, real *p*)

Computes the determinant of a 4x4 matrix.

Definition at line 1666 of file gs.f90.

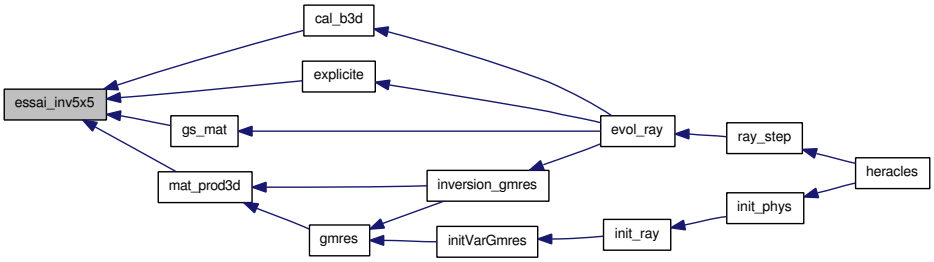
9.85.2.4 subroutine essai_inv5x5 (real,dimension(1:5,1:5) *mat*, real,dimension(1:5,1:5) *inv*)

Inverts a 5x5 matrix.

Definition at line 1580 of file gs.f90.

Referenced by cal_b3d(), explicite(), gs_mat(), and mat_prod3d().

Here is the caller graph for this function:



9.85.2.5 subroutine gs_mat ()

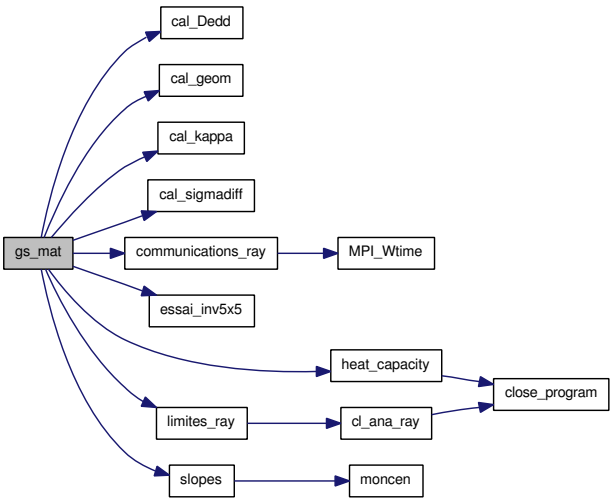
Performs implicit resolution of one timestep for radiative transfer using the Gauss-Seidel algorithm.

Definition at line 17 of file gs.f90.

References divers_ray::alpha, unites::c, unites::c2, cal_Dedd(), cal_geom(), cal_kappa(), cal_sigmadiff(), communications_ray(), geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, varray::Eray, varray::Erayold, essai_inv5x5(), varray::Fray, varray::Frayold, geom::geom_dir, heat_capacity(), cl_ray::icl_ray, divers::iout, varray::kappa_abs_t, limites_ray(), para::mype, parameters::ndim, divers_ray::nitemax_ray, parameters::nvar_ray, parameters::nx, parameters::nxmax, parameters::nxmin, divers_ray::rad_trans_model, var::rho, unites::seconde, varray::sigma_diff, divers_ray::slope_type_ray, slopes(), divers::temps, cputime::temps_comm, var::Tgaz, varold::Tgazold, divers::tout, divers::verbose, geom::x, para::x_start, geom::xcloc, and var_loc::xloc.

Referenced by evol_ray().

Here is the call graph for this function:



Here is the caller graph for this function:

9.86 rad_transfer/m1/init_ray.f90 File Reference

Contains subroutine init_ray().

Functions/Subroutines

- subroutine init_ray

  Initialises all the M1 radiative transfer parameters and variables.

9.86.1 Detailed Description

Contains subroutine init_ray().

Definition in file init_ray.f90.

9.86.2 Function Documentation

9.86.2.1 subroutine init_ray ()

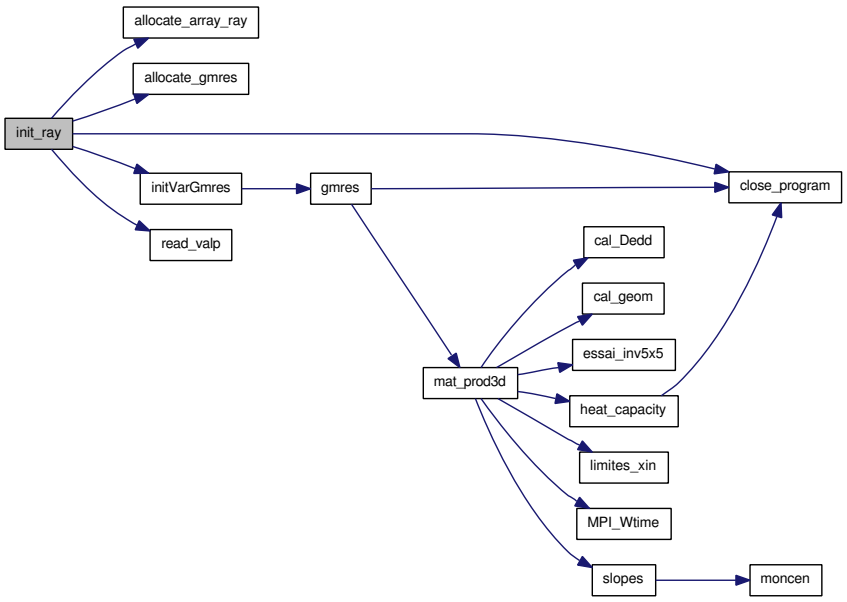
Initialises all the M1 radiative transfer parameters and variables.

Definition at line 15 of file init_ray.f90.

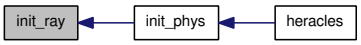
References allocate_array_ray(), allocate_gmres(), divers_ray::alpha, divers_ray::cal_valp, close_program(), initVarGmres(), para::mype, parameters::ndim, divers_ray::nitemax_ray, divers_ray::ordre_des_faces_pour_le_ray, divers_ray::rad_trans_model, read_valp(), divers_ray::slope_type_ray, divers_ray::varrel_gaz, and divers_ray::varrel_ray.

Referenced by init_phys().

Here is the call graph for this function:



Here is the caller graph for this function:



9.87 rad_transfer/m1/initvargmres.f90 File Reference

Contains subroutine initVarGmres().

Functions/Subroutines

- subroutine initVarGmres ()

  Initialises all the M1 parameters and variables used by the GMRES matrix inversion algorithm.

9.87.1 Detailed Description

Contains subroutine initVarGmres().

Definition in file initvargmres.f90.

9.87.2 Function Documentation

9.87.2.1 subroutine initVarGmres ()

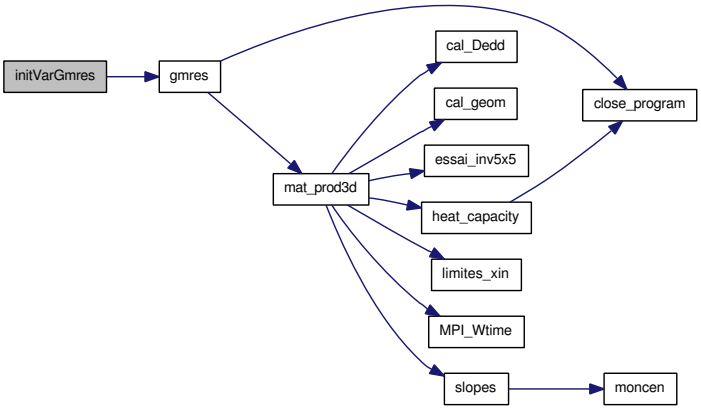
Initialises all the M1 parameters and variables used by the GMRES matrix inversion algorithm.

Definition at line 15 of file initvargmres.f90.

References gmres(), parameters::nvar_ray, and parameters::nx.

Referenced by init_ray().

Here is the call graph for this function:



Here is the caller graph for this function:

9.88 rad_transfer/m1/inversion_gmres.f90 File Reference

Contains subroutine inversion_gmres().

Functions/Subroutines

- subroutine inversion_gmres

  Performs the GMRES matrix inversion.

9.88.1 Detailed Description

Contains subroutine inversion_gmres().

Definition in file inversion_gmres.f90.

9.88.2 Function Documentation

9.88.2.1 subroutine inversion_gmres ()

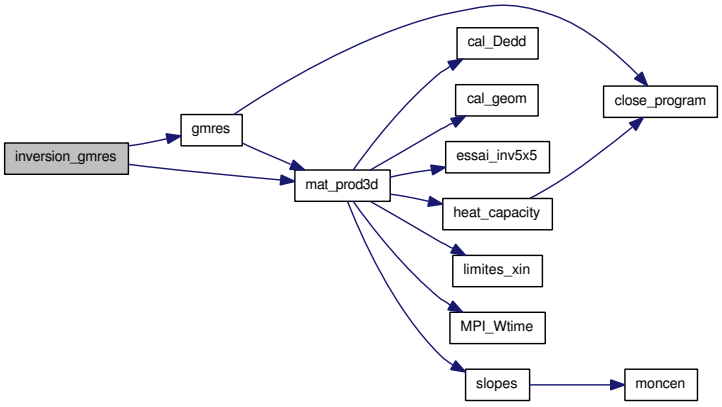
Performs the GMRES matrix inversion.

Definition at line 14 of file inversion_gmres.f90.

References divers_ray::deja_converge, gmres(), mat_prod3d(), para::mype, parameters::nvar_ray, parameters::nx, and parameters::nx_glob.

Referenced by evol_ray().

Here is the call graph for this function:



Here is the caller graph for this function:

9.89 rad_transfer/m1/kappa_ray.f90 File Reference

Contains subroutines cal_kappa() and cal_sigmadiff().

Functions/Subroutines

- subroutine cal_kappa (rho, Tgaz, kappa, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Computes the opacity κ.

- subroutine cal_sigmadiff (rho, Tgaz, sigma, nxmin, nxmax, nymin, nymax, nzmin, nzmax)

  Computes the scattering coefficient σ_s.

9.89.1 Detailed Description

Contains subroutines cal_kappa() and cal_sigmadiff().

Definition in file kappa_ray.f90.

9.89.2 Function Documentation

9.89.2.1 subroutine cal_kappa (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *Tgaz*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *kappa*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

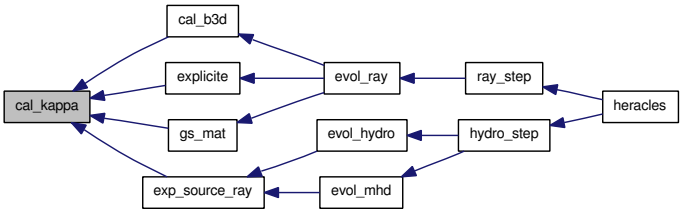
Computes the opacity κ.

