

Tempo2

Generated by Doxygen 1.8.9.1

Thu Sep 17 2015 08:22:37

Contents

1	Main Page	1
2	Developer Guide	3
2.1	Tempo2 Developer Guide	3
2.1.1	About this guide	3
2.1.2	General code guidelines	3
2.1.3	Development workflow	4
2.1.4	Coding style	4
3	Core Developers	7
4	Directory structure	9
5	User Guide	11
6	Todo List	13
7	Module Index	15
7.1	Modules	15
8	Class Index	17
8.1	Class List	17
9	File Index	19
9.1	File List	19
10	Module Documentation	23
10.1	libt2toolkit API	23
10.1.1	Detailed Description	23
10.2	libtempo2 External API	24
10.2.1	Detailed Description	24
11	Class Documentation	25
11.1	Cheby2D Struct Reference	25
11.1.1	Member Data Documentation	25

11.1.1.1	coeff	25
11.1.1.2	nx	25
11.1.1.3	ny	25
11.2	ChebyModel Struct Reference	25
11.2.1	Member Data Documentation	26
11.2.1.1	cheby	26
11.2.1.2	dispersion_constant	26
11.2.1.3	freq_end	26
11.2.1.4	freq_start	26
11.2.1.5	frequency_cheby	26
11.2.1.6	mjd_end	26
11.2.1.7	mjd_start	26
11.2.1.8	psrname	26
11.2.1.9	sitename	26
11.3	ChebyModelInfo Struct Reference	27
11.3.1	Member Data Documentation	27
11.3.1.1	compute_dispersion_constant	27
11.3.1.2	model	27
11.3.1.3	psr	27
11.4	ChebyModelSet Struct Reference	27
11.4.1	Member Data Documentation	28
11.4.1.1	nsegments	28
11.4.1.2	segments	28
11.5	clock_correction Struct Reference	28
11.5.1	Detailed Description	28
11.5.2	Member Data Documentation	29
11.5.2.1	correction	29
11.5.2.2	corrects_to	29
11.6	ClockCorrectionFunction Struct Reference	29
11.6.1	Member Data Documentation	29
11.6.1.1	badness	29
11.6.1.2	clockFrom	29
11.6.1.3	clockTo	29
11.6.1.4	table	29
11.7	complexVal Struct Reference	30
11.7.1	Member Data Documentation	30
11.7.1.1	imag	30
11.7.1.2	real	30
11.8	DynamicArray Struct Reference	30
11.8.1	Member Data Documentation	30

11.8.1.1	data	30
11.8.1.2	elem_size	30
11.8.1.3	nallocated	30
11.8.1.4	nelem	30
11.9	EOPSample Struct Reference	30
11.9.1	Member Data Documentation	31
11.9.1.1	dut1	31
11.9.1.2	mjd	31
11.9.1.3	xp	31
11.9.1.4	yp	31
11.10	FitInfo Struct Reference	31
11.10.1	Detailed Description	32
11.10.2	Member Data Documentation	32
11.10.2.1	constraintCounters	32
11.10.2.2	constraintDerivs	32
11.10.2.3	constraintIndex	32
11.10.2.4	nConstraints	32
11.10.2.5	nParams	32
11.10.2.6	paramCounters	32
11.10.2.7	paramDerivs	32
11.10.2.8	paramIndex	32
11.10.2.9	updateFunctions	32
11.11	glitchS Struct Reference	32
11.11.1	Member Data Documentation	32
11.11.1.1	fitf0	32
11.11.1.2	fitf0d	32
11.11.1.3	fitf1	33
11.11.1.4	fitph	33
11.11.1.5	fitted	33
11.11.1.6	glep	33
11.11.1.7	glf0	33
11.11.1.8	glf0d	33
11.11.1.9	glf1	33
11.11.1.10	glph	33
11.11.1.11	gltd	33
11.12	gwgeneralSrc Struct Reference	33
11.12.1	Member Data Documentation	34
11.12.1.1	across_g	34
11.12.1.2	across_im_g	34
11.12.1.3	aplus_g	34