Definition at line 14 of file kappa_ray.f90.

References unites::metre, and divers_ray::rho1.

Referenced by cal_b3d(), exp_source_ray(), explicite(), and gs_mat().

Here is the caller graph for this function:



9.89.2.2 subroutine cal_sigmadiff (real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *rho*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *Tgaz*, real,dimension(nxmin:nxmax,nymin:nymax,nzmin:nzmax) *sigma*, integer *nxmin*, integer *nxmax*, integer *nymin*, integer *nymax*, integer *nzmin*, integer *nzmax*)

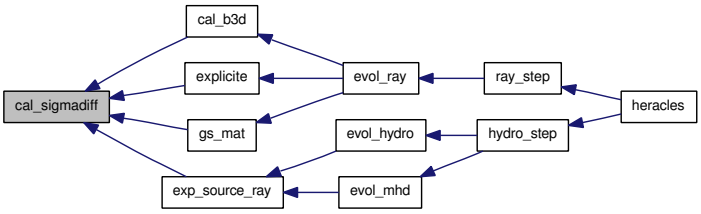
Computes the scattering coefficient σ_s.

Definition at line 40 of file kappa_ray.f90.

References unites::metre.

Referenced by cal_b3d(), exp_source_ray(), explicite(), and gs_mat().

Here is the caller graph for this function:



9.90 rad_transfer/m1/limites_ray.f90 File Reference

Contains subroutine limites_ray().

Functions/Subroutines

- subroutine limites_ray

  This routine fills the ghost cells at the domain boundaries with the appropriate radiative transfer values according to the type of boudary condition.

9.90.1 Detailed Description

Contains subroutine limites_ray().

Definition in file limites_ray.f90.

9.90.2 Function Documentation

9.90.2.1 subroutine limites_ray ()

This routine fills the ghost cells at the domain boundaries with the appropriate radiative transfer values according to the type of boudary condition. Type of boundaries:

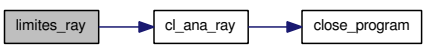
- 1 : free-flow
- 2 : periodic
- 3 : reflexive
- 4 : imposed values
- 5 : analytical

Definition at line 23 of file limites_ray.f90.

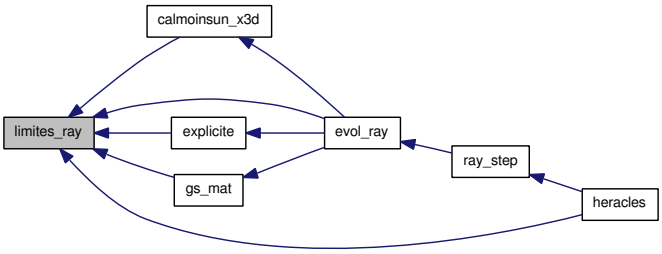
References cl_ana_ray(), cl_ray::Er_l, varray::Eray, cl_ray::Fr_l, varray::Fray, parameters::Nbuf, parameters::ndim, parameters::nx, parameters::nxmax, parameters::nxmin, divers_ray::ordre_des_faces_pour_le_ray, cl::T_l, var::Tgaz, and divers::verbose.

Referenced by calmoinsun_x3d(), evol_ray(), explicite(), gs_mat(), and heracles().

Here is the call graph for this function:



Here is the caller graph for this function:



9.91 rad_transfer/m1/mat_prod3d.f90 File Reference

Contains subroutines mat_prod3d(), com_1D_xin(), com_2D_xin(), com_3D_xin() and limites_xin().

Functions/Subroutines

- subroutine mat_prod3d (xin, xout)

  Computes the A∗xin product and stocks it in the xout vector.

- subroutine limites_xin (xin)

  Boundary conditions for rthe xin vector.

9.91.1 Detailed Description

Contains subroutines mat_prod3d(), com_1D_xin(), com_2D_xin(), com_3D_xin() and limites_xin().

Definition in file mat_prod3d.f90.

9.91.2 Function Documentation

9.91.2.1 subroutine limites_xin (real,dimension(-nvar_ray+1:(imax-imin+(jmax-jmin) *xin*)

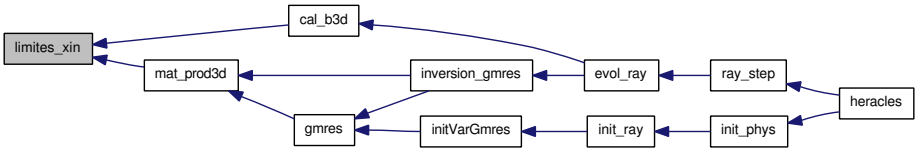
Boundary conditions for rthe xin vector.

Definition at line 1717 of file mat_prod3d.f90.

References unites::c, cl_ray::Er_l, cl_ray::Fr_l, parameters::ndim, parameters::nx, and divers::verbose.

Referenced by cal_b3d(), and mat_prod3d().

Here is the caller graph for this function:



9.91.2.2 subroutine mat_prod3d (real,dimension(-nvar_ray+1:(imax-imin+(jmax-jmin) *xin*, real,dimension(1: nx(1) *xout*)

Computes the A∗xin product and stocks it in the xout vector. In order to treat the boundary conditions properly, the xin and xout vectors have a ghost cell on each side.

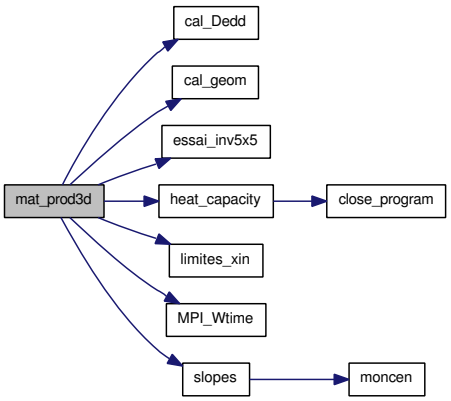
Moreover, the stocking formula for the xin and xout vectors is: x((i-imin+(j-jmin)∗(imax-imin+1)+(k-kmin)∗(imax-imin+1)∗(jmax-jmin+1)-1)∗nvar_ray+ivar) corresponds to the value of the variable ivar (1,2,3,4,5 for Tgaz, Eray, Frayx, Frayy, Frayz respectively) in the (i,j,k) cell.

Definition at line 26 of file mat_prod3d.f90.

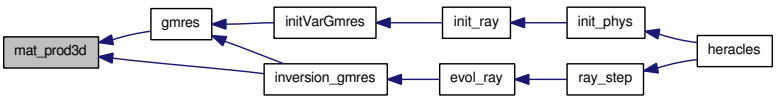
References divers_ray::alpha, communication::ar13_e, communication::ar13_r, communication::ar14_-e, communication::ar14_r, communication::ar15_e, communication::ar15_r, communication::ar16_e, communication::ar16_r, communication::ar23_e, communication::ar23_r, communication::ar24_e, communication::ar24_r, communication::ar25_e, communication::ar25_r, communication::ar26_e, communication::ar26_r, communication::ar35_e, communication::ar35_r, communication::ar36_e, communication::ar36_r, communication::ar45_e, communication::ar45_r, communication::ar46_-e, communication::ar46_r, unites::c, unites::c2, cal_Dedd(), cal_geom(), divers_ray::cal_-valp, communication::coin135_e, communication::coin135_r, communication::coin136_-e, communication::coin136_r, communication::coin13_e, communication::coin13_-r, communication::coin145_e, communication::coin145_r, communication::coin146_-e, communication::coin146_r, communication::coin14_e, communication::coin14_-r, communication::coin235_e, communication::coin235_r, communication::coin236_-e, communication::coin236_r, communication::coin23_e, communication::coin23_-r, communication::coin245_e, communication::coin245_r, communication::coin246_e, communication::coin246_r, communication::coin24_e, communication::coin24_r, geom::ds, divers::dt, geom::dv, geom::dx, geom::dxc, essai_inv5x5(), communication::face1_e, communication::face1_r, communication::face2_e, communication::face2_r, communication::face3_e, communication::face3_r, communication::face4_e, communication::face4_r, communication::face5_e, communication::face5_r, communication::face6_e, communication::face6_r, geom::geom_dir, heat_capacity(), cl_ray::icl_-ray, varray::kappa_abs_t, limites_xin(), MPI_Wtime(), parameters::nx, divers_ray::rad_trans_model, var::rho, varray::sigma_diff, divers_ray::slope_type_ray, slopes(), cputime::temps_comm, divers::verbose, para::voisin, geom::x, geom::xcloc, and var_loc::xloc.

Referenced by gmres(), and inversion_gmres().

Here is the call graph for this function:



Here is the caller graph for this function:



9.92 rad_transfer/m1/modules_ray.f90 File Reference

Contains modules varray, divers_ray, cl_ray, valeurs_propres, TabGmres, VarGmres and interfaces.

Data Types

- interface TabGmres::interface

  GMRES interfaces.

Modules

- module varray

  Contains the M1 radiative transfer variables.

- module divers_ray

  Contains various variables for the M1 radiative transfer.

- module cl_ray

  Contains the boundary conditions variables for the M1 radiative transfer.

- module valeurs_propres

  Contains the array to hold the tabulated M1 eignevalues.

- module TabGmres

  Contains workspace arrays for GMRES.

Variables

- real, dimension(:,:,:), allocatable varray::Eray

  Radiative energy.

- real, dimension(:,:,:), allocatable varray::Erayold

  Old radiative energy.

- real, dimension(:,:,:), allocatable varray::Eray_t

  Radiative energy at time $t + 0.5dt$.

- real, dimension(:,:,:,:), allocatable varray::Fray

  Radiative flux.

- real, dimension(:,:,:,:), allocatable varray::Frayold

  Old radiative flux.

- real, dimension(:,:,:,:), allocatable varray::Fray_t

  Radiative flux at time $t + 0.5dt$.

- real, dimension(:,:,:), allocatable varray::ff_t

  real, dimension(:,:,:), allocatable :: kappa_em_t !< Emission opacity $\kappa_{\rm em}$

- real, dimension(:,:,:), allocatable varray::kappa_abs_t

  Absorption opacity $\kappa_{\rm abs}$.

- real, dimension(:,:,:), allocatable varray::sigma_diff

  Scattering coefficient σ_s.

- integer, dimension(6) divers_ray::ordre_des_faces_pour_le_ray

  integer :: nsx !<

- integer divers_ray::nsy

  integer :: nsz !<

- integer divers_ray::slope_type_ray

  Type of slope limiter for radiative transfer.

- integer divers_ray::nitetot_ray

  Total number of iterations in the implicit resolution.

- integer divers_ray::nitemax_ray

  Maximum number of iterations per timestep allowed in the implicit resolution.

- real divers_ray::varrel_ray

  Maximum quantity variation allowed in an implicit timestep.

- real divers_ray::alpha

  real :: alpha_explicite !<

- real divers_ray::varrel_gaz

  real :: dt_ray_imp !< Implicit radiative transfer timestep

- real divers_ray::dt_ray_exp

  Explicit radiative transfer timestep.

- real divers_ray::dt_ray_old

  Old radiative transfer timestep.

- real divers_ray::dt_ray

  Radiative transfer timestep.

- real divers_ray::dt_ray_glob

  Global radiative transfer timestep.

- real divers_ray::rho1

  Density used to compute opacity κ.

- real divers_ray::T1

  Temperature used to compute opacity κ.

- real divers_ray::alpha_var

  real :: valp_min !< Minimum allowed value for eigenvalues

- logical divers_ray::IMP_RAY

  Solves radiative transfer implicitely if .true.

- logical divers_ray::EXP_RAY

  Solves radiative transfer explicitely if .true.

- logical divers_ray::rankine

  logical :: precond_ray !< Performs matrix preconditionning if .true.

- logical divers_ray::cal_valp

  Computes M1 eigenvalues if .true.

- logical divers_ray::deja_converge

  character (len=10) :: matrix_inv_solver !< Type of implicit solver ('GS', 'GMRES', 'GS-GMRES', 'GMRES-GS' or 'yoyo')

- character(len=2) divers_ray::rad_trans_model

  Type of radiative transfer model: 'M1' or 'P1'.

- integer, dimension(6) cl_ray::icl_ray

  Type of radiative boundary conditions.

- integer, dimension(6) cl_ray::icl_ray_glob

  Type of global radiative boundary conditions.

- real, dimension(:), allocatable cl_ray::Er_l

  Radiative energy imposed in the ghost cells.

- real, dimension(:), allocatable cl_ray::Tr_l

  Gas temperature imposed in the ghost cells.

- real, dimension(:,:), allocatable cl_ray::Fr_l

  Radiative flux imposed in the ghost cells.

- integer valeurs_propres::n_points

  Number of points in the tabulated eigenvalues curve.

- real, dimension(:,:,:,:), allocatable valeurs_propres::valp

  Array to hold the tabulated eigenvalues as a function of θ and ϵ.

- integer TabGmres::imin

  integer :: imax !<

- integer TabGmres::jmin

  integer :: jmax !<

- integer TabGmres::kmin
  integer :: kmax !<
- real, dimension(:), allocatable TabGmres::work
  real, dimension(:), allocatable :: workred !<
- real, dimension(:), allocatable TabGmres::xin
  real, dimension(:), allocatable :: xout !<
- real, dimension(:), allocatable TabGmres::b
  end module TabGmres
- integer, dimension(7) TabGmres::icntl
  Contains the parameters for GMRES.
- integer TabGmres::itermax
  integer :: lwork !<
- real, dimension(5) TabGmres::cntl
  end module VarGmres

9.92.1 Detailed Description

Contains modules varray, divers_ray, cl_ray, valeurs_propres, TabGmres, VarGmres and interfaces.

Definition in file modules_ray.f90.

9.93 rad_transfer/m1/para_ray.f90 File Reference

Contains module communication_ray and subroutines allocate_array_com_ray(), communications_ray(), com_1D_ray(), com_2D_ray() and com_3D_ray().

Modules

- module communication_ray

  Contains the communication arrays for the M1 radiative transfer.

Functions/Subroutines

- subroutine allocate_array_com_ray

  Allocates the communication arrays for the M1 radiative transfer.

- subroutine communications_ray (tcomm)

  Perform communications and return cputime used.

Variables

- real, dimension(:,:,:,:), allocatable communication_ray::face1_e

  real, dimension(:,:,:,:), allocatable :: face2_e !<

- real, dimension(:,:,:,:), allocatable communication_ray::face3_e

  real, dimension(:,:,:,:), allocatable :: face4_e !<

- real, dimension(:,:,:,:), allocatable communication_ray::face5_e

  real, dimension(:,:,:,:), allocatable :: face6_e !<

- real, dimension(:,:,:,:), allocatable communication_ray::face1_r

  real, dimension(:,:,:,:), allocatable :: face2_r !<

- real, dimension(:,:,:,:), allocatable communication_ray::face3_r

  real, dimension(:,:,:,:), allocatable :: face4_r !<

- real, dimension(:,:,:,:), allocatable communication_ray::face5_r

  real, dimension(:,:,:,:), allocatable :: face6_r !<

- real, dimension(:,:,:,:), allocatable communication_ray::ar13_e

  real, dimension(:,:,:,:), allocatable :: ar14_e !<

- real, dimension(:,:,:,:), allocatable communication_ray::ar23_e

  real, dimension(:,:,:,:), allocatable :: ar24_e !<

- real, dimension(:,:,:,:), allocatable communication_ray::ar15_e

  real, dimension(:,:,:,:), allocatable :: ar25_e !<

- real, dimension(:,:,:,:), allocatable communication_ray::ar35_e
  real, dimension(:,:,:,:), allocatable :: ar45_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar16_e
  real, dimension(:,:,:,:), allocatable :: ar26_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar36_e
  real, dimension(:,:,:,:), allocatable :: ar46_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar13_r
  real, dimension(:,:,:,:), allocatable :: ar14_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar23_r
  real, dimension(:,:,:,:), allocatable :: ar24_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar15_r
  real, dimension(:,:,:,:), allocatable :: ar25_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar35_r
  real, dimension(:,:,:,:), allocatable :: ar45_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar16_r
  real, dimension(:,:,:,:), allocatable :: ar26_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::ar36_r
  real, dimension(:,:,:,:), allocatable :: ar46_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin13_e
  real, dimension(:,:,:,:), allocatable :: coin14_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin23_e
  real, dimension(:,:,:,:), allocatable :: coin24_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin13_r
  real, dimension(:,:,:,:), allocatable :: coin14_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin23_r
  real, dimension(:,:,:,:), allocatable :: coin24_r !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin135_e
  real, dimension(:,:,:,:), allocatable :: coin136_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin145_e
  real, dimension(:,:,:,:), allocatable :: coin146_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin235_e
  real, dimension(:,:,:,:), allocatable :: coin236_e !<
- real, dimension(:,:,:,:), allocatable communication_ray::coin245_e
  real, dimension(:,:,:,:), allocatable :: coin246_e !<

- real, dimension(:,:,:,:), allocatable communication_ray::coin135_r

  real, dimension(:,:,:,:), allocatable :: coin136_r !<

- real, dimension(:,:,:,:), allocatable communication_ray::coin145_r

  real, dimension(:,:,:,:), allocatable :: coin146_r !<

- real, dimension(:,:,:,:), allocatable communication_ray::coin235_r

  real, dimension(:,:,:,:), allocatable :: coin236_r !<

- real, dimension(:,:,:,:), allocatable communication_ray::coin245_r

  real, dimension(:,:,:,:), allocatable :: coin246_r !<

9.93.1 Detailed Description

Contains module communication_ray and subroutines allocate_array_com_ray(), communications_ray(), com_1D_ray(), com_2D_ray() and com_3D_ray().

Definition in file para_ray.f90.

9.93.2 Function Documentation

9.93.2.1 subroutine allocate_array_com_ray ()

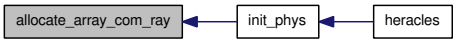
Allocates the communication arrays for the M1 radiative transfer.

Definition at line 91 of file para_ray.f90.

References communication_ray::ar13_e, communication_ray::ar13_r, communication_ray::ar15_e, communication_ray::ar15_r, communication_ray::ar16_e, communication_ray::ar16_r, communication_ray::ar23_e, communication_ray::ar23_r, communication_ray::ar35_e, communication_ray::ar35_r, communication_ray::ar36_e, communication_ray::ar36_r, communication_ray::coin135_e, communication_ray::coin135_r, communication_ray::coin13_e, communication_ray::coin13_r, communication_ray::coin145_e, communication_ray::coin145_r, communication_ray::coin235_e, communication_ray::coin235_r, communication_ray::coin23_e, communication_ray::coin23_r, communication_ray::coin245_e, communication_ray::coin245_r, communication_ray::face1_e, communication_ray::face1_r, communication_ray::face3_e, communication_ray::face3_r, communication_ray::face5_e, communication_ray::face5_r, parameters::Nbuf, parameters::ndim, parameters::nvar_ray, and parameters::nx.

Referenced by init_phys().

Here is the caller graph for this function:



9.93.2.2 subroutine communications_ray (real *tcomm*)

Perform communications and return cputime used.

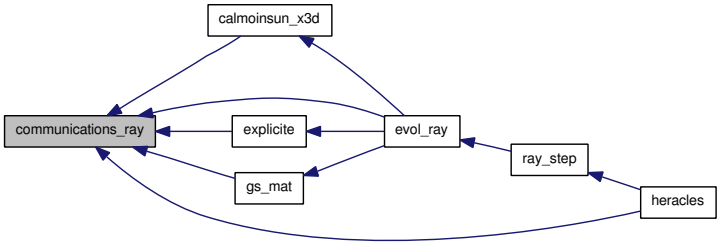
Definition at line 159 of file para_ray.f90.

References communication_ray::ar13_e, communication_ray::ar13_r, communication_ray::ar15_e, communication_ray::ar15_r, communication_ray::ar16_e, communication_ray::ar16_r, communication_ray::ar23_e, communication_ray::ar23_r, communication_ray::ar35_e, communication_ray::ar35_r, communication_ray::ar36_e, communication_ray::ar36_r, communication_ray::coin135_e, communication_ray::coin135_r, communication_ray::coin13_e, communication_ray::coin13_r, communication_ray::coin145_e, communication_ray::coin145_r, communication_ray::coin235_e, communication_ray::coin235_r, communication_ray::coin23_e, communication_ray::coin23_r, communication_ray::coin245_e, communication_ray::coin245_r, varray::Eray, communication_ray::face1_e, communication_ray::face1_r, communication_ray::face3_e, communication_ray::face3_r, communication_ray::face5_e, communication_ray::face5_r, varray::Fray, MPI_Wtime(), parameters::Nbuf, parameters::ndim, parameters::nvar_ray, parameters::nx, var::Tgaz, divers::verbose, and para::voisin.

Referenced by calmoinsun_x3d(), evol_ray(), explicite(), gs_mat(), and heracles().

Here is the call graph for this function:

Here is the caller graph for this function:



9.94 rad_transfer/m1/pasdt_ray.f90 File Reference

Contains subroutines pasdt_ray() and pasdt_ray_exp().

Functions/Subroutines

- subroutine pasdt_ray

  Computes the maximum allowed implicit timestep for the M1 radiative transfer.