11.12.1.4	aplu _s _im_g	34
11.12.1.5	asl_g	34
11.12.1.6	asl_im_g	34
11.12.1.7	ast_g	34
11.12.1.8	ast_im_g	34
11.12.1.9	avx_g	34
11.12.1.10	avx_im_g	34
11.12.1.11	avy_g	34
11.12.1.12	avy_im_g	34
11.12.1.13	dist_bin	34
11.12.1.14	h	34
11.12.1.15	h_im	34
11.12.1.16	nc_bin	34
11.12.1.17	kg	34
11.12.1.18	omega_g	34
11.12.1.19	phase_g	34
11.12.1.20	phi_bin	34
11.12.1.21	phi_g	34
11.12.1.22	phi_polar_g	34
11.12.1.23	theta_bin	34
11.12.1.24	theta_g	34
11.13	gwgenSpec Struct Reference	35
11.13.1	Member Data Documentation	35
11.13.1.1	sl_alpha	35
11.13.1.2	sl_amp	35
11.13.1.3	st_alpha	35
11.13.1.4	st_amp	35
11.13.1.5	tensor_alpha	35
11.13.1.6	tensor_amp	35
11.13.1.7	vl_alpha	35
11.13.1.8	vl_amp	35
11.14	gwSrc Struct Reference	35
11.14.1	Member Data Documentation	36
11.14.1.1	across_g	36
11.14.1.2	across_im_g	36
11.14.1.3	aplu _s _g	36
11.14.1.4	aplu _s _im_g	36
11.14.1.5	dist_bin	36
11.14.1.6	h	36
11.14.1.7	h_im	36

11.14.1.8 inc_bin	36
11.14.1.9 kg	36
11.14.1.10 omega_g	36
11.14.1.11 phase_g	36
11.14.1.12 phi_bin	36
11.14.1.13 phi_g	36
11.14.1.14 phi_polar_g	36
11.14.1.15 theta_bin	36
11.14.1.16 theta_g	36
11.15 IFTE_interpolation_info Struct Reference	36
11.15.1 Member Data Documentation	37
11.15.1.1 np	37
11.15.1.2 nv	37
11.15.1.3 pc	37
11.15.1.4 twot	37
11.15.1.5 vc	37
11.16 IFTEphemeris Struct Reference	37
11.16.1 Member Data Documentation	38
11.16.1.1 buf	38
11.16.1.2 endJD	38
11.16.1.3 ephver	38
11.16.1.4 f	38
11.16.1.5 iinfo	38
11.16.1.6 ipt	38
11.16.1.7 irec	38
11.16.1.8 L_C	38
11.16.1.9 reclen	38
11.16.1.10 startJD	38
11.16.1.11 stepJD	38
11.16.1.12 swap_endian	38
11.16.1.13 title	38
11.17 interpolation_info Struct Reference	38
11.17.1 Member Data Documentation	38
11.17.1.1 np	38
11.17.1.2 nv	38
11.17.1.3 pc	38
11.17.1.4 twot	38
11.17.1.5 vc	39
11.18 jpl_eph_data Struct Reference	39
11.18.1 Member Data Documentation	39

11.18.1.1 au	39
11.18.1.2 cache	39
11.18.1.3 curr_cache_loc	39
11.18.1.4 emrat	39
11.18.1.5 ephem_end	39
11.18.1.6 ephem_start	39
11.18.1.7 ephem_step	39
11.18.1.8 ephemeris_version	39
11.18.1.9 ifile	39
11.18.1.10 info	39
11.18.1.11 iplt	40
11.18.1.12 kernel_size	40
11.18.1.13 ncoeff	40
11.18.1.14 ncon	40
11.18.1.15 pvsun	40
11.18.1.16 recsize	40
11.18.1.17 swap_bytes	40
11.19 lm_control_struct Struct Reference	40
11.19.1 Member Data Documentation	40
11.19.1.1 epsilon	40
11.19.1.2 ftol	40
11.19.1.3 gtol	40
11.19.1.4 maxcall	40
11.19.1.5 printflags	40
11.19.1.6 scale_diag	40
11.19.1.7 stepbound	40
11.19.1.8 xt看ol	40
11.20 lm_status_struct Struct Reference	41
11.20.1 Member Data Documentation	41
11.20.1.1 fnorm	41
11.20.1.2 info	41
11.20.1.3 nfev	41
11.21 lmcurve_data_struct Struct Reference	41
11.21.1 Member Data Documentation	41
11.21.1.1 f	41
11.21.1.2 t	41
11.21.1.3 y	41
11.22 MeteorologyFunction Struct Reference	42
11.22.1 Member Data Documentation	42
11.22.1.1 siteName	42

11.22.1.2 table	42
11.23observation Struct Reference	42
11.23.1 Detailed Description	44
11.23.2 Member Data Documentation	44
11.23.2.1 addedNoise	44
11.23.2.2 averagebat	44
11.23.2.3 averageerr	44
11.23.2.4 averageres	45
11.23.2.5 bat	45
11.23.2.6 batCorr	45
11.23.2.7 bbat	45
11.23.2.8 clockCorr	45
11.23.2.9 correctionsTT	45
11.23.2.10correctionTT_TB	45
11.23.2.11correctionTT_Teph	45
11.23.2.12correctionUT1	45
11.23.2.13delayCorr	45
11.23.2.14deleted	45
11.23.2.15earth_ssb	45
11.23.2.16earthMoonBary_earth	45
11.23.2.17earthMoonBary_ssb	46
11.23.2.18efac	46
11.23.2.19einsteinRate	46
11.23.2.20equad	46
11.23.2.21flagID	46
11.23.2.22flagVal	46
11.23.2.23fname	46
11.23.2.24freq	46
11.23.2.25freqSSB	46
11.23.2.26jump	46
11.23.2.27jupiter_earth	46
11.23.2.28nclock_correction	46
11.23.2.29neptune_earth	46
11.23.2.30nFlags	47
11.23.2.31nphase	47
11.23.2.32nutations	47
11.23.2.33observatory_earth	47
11.23.2.34obsNjump	47
11.23.2.35origErr	47
11.23.2.36rigsat	47

11.23.2.37pet	47
11.23.2.38phase	47
11.23.2.39phaseOffset	47
11.23.2.40planet_ssb	47
11.23.2.41prefitResidual	47
11.23.2.42psrPos	47
11.23.2.43pulseN	47
11.23.2.44residual	47
11.23.2.45roemer	48
11.23.2.46sat	48
11.23.2.47sat_day	48
11.23.2.48sat_sec	48
11.23.2.49saturn_earth	48
11.23.2.50shapiroDelayJupiter	48
11.23.2.51shapiroDelayNeptune	48
11.23.2.52shapiroDelaySaturn	48
11.23.2.53shapiroDelaySun	48
11.23.2.54shapiroDelayUranus	48
11.23.2.55shapiroDelayVenus	48
11.23.2.56shklovskii	48
11.23.2.57siteVel	48
11.23.2.58sun_earth	49
11.23.2.59sun_ssb	49
11.23.2.60dis1	49
11.23.2.61tdis2	49
11.23.2.62elID	49
11.23.2.63TNDMErr	49
11.23.2.64TNDMSignal	49
11.23.2.65TNGroupErr	49
11.23.2.66TNGroupSignal	49
11.23.2.67TNRedErr	49
11.23.2.68TNRedSignal	49
11.23.2.69toaDMErr	49
11.23.2.70toaErr	50
11.23.2.71torb	50
11.23.2.72troposphericDelay	50
11.23.2.73uranus_earth	50
11.23.2.74venus_earth	50
11.23.2.75zenith	50
11.24observatory Struct Reference	50

11.24.1 Member Data Documentation	50
11.24.1.1 clock_name	50
11.24.1.2 code	50
11.24.1.3 height_grs80	51
11.24.1.4 latitude_grs80	51
11.24.1.5 longitude_grs80	51
11.24.1.6 name	51
11.24.1.7 x	51
11.24.1.8 y	51
11.24.1.9 z	51
11.25 ObservatoryAliasList Struct Reference	51
11.25.1 Member Data Documentation	51
11.25.1.1 aliases	51
11.25.1.2 code	51
11.26 parameter Struct Reference	52
11.26.1 Detailed Description	52
11.26.2 Member Data Documentation	52
11.26.2.1 aSize	52
11.26.2.2 err	52
11.26.2.3 fitFlag	52
11.26.2.4 label	52
11.26.2.5 linkFrom	53
11.26.2.6 linkTo	53
11.26.2.7 nLinkFrom	53
11.26.2.8 nLinkTo	53
11.26.2.9 paramSet	53
11.26.2.10 pfit	53
11.26.2.11 pfitErr	53
11.26.2.12 shortlabel	53
11.26.2.13 val	53
11.27 pulsar Struct Reference	53
11.27.1 Detailed Description	59
11.27.2 Member Data Documentation	60
11.27.2.1 addTNGlobalEQ	60
11.27.2.2 auto_constraints	60
11.27.2.3 AverageEpochWidth	60
11.27.2.4 AverageFlag	60
11.27.2.5 AverageResiduals	60
11.27.2.6 binaryModel	60
11.27.2.7 bootStrap	60