- subroutine pasdt_ray_exp

  Computes the maximum allowed explicit timestep for the M1 radiative transfer.

9.94.1 Detailed Description

Contains subroutines pasdt_ray() and pasdt_ray_exp().

Definition in file pasdt_ray.f90.

9.94.2 Function Documentation

9.94.2.1 subroutine pasdt_ray ()

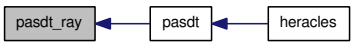
Computes the maximum allowed implicit timestep for the M1 radiative transfer.

Definition at line 15 of file pasdt_ray.f90.

References unites::c, divers::dt_max, divers_ray::dt_ray, divers_ray::dt_ray_exp, divers_ray::dt_ray_glob, divers_ray::dt_ray_old, varray::Eray, varray::Erayold, divers_ray::EXP_RAY, varray::Fray, varray::Frayold, divers_ray::IMP_RAY, para::mype, parameters::ndim, parameters::nx, var::Tgaz, varold::Tgazold, divers_ray::varrel_gaz, divers_ray::varrel_ray, divers::verbose, and para::x_start.

Referenced by pasdt().

Here is the caller graph for this function:



9.94.2.2 subroutine pasdt_ray_exp ()

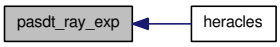
Computes the maximum allowed explicit timestep for the M1 radiative transfer.

Definition at line 223 of file pasdt_ray.f90.

References unites::c, divers_ray::dt_ray_exp, divers_ray::dt_ray_glob, geom::geom_dir, para::mype, parameters::ndim, parameters::nx, unites::seconde, divers::verbose, and geom::x.

Referenced by heracles().

Here is the caller graph for this function:



9.95 rad_transfer/m1/ray_step.f90 File Reference

Contains subroutine ray_step().

Functions/Subroutines

- subroutine ray_step (t_ray, t_comm, monit)

  Performs the evolution over one timstep of the M1 radiative variables.

9.95.1 Detailed Description

Contains subroutine ray_step().

Definition in file ray_step.f90.

9.95.2 Function Documentation

9.95.2.1 subroutine ray_step (real *t_ray*, real *t_comm*, logical *monit*)

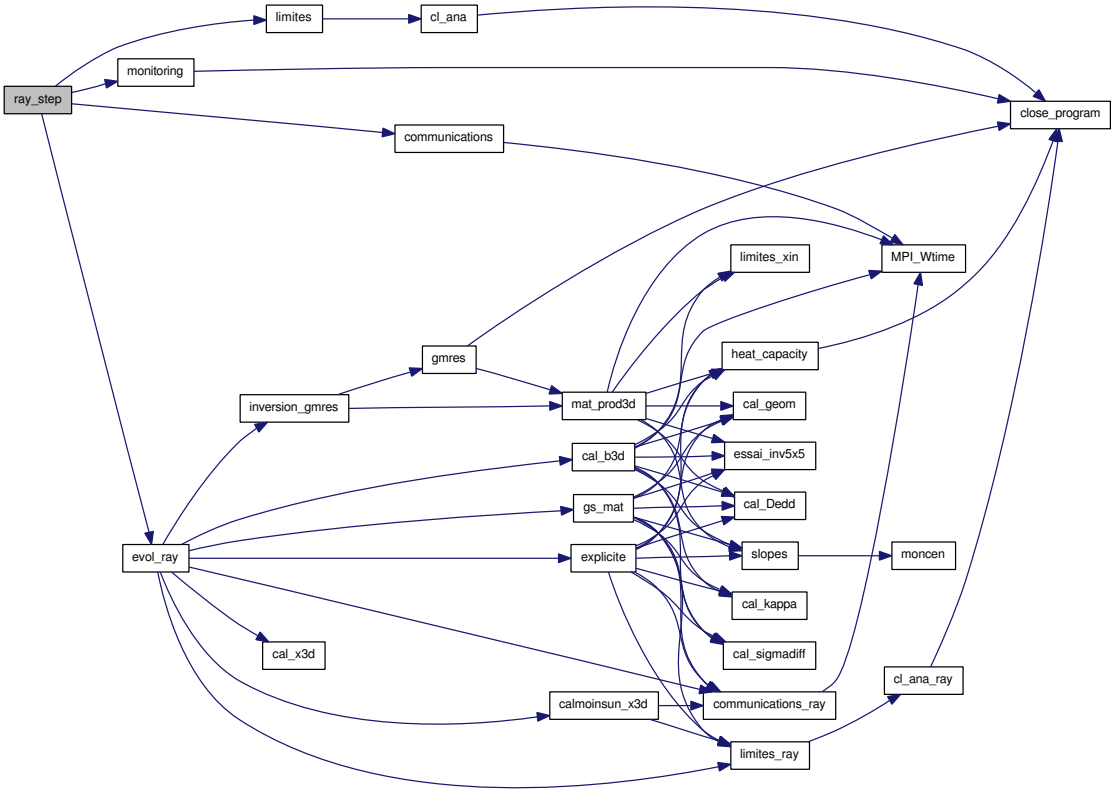
Performs the evolution over one timstep of the M1 radiative variables.

Definition at line 15 of file ray_step.f90.

References communications(), evol_ray(), limites(), and monitoring().

Referenced by heracles().

Here is the call graph for this function:



Here is the caller graph for this function:

9.96 rad_transfer/m1/sources_ray.f90 File Reference

Contains functions source() and derive_source().

Functions/Subroutines

- real source (Temp)

  Computes the energy-matter coupling source term $a_R T^4$.

- real derive_source (Temp)

  Computes the derivative of the energy-matter coupling source term $4a_R T^3$.

9.96.1 Detailed Description

Contains functions source() and derive_source().

Definition in file sources_ray.f90.

9.96.2 Function Documentation

9.96.2.1 real derive_source (real *Temp*)

Computes the derivative of the energy-matter coupling source term $4a_R T^3$.

Definition at line 37 of file sources_ray.f90.

9.96.2.2 real source (real *Temp*)

Computes the energy-matter coupling source term $a_R T^4$.

Definition at line 15 of file sources_ray.f90.

9.97 rad_transfer/m1/valp.f90 File Reference

Contains subroutine read_valp() and function interpol_valp().

Functions/Subroutines

- subroutine read_valp

  Reads the tabulated eigenvalues for the M1 radiative transfer solver.

- real interpol_valp (f, t, e, ivalp)

  Performs interpolation between tabulated point on the eigenvalues curves to find a particular eigenvalue during the simulation.

9.97.1 Detailed Description

Contains subroutine read_valp() and function interpol_valp().

Definition in file valp.f90.

9.97.2 Function Documentation

9.97.2.1 real interpol_valp (real *f*, real *t*, real *e*, integer *ivalp*)

Performs interpolation between tabulated point on the eigenvalues curves to find a particular eigenvalue during the simulation.

Definition at line 71 of file valp.f90.

References valeurs_propres::n_points, and valeurs_propres::valp.

9.97.2.2 subroutine read_valp ()

Reads the tabulated eigenvalues for the M1 radiative transfer solver.

Definition at line 16 of file valp.f90.

References valeurs_propres::n_points, and valeurs_propres::valp.

Referenced by init_ray().

Here is the caller graph for this function:

Index

a_R
    unites, 152
aff.f90
    aff_new, 230
    convtoasc, 231
    make_name, 231
    mk_dir_out, 232
    print_configuration, 232
    sortie_ecran, 232
aff_new
    aff.f90, 230
algebra/ Directory Reference, 13
algebra/cg.f90, 181
allocate_array
    allocate_array.f90, 233
allocate_array.f90
    allocate_array, 233
allocate_array_com
    para.f90, 277
allocate_array_com_gen
    para_generique.f90, 284
allocate_array_com_ray
    para_ray.f90, 364
allocate_array_dif
    init_diffusion.f90, 324
allocate_array_gra
    modules_gra.f90, 197
allocate_array_ray
    allocate_array_ray.f90, 332
allocate_array_ray.f90
    allocate_array_ray, 332
allocate_gmres
    cal_x3d.f90, 335
alpha
    divers, 84
    divers_ray, 95
alpha_var
    divers_ray, 95
an
    unites, 152
ar13_e
    communication, 41
    communication_gen, 55
    communication_ray, 66
ar13_r
    communication, 41
    communication_gen, 55
    communication_ray, 66
ar14_e
    communication, 42
    communication_gen, 55
ar14_r
    communication, 42
    communication_gen, 55
ar15_e
    communication, 42
    communication_gen, 55
    communication_ray, 66
ar15_r
    communication, 42
    communication_gen, 55
    communication_ray, 66
ar16_e
    communication, 42
    communication_gen, 55
    communication_ray, 66
ar16_r
    communication, 42
    communication_gen, 55
    communication_ray, 66
ar23_e
    communication, 42
    communication_gen, 56
    communication_ray, 66
ar23_r
    communication, 43
    communication_gen, 56
    communication_ray, 66
ar24_e
    communication, 43
    communication_gen, 56
ar24_r
    communication, 43
    communication_gen, 56
ar25_e
    communication, 43
    communication_gen, 56
ar25_r
    communication, 43
    communication_gen, 56

ar26_e
communication, 43
communication_gen, 56
ar26_r
communication, 43
communication_gen, 57
ar35_e
communication, 44
communication_gen, 57
communication_ray, 67
ar35_r
communication, 44
communication_gen, 57
communication_ray, 67
ar36_e
communication, 44
communication_gen, 57
communication_ray, 67
ar36_r
communication, 44
communication_gen, 57
communication_ray, 67
ar45_e
communication, 44
communication_gen, 57
ar45_r
communication, 44
communication_gen, 57
ar46_e
communication, 44
communication_gen, 58
ar46_r
communication, 45
communication_gen, 58
Arret
main.f90, 247
athena_roe
godunov_utils.f90, 303

B
var, 164
b
TabGmres, 147
B0
param_ini, 126
B_l
cl, 32
bar
unites, 152
bigreal
const, 74
Bold
varold, 169
box_max
geom, 106
box_min
geom, 106

c
unites, 152
c2
unites, 152
cal_b3d
cal_b3d.f90, 333
cal_b3d.f90
cal_b3d, 333
cal_barycentre
limites_gra.f90, 195
cal_Dedd
gs.f90, 345
cal_dif
cg.f90, 181
cal_gamma
solvers_hydro, 138
cal_geom
geom.f90, 241
cal_histogramme
histogramme.f90, 219
cal_kappa
kappa_ray.f90, 352
cal_kappa_con
kappa_con.f90, 188
cal_kappa_dif
kappa_dif.f90, 326
cal_sigmadiff
kappa_ray.f90, 352
cal_valp
divers_ray, 95
cal_x3d
cal_x3d.f90, 335
cal_x3d.f90
allocate_gmres, 335
cal_x3d, 335
calmoinsun_x3d, 336
calmoinsun_x3d
cal_x3d.f90, 336
Cartesien
geom, 106
cellules, 27
dt_film, 28
dx_cell, 28
fff, 28
fff_t, 28
ifich, 28
ifilm, 29
Imax, 29
Imax2, 29
ioffset, 29