11.27.2.8 calcShapiro	60
11.27.2.9 cgw_angpol	60
11.27.2.10 cgw_cosinc	60
11.27.2.11 cgw_h0	60
11.27.2.12 cgw_mc	60
11.27.2.13 clk_offsE	60
11.27.2.14 clk_offsT	60
11.27.2.15 clk_offsV	60
11.27.2.16 clkOffsN	60
11.27.2.17 clock	60
11.27.2.18 clockFromOverride	60
11.27.2.19 constraints	60
11.27.2.20 correctTroposphere	61
11.27.2.21 covar	61
11.27.2.22 decjStrPost	61
11.27.2.23 decjStrPre	61
11.27.2.24 decsim	61
11.27.2.25 deleteFileName	61
11.27.2.26 dilateFreq	61
11.27.2.27 dmoffsCM	61
11.27.2.28 dmoffsCM_error	61
11.27.2.29 dmoffsCM_mjd	61
11.27.2.30 dmoffsCM_weight	61
11.27.2.31 dmoffsCMnum	61
11.27.2.32 dmoffsDM	61
11.27.2.33 dmoffsDM_error	61
11.27.2.34 dmoffsDM_mjd	61
11.27.2.35 dmoffsDM_weight	61
11.27.2.36 dmoffsDMnum	61
11.27.2.37 dmOffset	61
11.27.2.38 eclCoord	61
11.27.2.39 eopc04_file	62
11.27.2.40 ephemeris	62
11.27.2.41 filterStr	62
11.27.2.42 fitChisq	62
11.27.2.43 fitFunc	62
11.27.2.44 fitinfo	62
11.27.2.45 fitJump	62
11.27.2.46 fitMode	62
11.27.2.47 fitNfree	62

11.27.2.48fitParamGlobal	62
11.27.2.49fitParamGlobalK	62
11.27.2.50fitParamI	62
11.27.2.51fitParamK	62
11.27.2.52fixedFormat	62
11.27.2.53jumpID	62
11.27.2.54globalNfit	62
11.27.2.55globalNoConstrain	62
11.27.2.56gwb_decj	63
11.27.2.57gwb_epoch	63
11.27.2.58gwb_geom_c	63
11.27.2.59gwb_geom_p	63
11.27.2.60gwb_raj	63
11.27.2.61gwb_width	63
11.27.2.62gwecc_dec	63
11.27.2.63gwecc_distance	63
11.27.2.64gwecc_e	63
11.27.2.65gwecc_epoch	63
11.27.2.66gwecc_inc	63
11.27.2.67gwecc_m1	63
11.27.2.68gwecc_m2	63
11.27.2.69gwecc_nodes_orientation	63
11.27.2.70gwecc_orbital_period	63
11.27.2.71gwecc_psrdist	63
11.27.2.72gwecc_pulsarTermOn	63
11.27.2.73gwecc_ra	63
11.27.2.74gwecc_redshift	63
11.27.2.75gwecc_theta_0	63
11.27.2.76gwecc_theta_nodes	63
11.27.2.77gwm_decj	63
11.27.2.78gwm_dphase	63
11.27.2.79gwm_epoch	63
11.27.2.80gwm_phi	63
11.27.2.81gwm_raj	63
11.27.2.82gwsrsrc_across_i	63
11.27.2.83gwsrsrc_across_i_e	63
11.27.2.84gwsrsrc_across_r	64
11.27.2.85gwsrsrc_across_r_e	64
11.27.2.86gwsrsrc_aplus_i	64
11.27.2.87gwsrsrc_aplus_i_e	64