N_cell, 29
Pcell, 29
Pmax, 29
Pmax2, 29
size_offset, 29
t_film, 30
centimetre
unites, 152
cg
cg.f90, 182
cg.f90
cal_dif, 181
cg, 182
vec_prod, 182
cg__interface, 175
cmp_precond, 175
mat_prod, 175
check_flags
check_flags.f90, 234
check_flags.f90
check_flags, 234
check_size
check_size.f90, 235
check_size.f90
check_size, 235
cl, 31
B_l, 32
E_l, 32
fx_l, 32
icl, 32
icl0, 32
icl_glob, 32
P_l, 32
rho_l, 33
rhou_l, 33
T_l, 33
u_l, 33
cl_ana
cl_ana.f90, 236
cl_ana.f90
cl_ana, 236
cl_ana_gen, 236
cl_ana_gen
cl_ana.f90, 236
cl_ana_gen_ray
cl_ana_ray.f90, 337
cl_ana_ray
cl_ana_ray.f90, 337
cl_ana_ray.f90
cl_ana_gen_ray, 337
cl_ana_ray, 337
cl_con, 34
icl_con, 34
icl_con_glob, 34
T_l, 34
cl_dif, 35
Er_dif_l, 35
icl_dif, 35
icl_dif_glob, 35
cl_ray, 36
Er_l, 36
Fr_l, 36
icl_ray, 36
icl_ray_glob, 36
Tr_l, 37
close_program
main.f90, 247
cm2
unites, 152
cm3
unites, 153
cmp_mag_flx
umuscl.f90, 312
cmp_precond
cg__interface, 175
cmp_precond_con
conduction_imp_gc.f90, 184
conduction_imp_gc__interface, 177
cmp_precond_dif
diffusion_imp__interface, 178
diffusion_imp_gc.f90, 321
cmp_precond_gra
gc.f90, 191
gc__interface, 179
cmpflxm
umuscl.f90, 313
cntl
TabGmres, 147
coin135_e
communication, 45
communication_gen, 58
communication_ray, 67
coin135_r
communication, 45
communication_gen, 58
communication_ray, 67
coin136_e
communication, 45
communication_gen, 58
coin136_r
communication, 45
communication_gen, 58
coin13_e
communication, 45
communication_gen, 58
communication_ray, 67
coin13_r
communication, 45

communication_gen, 59
communication_ray, 68
coin145_e
communication, 46
communication_gen, 59
communication_ray, 68
coin145_r
communication, 46
communication_gen, 59
communication_ray, 68
coin146_e
communication, 46
communication_gen, 59
coin146_r
communication, 46
communication_gen, 59
coin14_e
communication, 46
communication_gen, 59
coin14_r
communication, 46
communication_gen, 59
coin235_e
communication, 46
communication_gen, 60
communication_ray, 68
coin235_r
communication, 47
communication_gen, 60
communication_ray, 68
coin236_e
communication, 47
communication_gen, 60
coin236_r
communication, 47
communication_gen, 60
coin23_e
communication, 47
communication_gen, 60
communication_ray, 68
coin23_r
communication, 47
communication_gen, 60
communication_ray, 68
coin245_e
communication, 47
communication_gen, 60
communication_ray, 69
coin245_r
communication, 47
communication_gen, 61
communication_ray, 69
coin246_e
communication, 48
communication_gen, 61
coin246_r
communication, 48
communication_gen, 61
coin24_e
communication, 48
communication_gen, 61
coin24_r
communication, 48
communication_gen, 61
COM
divers, 84
comments
divers, 85
communication, 38
ar13_e, 41
ar13_r, 41
ar14_e, 42
ar14_r, 42
ar15_e, 42
ar15_r, 42
ar16_e, 42
ar16_r, 42
ar23_e, 42
ar23_r, 43
ar24_e, 43
ar24_r, 43
ar25_e, 43
ar25_r, 43
ar26_e, 43
ar26_r, 43
ar35_e, 44
ar35_r, 44
ar36_e, 44
ar36_r, 44
ar45_e, 44
ar45_r, 44
ar46_e, 44
ar46_r, 45
coin135_e, 45
coin135_r, 45
coin136_e, 45
coin136_r, 45
coin13_e, 45
coin13_r, 45
coin145_e, 46
coin145_r, 46
coin146_e, 46
coin146_r, 46
coin14_e, 46
coin14_r, 46
coin235_e, 46
coin235_r, 47
coin236_e, 47

coin236_r, 47
coin23_e, 47
coin23_r, 47
coin245_e, 47
coin245_r, 47
coin246_e, 48
coin246_r, 48
coin24_e, 48
coin24_r, 48
face1_e, 48
face1_r, 48
face2_e, 48
face2_r, 49
face3_e, 49
face3_r, 49
face4_e, 49
face4_r, 49
face5_e, 49
face5_r, 49
face6_e, 50
face6_r, 50
communication_gen, 51
ar13_e, 55
ar13_r, 55
ar14_e, 55
ar14_r, 55
ar15_e, 55
ar15_r, 55
ar16_e, 55
ar16_r, 55
ar23_e, 56
ar23_r, 56
ar24_e, 56
ar24_r, 56
ar25_e, 56
ar25_r, 56
ar26_e, 56
ar26_r, 57
ar35_e, 57
ar35_r, 57
ar36_e, 57
ar36_r, 57
ar45_e, 57
ar45_r, 57
ar46_e, 58
ar46_r, 58
coin135_e, 58
coin135_r, 58
coin136_e, 58
coin136_r, 58
coin13_e, 58
coin13_r, 59
coin145_e, 59
coin145_r, 59
coin146_e, 59
coin146_r, 59
coin14_e, 59
coin14_r, 59
coin235_e, 60
coin235_r, 60
coin236_e, 60
coin236_r, 60
coin23_e, 60
coin23_r, 60
coin245_e, 60
coin245_r, 61
coin246_e, 61
coin246_r, 61
coin24_e, 61
coin24_r, 61
face1_e, 61
face1_r, 61
face2_e, 62
face2_r, 62
face3_e, 62
face3_r, 62
face4_e, 62
face4_r, 62
face5_e, 62
face5_r, 63
face6_e, 63
face6_r, 63
Nmax, 63
communication_ray, 64
ar13_e, 66
ar13_r, 66
ar15_e, 66
ar15_r, 66
ar16_e, 66
ar16_r, 66
ar23_e, 66
ar23_r, 66
ar35_e, 67
ar35_r, 67
ar36_e, 67
ar36_r, 67
coin135_e, 67
coin135_r, 67
coin13_e, 67
coin13_r, 68
coin145_e, 68
coin145_r, 68
coin235_e, 68
coin235_r, 68
coin23_e, 68
coin23_r, 68
coin245_e, 69
coin245_r, 69

face1_e, 69
face1_r, 69
face3_e, 69
face3_r, 69
face5_e, 69
face5_r, 70
communications
para.f90, 277
communications_dif
para_dif.f90, 330
communications_gen
para_generique.f90, 284
communications_ray
para_ray.f90, 364
compute_cputime
main.f90, 250
CONDUC
divers, 85
conduction, 71
dt_con_imp, 71
implicit_con, 71
nitetot_con, 71
precond_con, 72
vardt_con, 72
varrel_con, 72
conduction.f90
conduction_ex, 183
conduction/ Directory Reference, 14
conduction/conduction.f90, 183
conduction/conduction_imp_gc.f90, 184
conduction/conduction_step.f90, 186
conduction/init_conduction.f90, 187
conduction/kappa_con.f90, 188
conduction/modules_con.f90, 189
conduction/pasdt_con.f90, 190
conduction_ex
conduction.f90, 183
conduction_imp_gc
conduction_imp_gc.f90, 184
conduction_imp_gc.f90
cmp_precond_con, 184
conduction_imp_gc, 184
Mat_prod_con, 185
conduction_imp_gc__interface, 177
cmp_precond_con, 177
mat_prod_con, 177
conduction_step
conduction_step.f90, 186
conduction_step.f90
conduction_step, 186
const, 73
bigreal, 74
forth, 74
four, 74
half, 74
one, 74
sixth, 74
third, 74
three, 74
two, 75
two3rd, 75
zero, 75
convtoasc
aff.f90, 231
cpu.f90
cpu_, 238
cpu_
cpu.f90, 238
CpuDec
prime.f90, 287
cputime, 76
temps1, 76
temps_comm, 76
temps_conduc, 77
temps_dif, 77
temps_hydro, 77
temps_io, 77
temps_ray, 77
temps_ref, 77
cs1
param_ini, 126
Cs_iso
gammas, 102
Cs_iso2
gammas, 102
cst_gravity
poisson.f90, 200
ctoprim
ctoprim.f90, 299
ctoprim.f90
ctoprim, 299
Cylindrique
geom, 106

d1
param_ini, 126
d2
param_ini, 126
datanames
divers, 85
dataunits
divers, 85
def_geom
geom.f90, 242
default_params
read_params.f90, 289
degre
unites, 153

deja_converge
divers_ray, 95
derive_source
sources_ray.f90, 370
derive_source_dif
init_diffusion.f90, 324
det_3x3
gs.f90, 346
det_4x4
gs.f90, 346
DETO
divers, 85
Dif_imp
diffusion, 78
DIFFU
divers, 85
diffusion, 78
Dif_imp, 78
dt_dif, 78
dt_dif_exp, 79
dt_dif_imp, 79
Er_dif, 79
fld_limiter, 79
nitetot_dif, 79
precond_dif, 79
vardt_dif, 79
varrel_dif, 80
diffusion.f90
diffusion_ex, 320
diffusion_ex
diffusion.f90, 320
diffusion_imp
diffusion_imp_gc.f90, 322
diffusion_imp__interface, 178
cmp_precond_dif, 178
mat_prod_dif, 178
diffusion_imp_gc.f90
cmp_precond_dif, 321
diffusion_imp, 322
mat_prod_dif, 322
diffusion_step
diffusion_step.f90, 323
diffusion_step.f90
diffusion_step, 323
dimensions
init.f90, 228
divers, 81
alpha, 84
COM, 84
comments, 85
CONDUC, 85
datanames, 85
dataunits, 85
DETO, 85
DIFFU, 85
dt, 85
dt_cfl, 86
dt_ff, 86
dt_glob, 86
dt_max, 86
dt_mhd, 86
dtold, 86
fourdim, 87
GRAV, 87
HISTOGRAMME, 87
HYDRO, 87
i_restart, 87
iout, 87
irepr, 87
ISOTHERME, 88
limdx, 88
limfr, 88
limnite, 88
mu, 88
name_of_job, 88
ndata, 88
Nimp, 89
nout, 89
noutd, 89
nrepr, 89
nreprd, 89
nsupp, 89
nsx, 89
nsy, 90
nsz, 90
rankine, 90
RAY, 90
REF, 90
reprise, 90
Solver, 90
sortie, 91
STIR, 91
subcycle, 91
suppdim, 91
suppnames, 91
suppunits, 91
swip, 91
tcpu_ini, 92
tcpu_last, 92
tcpu_max, 92
tcpu_min, 92
temps, 92
tend, 92
tout, 92
trepr, 93
verbose, 93
divers_ray, 94
alpha, 95