11.27.2.88	gwsrsrc_aplus_r	64
11.27.2.89	gwsrsrc_aplus_r_e	64
11.27.2.90	gwsrsrc_dec	64
11.27.2.91	gwsrsrc_epoch	64
11.27.2.92	gwsrsrc_psrdist	64
11.27.2.93	gwsrsrc_ra	64
11.27.2.94	func_weights	64
11.27.2.95	funcE	64
11.27.2.96	funcN	64
11.27.2.97	funcT	64
11.27.2.98	funcV	64
11.27.2.99	gpm	64
11.27.2.100	boFormat	64
11.27.2.101	WPL_EPHEMERIS	64
11.27.2.102	ampStr	64
11.27.2.103	ampVal	64
11.27.2.104	ampValErr	64
11.27.2.105	same	65
11.27.2.106	Companion	65
11.27.2.107	constraints	65
11.27.2.108	DMEvents	65
11.27.2.109	dmx	65
11.27.2.110	de_sw	65
11.27.2.111	Fit	65
11.27.2.112	Global	65
11.27.2.113	fits	65
11.27.2.114	Jumps	65
11.27.2.115	obs	65
11.27.2.116	Warnings	65
11.27.2.117	Param	65
11.27.2.118	PhaseJump	66
11.27.2.119	Quad	66
11.27.2.120	StorePrecision	66
11.27.2.121	T2efac	66
11.27.2.122	T2equad	66
11.27.2.123	TelDX	66
11.27.2.124	TelDY	66
11.27.2.125	TelDZ	66
11.27.2.126	TNBandNoise	66
11.27.2.127	TNECORR	66

11.27.2.128	TNEF	66
11.27.2.129	TNEQ	66
11.27.2.130	TNGroupNoise	66
11.27.2.131	TNShapeletEvents	66
11.27.2.132	TNSQ	66
11.27.2.133	TOffset	66
11.27.2.134	White	66
11.27.2.135	White_dm	66
11.27.2.136	bsn	66
11.27.2.137	ffset	66
11.27.2.138	ffset_e	66
11.27.2.139	OutputTMatrix	66
11.27.2.140	param	66
11.27.2.141	ClassStr	67
11.27.2.142	PhaseJump	67
11.27.2.143	PhaseJumpDir	67
11.27.2.144	PhaseJumpID	67
11.27.2.145	PlanetShapiro	67
11.27.2.146	PosPulsar	67
11.27.2.147	quad_across_i	67
11.27.2.148	quad_across_i_e	67
11.27.2.149	quad_across_r	67
11.27.2.150	quad_across_r_e	67
11.27.2.151	quad_aplus_i	67
11.27.2.152	quad_aplus_i_e	67
11.27.2.153	quad_aplus_r	67
11.27.2.154	quad_aplus_r_e	67
11.27.2.155	quad_ifunc_c_DEC	67
11.27.2.156	quad_ifunc_c_RA	67
11.27.2.157	quad_ifunc_geom_c	67
11.27.2.158	quad_ifunc_geom_p	67
11.27.2.159	quad_ifunc_p_DEC	67
11.27.2.160	quad_ifunc_p_RA	67
11.27.2.161	quad_ifuncE_c	68
11.27.2.162	quad_ifuncE_p	68
11.27.2.163	quad_ifuncN_c	68
11.27.2.164	quad_ifuncN_p	68
11.27.2.165	quad_ifuncT_c	68
11.27.2.166	quad_ifuncT_p	68
11.27.2.167	quad_ifuncV_c	68