alpha_var, 95
cal_valp, 95
deja_converge, 95
dt_ray, 96
dt_ray_exp, 96
dt_ray_glob, 96
dt_ray_old, 96
EXP_RAY, 96
IMP_RAY, 96
nitemax_ray, 96
nitetot_ray, 97
nsy, 97
ordre_des_faces_pour_le_ray, 97
rad_trans_model, 97
rankine, 97
rho1, 97
slope_type_ray, 97
T1, 98
varrel_gaz, 98
varrel_ray, 98
ds
geom, 106
dt
divers, 85
dt_cfl
divers, 86
dt_con_imp
conduction, 71
dt_dif
diffusion, 78
dt_dif_exp
diffusion, 79
dt_dif_imp
diffusion, 79
dt_ff
divers, 86
dt_film
cellules, 28
mfilm, 117
dt_glob
divers, 86
dt_max
divers, 86
dt_mhd
divers, 86
dt_ray
divers_ray, 96
dt_ray_exp
divers_ray, 96
dt_ray_glob
divers_ray, 96
dt_ray_old
divers_ray, 96
dtold
divers, 86
dv
geom, 107
dx
geom, 107
dx_cell
cellules, 28
dxc
geom, 107
dyne
unites, 153

E
var, 164
E1
param_ini, 126
E2
param_ini, 126
E3
param_ini, 126
E4
param_ini, 126
E_l
cl, 32
eigen_cons
godunov_utils.f90, 303
eigenvalues
godunov_utils.f90, 304
Einloc
var_loc, 166
Eloc
var_loc, 166
Energy
eos.f90, 211
Eold
varold, 169
eos.f90
Energy, 211
heat_capacity, 211
init_eos, 212
Pressure, 212
Sound_speed, 213
Temperature, 213
Er_dif
diffusion, 79
Er_dif_l
cl_dif, 35
Er_l
cl_ray, 36
Eray
varray, 171
Eray_t
varray, 171
Erayold

varray, 172
erg
unites, 153
eS
Etat_GDS, 99
essai_inv5x5
gs.f90, 346
Etat_GDS, 99
eS, 99
FxS, 99
pS, 99
rhoS, 99
uS, 99
eV
unites, 153
evol_hydro
evol_hydro.f90, 214
evol_hydro.f90
evol_hydro, 214
evol_mhd
evol_mhd.f90, 300
evol_mhd.f90
evol_mhd, 300
evol_ray
evol_ray.f90, 339
evol_ray.f90
evol_ray, 339
pente_minmod, 340
exp_comobile_ray
exp_comobile_ray.f90, 317
exp_comobile_ray.f90
exp_comobile_ray, 317
EXP_RAY
divers_ray, 96
exp_source_ray
exp_source_ray.f90, 318
exp_source_ray.f90
exp_source_ray, 318
explicite
explicite.f90, 341
explicite.f90
explicite, 341

f
rh.f90, 225
face1_e
communication, 48
communication_gen, 61
communication_ray, 69
face1_r
communication, 48
communication_gen, 61
communication_ray, 69
face2_e
communication, 48
communication_gen, 62
face2_r
communication, 49
communication_gen, 62
face3_e
communication, 49
communication_gen, 62
communication_ray, 69
face3_r
communication, 49
communication_gen, 62
communication_ray, 69
face4_e
communication, 49
communication_gen, 62
face4_r
communication, 49
communication_gen, 62
face5_e
communication, 49
communication_gen, 62
communication_ray, 69
face5_r
communication, 49
communication_gen, 63
communication_ray, 70
face6_e
communication, 50
communication_gen, 63
face6_r
communication, 50
communication_gen, 63
fast_mhd_speed
umuscl.f90, 314
ff_t
varray, 172
fff
cellules, 28
fff_t
cellules, 28
fill_new_arrays
fill_new_arrays.f90, 216
fill_new_arrays.f90
fill_new_arrays, 216
FILM
mfilm, 117
film.f90
film_, 240
init_film, 240
film_
film.f90, 240
film_hdf
film_hdf.f90, 203

film_hdf.f90
film_hdf, 203
find_mhd_flux
godunov_utils.f90, 304
find_mhd_flux2
godunov_utils.f90, 305
find_prime
prime.f90, 288
find_speed_alfven
godunov_utils.f90, 305
find_speed_fast
godunov_utils.f90, 305
find_speed_info
godunov_utils.f90, 306
fld_limiter
diffusion, 79
forth
const, 74
four
const, 74
fourdim
divers, 87
Fr_l
cl_ray, 36
Fray
varray, 172
Fray_t
varray, 172
Frayold
varray, 172
Fx
var, 164
fx_l
cl, 32
Fxloc
var_loc, 167
Fxold
varold, 169
FxS
Etat_GDS, 99

G
unites, 153
g1
gammas, 102
g2
gammas, 102
g3
gammas, 102
g4
gammas, 102
g5
gammas, 103
g6
gammas, 103
g7
gammas, 103
g8
gammas, 103
gamma
gammas, 103
gamma1
gammas, 103
gamma2
gammas, 104
gammas, 101
Cs_iso, 102
Cs_iso2, 102
g1, 102
g2, 102
g3, 102
g4, 102
g5, 103
g6, 103
g7, 103
g8, 103
gamma, 103
gamma1, 103
gamma2, 104
mu_reac, 104
Gans
unites, 153
Gauss
unites, 154
gc
gc.f90, 191
gc.f90
cmp_precond_gra, 191
gc, 191
mat_prod_gra, 192
gc__interface, 179
cmp_precond_gra, 179
mat_prod_gra, 179
geom, 105
box_max, 106
box_min, 106
Cartesien, 106
Cylindrique, 106
ds, 106
dv, 107
dx, 107
dxc, 107
geom_dir, 107
geometrie, 107
Lbox, 107
nshift_gr, 108
shift_gr, 108
Spherique, 108

x, 108
x_glob, 108
xcloc, 108
geom.f90
cal_geom, 241
def_geom, 242
geom_dir
geom, 107
geometrie
geom, 107
gmres
gmres.f90, 343
TabGmres::interface, 180
gmres.f90
gmres, 343
godunov_utils.f90
athena_roe, 303
eigen_cons, 303
eigenvalues, 304
find_mhd_flux, 304
find_mhd_flux2, 305
find_speed_alfven, 305
find_speed_fast, 305
find_speed_info, 306
hll, 306
hlld, 306
hydro_acoustic, 307
lax_friedrich, 307
upwind, 308
gramme
unites, 154
GRAV
divers, 87
grav_predictor
grav_predictor.f90, 193
grav_predictor.f90
grav_predictor, 193
gravity, 110
gravity_params, 110
gravity_type, 110
ISOLE, 110
phi, 110
phiold, 110
gravity/ Directory Reference, 16
gravity/gc.f90, 191
gravity/grav_predictor.f90, 193
gravity/init_gravity.f90, 194
gravity/limites_gra.f90, 195
gravity/modules_gra.f90, 197
gravity/pasdt_grav.f90, 199
gravity/poisson.f90, 200
gravity/update_gravity.f90, 202
gravity_params
gravity, 110
gravity_type
gravity, 110
grid_cpu
para, 121
grid_pos
para, 121
gs.f90
cal_Dedd, 345
det_3x3, 346
det_4x4, 346
essai_inv5x5, 346
gs_mat, 346
gs_mat
gs.f90, 346
guessp
solvers_hydro, 138
guessp_iso
solvers_hydro, 138

H0
unites, 154
half
const, 74
hdf/ Directory Reference, 17
hdf/film_hdf.f90, 203
hdf/rd_restart_h5.f90, 204
hdf/rdwrt_h5.f90, 205
hdf/utils_h5.f90, 206
hdf/wrt_film_h5.f90, 207
hdf/wrt_main_h5.f90, 208
hdf/wrt_output_h5.f90, 209
hdf/wrt_restart_h5.f90, 210
heat_capacity
eos.f90, 211
heracles
main.f90, 250
histo, 112
hN, 113
hNg, 113
hP, 113
hPg, 113
hT, 114
hTg, 114
lr_N, 114
lr_P, 114
lr_T, 114
N, 114
Nmax, 114
Nmin, 115
Npoint, 115
P, 115
Pmax, 115
Pmin, 115
r_N, 115

r_P, 115
r_T, 116
T, 116
Tmax, 116
Tmin, 116
HISTOGRAMME
divers, 87
histogramme.f90
cal_histogramme, 219
init_histo, 219
hll
godunov_utils.f90, 306
hlld
godunov_utils.f90, 306
hN
histo, 113
hNg
histo, 113
hP
histo, 113
hPg
histo, 113
hplanck
unites, 154
hT
histo, 114
hTg
histo, 114
HYDRO
divers, 87
hydro/ Directory Reference, 18
hydro/eos.f90, 211
hydro/evol_hydro.f90, 214
hydro/fill_new_arrays.f90, 216
hydro/histogramme.f90, 217
hydro/hydro_step.f90, 220
hydro/monitoring.f90, 223
hydro/pasdt_hydro.f90, 224
hydro/rh.f90, 225
hydro/solvers.f90, 226
hydro_acoustic
godunov_utils.f90, 307
hydro_step
hydro_step.f90, 220
hydro_step.f90
hydro_step, 220

i_restart
divers, 87
icl
cl, 32
icl0
cl, 32
icl_con
cl_con, 34
icl_con_glob
cl_con, 34
icl_dif
cl_dif, 35
icl_dif_glob
cl_dif, 35
icl_glob
cl, 32
icl_ray
cl_ray, 36
icl_ray_glob
cl_ray, 36
icntl
TabGmres, 147
ifich
cellules, 28
ifilm
cellules, 29
mfilm, 118
Imax
cellules, 29
Imax2
cellules, 29
imin
TabGmres, 147
IMP_RAY
divers_ray, 96
implicit_con
conduction, 71
init
init.f90, 228
init.f90
dimensions, 228
init, 228
init/ Directory Reference, 19
init/init.f90, 228
init/init_out.f90, 229
init_conduction
init_conduction.f90, 187
init_conduction.f90
init_conduction, 187
init_diffusion
init_diffusion.f90, 324
init_diffusion.f90
allocate_array_dif, 324
derive_source_dif, 324
init_diffusion, 324
source_dif, 325
init_eos
eos.f90, 212
init_film
film.f90, 240
init_gravity

init_gravity.f90, 194
init_gravity.f90
init_gravity, 194
init_histo
histogramme.f90, 219
init_out
init_out.f90, 229
init_out.f90
init_out, 229
init_para
para.f90, 278
init_phys
init_phys.f90, 243
init_phys.f90
init_phys, 243
init_ray
init_ray.f90, 348
init_ray.f90
init_ray, 348
initVarGmres
initvargmres.f90, 350
initvargmres.f90
initVarGmres, 350
interpol_valp
valp.f90, 371
inversion_gmres
inversion_gmres.f90, 351
inversion_gmres.f90
inversion_gmres, 351
ioffset
cellules, 29
iout
divers, 87
irepr
divers, 87
ISOLE
gravity, 110
ISOTHERME
divers, 88
itermax
TabGmres, 147

jmin
TabGmres, 147
joule
unites, 154

kappa_abs_t
varray, 172
kappa_con.f90
cal_kappa_con, 188
kappa_dif.f90
cal_kappa_dif, 326
kappa_ray.f90
cal_kappa, 352
cal_sigmadiff, 352
kB
unites, 154
Kelvin
unites, 154
kg
unites, 155
kmin
TabGmres, 147
kms
unites, 155
kpc
unites, 155

lax_friedrich
godunov_utils.f90, 307
Lbox
geom, 107
lim_x
para, 121
lim_y
para, 121
lim_z
para, 121
limdx
divers, 88
limfr
divers, 88
limites
limites.f90, 245
limites.f90
limites, 245
limites_dif
limites_dif.f90, 327
limites_dif.f90
limites_dif, 327
limites_gen
limites_generique.f90, 246
limites_generique.f90
limites_gen, 246
limites_gra
limites_gra.f90, 195
limites_gra.f90
cal_barycentre, 195
limites_gra, 195
limites_ray
limites_ray.f90, 354
limites_ray.f90
limites_ray, 354
limites_xin
mat_prod3d.f90, 356
limnite
divers, 88

lr_N
histo, 114
lr_P
histo, 114
lr_T
histo, 114
Lsol
unites, 155

m2
unites, 155
m3
unites, 155
main.f90
Arret, 247
close_program, 247
compute_cputime, 250
heracles, 250
MPI_Wtime, 252
output, 252
tremain, 252
main/ Directory Reference, 22
main/aff.f90, 230
main/allocate_array.f90, 233
main/check_flags.f90, 234
main/check_size.f90, 235
main/cl_ana.f90, 236
main/cpu.f90, 238
main/film.f90, 239
main/geom.f90, 241
main/init_phys.f90, 243
main/limites.f90, 245
main/limites_generique.f90, 246
main/main.f90, 247
main/modules.f90, 254
main/para.f90, 272
main/para_generique.f90, 280
main/pasdt.f90, 286
main/prime.f90, 287
main/read_params.f90, 289
main/restart.f90, 291
main/set_units_out.f90, 292
main/slopes.f90, 293
main/user_init.f90, 296
main/user_output.f90, 297
main/user_step.f90, 298
make_name
aff.f90, 231
Mans
unites, 155
mat_prod
cg__interface, 175
mat_prod3d
mat_prod3d.f90, 356
mat_prod3d.f90
limites_xin, 356
mat_prod3d, 356
Mat_prod_con
conduction_imp_gc.f90, 185
mat_prod_con
conduction_imp_gc__interface, 177
mat_prod_dif
diffusion_imp__interface, 178
diffusion_imp_gc.f90, 322
mat_prod_gra
gc.f90, 192
gc__interface, 179
metre
unites, 156
mfilm, 117
dt_film, 117
FILM, 117
ifilm, 118
nl1, 118
nl2, 118
nl3, 118
t_end, 118
t_film, 118
t_start, 118
mhd/ Directory Reference, 24
mhd/ctoprim.f90, 299
mhd/evol_mhd.f90, 300
mhd/godunov_utils.f90, 302
mhd/pasdt_mhd.f90, 309
mhd/trace.f90, 310
mhd/umuscl.f90, 312
mhd/update.f90, 316
micron
unites, 156
minmod
slopes.f90, 293
mk_dir_out
aff.f90, 232
modules_gra.f90
allocate_array_gra, 197
moncen
slopes.f90, 293
monitoring
monitoring.f90, 223
monitoring.f90
monitoring, 223
moyharm
slopes.f90, 294
Mpc
unites, 156
MPI_Wtime
main.f90, 252
ms

unites, 156
Msol
unites, 156
mu
divers, 88
mu0
unites, 156
mu_reac
gammas, 104
mype
para, 121

N
histo, 114
N_cell
cellules, 29
n_points
valeurs_propres, 162
N_vit
parameters, 131
name_of_job
divers, 88
Nbuf
parameters, 131
ncpu
para, 122
ncpu_x
para, 122
ncpu_y
para, 122
ncpu_z
para, 122
ndata
divers, 88
ndim
parameters, 131
nFx
parameters, 132
Nimp
divers, 89
nitemax_ray
divers_ray, 96
nitetot_con
conduction, 71
nitetot_dif
diffusion, 79
nitetot_ray
divers_ray, 97
nl1
mfilm, 118
nl2
mfilm, 118
nl3
mfilm, 118
Nmax
communication_gen, 63
histo, 114
Nmin
histo, 115
nout
divers, 89
noutd
divers, 89
Npoint
histo, 115
Nprime
prime, 135
nrepr
divers, 89
nreprd
divers, 89
ns
unites, 156
nshift_gr
geom, 108
nsupp
divers, 89
nsx
divers, 89
nsy
divers, 90
divers_ray, 97
nsz
divers, 90
Number
prime, 135
nvar
parameters, 132
nvar_ray
parameters, 132
nx
parameters, 132
nx_cpu
parameters, 132
nx_glob
parameters, 133
nx_glob_max
parameters, 133
nx_max
parameters, 133
nxmax
parameters, 133
nxmin
parameters, 133

one
const, 74
ordre_des_faces_pour_le_ray

divers_ray, 97
output
main.f90, 252
output_data
utils_h5.f90, 206

P
histo, 115
p1
param_ini, 127
p2
param_ini, 127
p3
param_ini, 127
p4
param_ini, 127
P_l
cl, 32
para, 120
grid_cpu, 121
grid_pos, 121
lim_x, 121
lim_y, 121
lim_z, 121
mype, 121
ncpu, 122
ncpu_x, 122
ncpu_y, 122
ncpu_z, 122
voisin, 122
x_end, 122
x_end_cpu, 122
x_start, 123
x_start_cpu, 123
para.f90
allocate_array_com, 277
communications, 277
init_para, 278
para_dif.f90
communications_dif, 330
para_generique.f90
allocate_array_com_gen, 284
communications_gen, 284
para_ray.f90
allocate_array_com_ray, 364
communications_ray, 364
param_ini, 124
B0, 126
cs1, 126
d1, 126
d2, 126
E1, 126
E2, 126
E3, 126
E4, 126
p1, 127
p2, 127
p3, 127
p4, 127
rho1, 127
rho2, 127
rho3, 127
rho4, 127
T1, 127
T2, 128
T3, 128
T4, 128
Temperature_isotherme, 128
u1, 128
u2, 128
u3, 128
u4, 128
v1, 129
v2, 129
v3, 129
v4, 129
ww, 129
x1, 129
x2, 129
parameters, 130
N_vit, 131
Nbuf, 131
ndim, 131
nFx, 132
nvar, 132
nvar_ray, 132
nx, 132
nx_cpu, 132
nx_glob, 133
nx_glob_max, 133
nx_max, 133
nxmax, 133
nxmin, 133
Pi, 133
quatre_Pi, 134
riemann, 134
riemann2d, 134
slope_type, 134
smallc, 134
smallr, 134
Pascal
unites, 156
pasdt
pasdt.f90, 286
pasdt.f90
pasdt, 286
pasdt_con
pasdt_con.f90, 190

pasdt_con.f90
    pasdt_con, 190
pasdt_dif
    pasdt_dif.f90, 331
pasdt_dif.f90
    pasdt_dif, 331
pasdt_grav
    pasdt_grav.f90, 199
pasdt_grav.f90
    pasdt_grav, 199
pasdt_hydro
    pasdt_hydro.f90, 224
pasdt_hydro.f90
    pasdt_hydro, 224
pasdt_mhd
    pasdt_mhd.f90, 309
pasdt_mhd.f90
    pasdt_mhd, 309
pasdt_ray
    pasdt_ray.f90, 366
pasdt_ray.f90
    pasdt_ray, 366
    pasdt_ray_exp, 366
pasdt_ray_exp
    pasdt_ray.f90, 366
pc
    unites, 157
Pcell
    cellules, 29
Pdmax
    prime, 135
pente_minmod
    evol_ray.f90, 340
phi
    gravity, 110
philoc
    var_loc, 167
phiold
    gravity, 110
Pi
    parameters, 133
Ploc
    var_loc, 167
Pmax
    cellules, 29
    histo, 115
    prime, 135
Pmax2
    cellules, 29
Pmin
    histo, 115
poisson
    poisson.f90, 200
poisson.f90
    cst_gravity, 200
    poisson, 200
    pt_mass, 201
precond_con
    conduction, 72
precond_dif
    diffusion, 79
prefun
    solvers_hydro, 139
prefun_iso
    solvers_hydro, 139
Pressure
    eos.f90, 212
prime, 135
    Nprime, 135
    Number, 135
    Pdmax, 135
    Pmax, 135
    PrimeNumber, 135
prime.f90
    CpuDec, 287
    find_prime, 288
    PrimeDec, 288
PrimeDec
    prime.f90, 288
PrimeNumber
    prime, 135
print_configuration
    aff.f90, 232
pS
    Etat_GDS, 99
pt_mass
    poisson.f90, 201

quatre_Pi
    parameters, 134

r_N
    histo, 115
r_P
    histo, 115
r_T
    histo, 116
rad_trans_model
    divers_ray, 97
rad_transfer/ Directory Reference, 25
rad_transfer/exp_comobile_ray.f90, 317
rad_transfer/exp_source_ray.f90, 318
rad_transfer/fld/ Directory Reference, 15
rad_transfer/fld/diffusion.f90, 320
rad_transfer/fld/diffusion_imp_gc.f90, 321
rad_transfer/fld/diffusion_step.f90, 323
rad_transfer/fld/init_diffusion.f90, 324
rad_transfer/fld/kappa_dif.f90, 326

rad_transfer/fld/limites_dif.f90, 327
rad_transfer/fld/modules_dif.f90, 328
rad_transfer/fld/para_dif.f90, 330
rad_transfer/fld/pasdt_dif.f90, 331
rad_transfer/m1/ Directory Reference, 20
rad_transfer/m1/allocate_array_ray.f90, 332
rad_transfer/m1/cal_b3d.f90, 333
rad_transfer/m1/cal_x3d.f90, 335
rad_transfer/m1/cl_ana_ray.f90, 337
rad_transfer/m1/evol_ray.f90, 339
rad_transfer/m1/explicite.f90, 341
rad_transfer/m1/gmres.f90, 343
rad_transfer/m1/gs.f90, 345
rad_transfer/m1/init_ray.f90, 348
rad_transfer/m1/initvargmres.f90, 350
rad_transfer/m1/inversion_gmres.f90, 351
rad_transfer/m1/kappa_ray.f90, 352
rad_transfer/m1/limites_ray.f90, 354
rad_transfer/m1/mat_prod3d.f90, 356
rad_transfer/m1/modules_ray.f90, 358
rad_transfer/m1/para_ray.f90, 362
rad_transfer/m1/pasdt_ray.f90, 366
rad_transfer/m1/ray_step.f90, 368
rad_transfer/m1/sources_ray.f90, 370
rad_transfer/m1/valp.f90, 371
rankine
    divers, 90
    divers_ray, 97
RAY
    divers, 90
ray_step
    ray_step.f90, 368
ray_step.f90
    ray_step, 368
rd_restart_h5
    rd_restart_h5.f90, 204
rd_restart_h5.f90
    rd_restart_h5, 204
rdwrt_h5, 136
read_params
    read_params.f90, 289
read_params.f90
    default_params, 289
    read_params, 289
read_valp
    valp.f90, 371
REF
    divers, 90
reprise
    divers, 90
restart.f90
    restart_new, 291
restart_new
    restart.f90, 291
rh
    rh.f90, 225
rh.f90
    f, 225
    rh, 225
rho
    var, 164
rho1
    divers_ray, 97
    param_ini, 127
rho2
    param_ini, 127
rho3
    param_ini, 127
rho4
    param_ini, 127
rho_l
    cl, 33
rholoc
    var_loc, 167
rhoold
    varold, 170
rhoS
    Etat_GDS, 99
rhou
    var, 164
rhou_l
    cl, 33
rhouloc
    var_loc, 167
rhouold
    varold, 170
riemann
    parameters, 134
riemann2d
    parameters, 134
Rsol
    unites, 157

sample
    solvers_hydro, 139
sample_iso
    solvers_hydro, 140
seconde
    unites, 157
set_units_out
    set_units_out.f90, 292
set_units_out.f90
    set_units_out, 292
shift_gr
    geom, 108
sigma_diff
    varray, 173
sixth

const, 74
size_offset
cellules, 29
slope_type
parameters, 134
slope_type_ray
divers_ray, 97
slopes
slopes.f90, 294
slopes.f90
minmod, 293
moncen, 293
moyharm, 294
slopes, 294
vavl, 295
smallc
parameters, 134
smallr
parameters, 134
Solver
divers, 90
solver_acoustic
solvers_hydro, 140
solver_cc
solvers_hydro, 140
solver_eos
solvers_hydro, 141
solver_exact
solvers_hydro, 141
solver_g
solvers_hydro, 142
solver_iso
solvers_hydro, 143
solver_relax
solvers_hydro, 143
solver_relaxation
solvers_hydro, 144
solvers_hydro, 137
cal_gamma, 138
guessp, 138
guessp_iso, 138
prefun, 139
prefun_iso, 139
sample, 139
sample_iso, 140
solver_acoustic, 140
solver_cc, 140
solver_eos, 141
solver_exact, 141
solver_g, 142
solver_iso, 143
solver_relax, 143
solver_relaxation, 144
starpu, 144
starpu_iso, 145
sortie
divers, 91
sortie_ecran
aff.f90, 232
Sound_speed
eos.f90, 213
source
sources_ray.f90, 370
source_dif
init_diffusion.f90, 325
sources_ray.f90
derive_source, 370
source, 370
Spherique
geom, 108
starpu
solvers_hydro, 144
starpu_iso
solvers_hydro, 145
STIR
divers, 91
subcycle
divers, 91
supp1
var, 165
supp2
var, 165
supp3
var, 165
supp4
var, 165
supp5
var, 165
suppdim
divers, 91
suppnames
divers, 91
suppunits
divers, 91
swip
divers, 91

T
histo, 116
T1
divers_ray, 98
param_ini, 127
T2
param_ini, 128
T3
param_ini, 128
T4
param_ini, 128

t_end
mfilm, 118
t_film
cellules, 30
mfilm, 118
T_l
cl, 33
cl_con, 34
t_start
mfilm, 118
TabGmres, 146
b, 147
cntl, 147
icntl, 147
imin, 147
itermax, 147
jmin, 147
kmin, 147
work, 147
xin, 147
TabGmres::interface, 180
gmres, 180
tcpu_ini
divers, 92
tcpu_last
divers, 92
tcpu_max
divers, 92
tcpu_min
divers, 92
Temperature
eos.f90, 213
Temperature_isotherme
param_ini, 128
temps
divers, 92
temps1
cputime, 76
temps_comm
cputime, 76
temps_conduc
cputime, 77
temps_dif
cputime, 77
temps_hydro
cputime, 77
temps_io
cputime, 77
temps_ray
cputime, 77
temps_ref
cputime, 77
tend
divers, 92
Tgaz
var, 165
Tgazold
varold, 170
third
const, 74
three
const, 74
Tloc
var_loc, 167
Tmax
histo, 116
Tmin
histo, 116
tout
divers, 92
Tr_l
cl_ray, 37
trace.f90
trace1d, 310
trace2d, 310
trace3d, 311
trace1d
trace.f90, 310
trace2d
trace.f90, 310
trace3d
trace.f90, 311
tremain
main.f90, 252
trepr
divers, 93
two
const, 75
two3rd
const, 75

u1
param_ini, 128
u2
param_ini, 128
u3
param_ini, 128
u4
param_ini, 128
u_dens
unites_sortie, 159
u_E
unites_sortie, 159
u_evol
unites_sortie, 160
u_Fr
unites_sortie, 160
u_L

unites_sortie, 160
u_l
cl, 33
u_M
unites_sortie, 160
u_Mom
unites_sortie, 160
u_T
unites_sortie, 160
u_Vit
unites_sortie, 160
u_Vol
unites_sortie, 161
uloc
var_loc, 167
uma
unites, 157
umuscl.f90
cmp_mag_flx, 312
cmpflxm, 313
fast_mhd_speed, 314
uslope, 314
unitd
unites, 157
unite
unites, 157
unites, 149
a_R, 152
an, 152
bar, 152
c, 152
c2, 152
centimetre, 152
cm2, 152
cm3, 153
degre, 153
dyne, 153
erg, 153
eV, 153
G, 153
Gans, 153
Gauss, 154
gramme, 154
H0, 154
hplanck, 154
joule, 154
kB, 154
Kelvin, 154
kg, 155
kms, 155
kpc, 155
Lsol, 155
m2, 155
m3, 155
Mans, 155
metre, 156
micron, 156
Mpc, 156
ms, 156
Msol, 156
mu0, 156
ns, 156
Pascal, 156
pc, 157
Rsol, 157
seconde, 157
uma, 157
unitd, 157
unite, 157
unitf, 157
unitl, 158
unitm, 158
unitmom, 158
unitt, 158
unites_sortie, 159
u_dens, 159
u_E, 159
u_evol, 160
u_Fr, 160
u_L, 160
u_M, 160
u_Mom, 160
u_T, 160
u_Vit, 160
u_Vol, 161
unitf
unites, 157
unitl
unites, 158
unitm
unites, 158
unitmom
unites, 158
unitt
unites, 158
update
update.f90, 316
update.f90
update, 316
update_gravity
update_gravity.f90, 202
update_gravity.f90
update_gravity, 202
upwind
godunov_utils.f90, 308
uS
Etat_GDS, 99
user_init

user_init.f90, 296
user_init.f90
user_init, 296
user_output
user_output.f90, 297
user_output.f90
user_output, 297
user_step
user_step.f90, 298
user_step.f90
user_step, 298
uslope
umuscl.f90, 314
utils_h5.f90
output_data, 206

v1
param_ini, 129
v2
param_ini, 129
v3
param_ini, 129
v4
param_ini, 129
valeurs_propres, 162
n_points, 162
valp, 162
valp
valeurs_propres, 162
valp.f90
interpol_valp, 371
read_valp, 371
var, 163
B, 164
E, 164
Fx, 164
rho, 164
rhou, 164
supp1, 165
supp2, 165
supp3, 165
supp4, 165
supp5, 165
Tgaz, 165
var_loc, 166
Einloc, 166
Eloc, 166
Fxloc, 167
philoc, 167
Ploc, 167
rholoc, 167
rhouloc, 167
Tloc, 167
uloc, 167
xloc, 168
vardt_con
conduction, 72
vardt_dif
diffusion, 79
varold, 169
Bold, 169
Eold, 169
Fxold, 169
rhoold, 170
rhouold, 170
Tgazold, 170
varray, 171
Eray, 171
Eray_t, 171
Erayold, 172
ff_t, 172
Fray, 172
Fray_t, 172
Frayold, 172
kappa_abs_t, 172
sigma_diff, 173
varrel_con
conduction, 72
varrel_dif
diffusion, 80
varrel_gaz
divers_ray, 98
varrel_ray
divers_ray, 98
vavl
slopes.f90, 295
vec_prod
cg.f90, 182
verbose
divers, 93
voisin
para, 122

work
TabGmres, 147
wrt_film_h5
wrt_film_h5.f90, 207
wrt_film_h5.f90
wrt_film_h5, 207
wrt_main_h5
wrt_main_h5.f90, 208
wrt_main_h5.f90
wrt_main_h5, 208
wrt_output_h5
wrt_output_h5.f90, 209
wrt_output_h5.f90
wrt_output_h5, 209
wrt_restart_h5

wrt_restart_h5.f90, 210
wrt_restart_h5.f90
wrt_restart_h5, 210
ww
param_ini, 129

x
geom, 108
x1
param_ini, 129
x2
param_ini, 129
x_end
para, 122
x_end_cpu
para, 122
x_glob
geom, 108
x_start
para, 123
x_start_cpu
para, 123
xcloc
geom, 108
xin
TabGmres, 147
xloc
var_loc, 168

zero
const, 75