11.27.2.169	quad_ifuncV_p	68
11.27.2.169	quadDEC	68
11.27.2.170	quadEpoch	68
11.27.2.171	quadRA	68
11.27.2.172	qjStrPost	68
11.27.2.173	qjStrPre	68
11.27.2.174	rsim	68
11.27.2.175	rscaleErrChisq	68
11.27.2.176	rsnsPost	68
11.27.2.177	rsnsPre	68
11.27.2.178	rsnust	68
11.27.2.179	setTelVelX	68
11.27.2.180	setTelVelY	68
11.27.2.181	setTelVelZ	68
11.27.2.182	setUnits	68
11.27.2.183	simflag	68
11.27.2.184	sorted	68
11.27.2.185	storePrec	69
11.27.2.186	svm	69
11.27.2.187	tcMethod	69
11.27.2.188	tefacFlagID	69
11.27.2.189	tefacFlagVal	69
11.27.2.190	tefacVal	69
11.27.2.191	tequadFlagID	69
11.27.2.192	tequadFlagVal	69
11.27.2.193	tequadVal	69
11.27.2.194	teglobalEfac	69
11.27.2.195	teDX_e	69
11.27.2.196	teDX_t	69
11.27.2.197	teDX_v	69
11.27.2.198	teDX_vel	69
11.27.2.199	teDX_vel_e	69
11.27.2.200	teDY_e	69
11.27.2.201	teDY_t	69
11.27.2.202	teDY_v	69
11.27.2.203	teDY_vel	69
11.27.2.204	teDY_vel_e	69
11.27.2.205	teDZ_e	69
11.27.2.206	teDZ_t	69
11.27.2.207	teDZ_v	69

11.27.2.2031DZ_vel	69
11.27.2.2031DZ_vel_e	69
11.27.2.2110mpo1	70
11.27.2.2111meEphemeris	70
11.27.2.2112NBandDMamp	70
11.27.2.2113NBandDMC	70
11.27.2.2114NBandDMGam	70
11.27.2.2115NBandNoiseAmp	70
11.27.2.2116NBandNoiseC	70
11.27.2.2117NBandNoiseGam	70
11.27.2.2118NBandNoiseHF	70
11.27.2.2119NBandNoiseLF	70
11.27.2.2210NDMAmp	70
11.27.2.2211NDMC	70
11.27.2.2212NDMCoeffs	70
11.27.2.2213NDMEvAmp	70
11.27.2.2214NDMEvGam	70
11.27.2.2215NDMEvLength	70
11.27.2.2216NDMEvLin	70
11.27.2.2217NDMEvOff	70
11.27.2.2218NDMEvQuad	70
11.27.2.2219NDMEvStart	70
11.27.2.2310NDMGam	70
11.27.2.2311NECORRFlagID	70
11.27.2.2312NECORRFlagVal	70
11.27.2.2313NECORRVal	70
11.27.2.2314NEFFFlagID	70
11.27.2.2315NEFFFlagVal	70
11.27.2.2316NEFVal	71
11.27.2.2317NEQFlagID	71
11.27.2.2318NEQFlagVal	71
11.27.2.2319NEQVal	71
11.27.2.2410NGlobalEF	71
11.27.2.2411NGlobalEQ	71
11.27.2.2412NGroupNoiseAmp	71
11.27.2.2413NGroupNoiseC	71
11.27.2.2414NGroupNoiseFlagID	71
11.27.2.2415NGroupNoiseFlagVal	71
11.27.2.2416NGroupNoiseGam	71
11.27.2.2417NRedAmp	71

11.27.2.241BNRedC	71
11.27.2.241BNRedCoeffs	71
11.27.2.251BNRedCorner	71
11.27.2.251BNRedFlow	71
11.27.2.252BNRedGam	71
11.27.2.253BNShapeletEvFScale	71
11.27.2.253BNShapeletEvN	71
11.27.2.253BNShapeletEvPos	71
11.27.2.253BNShapeletEvWidth	71
11.27.2.257BNSQFlagID	71
11.27.2.257BNSQFlagVal	71
11.27.2.257BNSQVal	71
11.27.2.261BNsubtractDM	71
11.27.2.261BNsubtractRed	71
11.27.2.261BNBextraCovar	71
11.27.2.263BOffset	72
11.27.2.263BOffset_f1	72
11.27.2.263BOffset_f2	72
11.27.2.263BOffset_t1	72
11.27.2.263BOffset_t2	72
11.27.2.263BOffsetFlags	72
11.27.2.263BOffsetSite	72
11.27.2.270Borsite	72
11.27.2.271Bunits	72
11.27.2.272BseCalceph	72
11.27.2.272BseTNOrth	72
11.27.2.274Belpulsar	72
11.27.2.275Bave_cos	72
11.27.2.275Bave_cos_dm	72
11.27.2.275Bave_cos_dm_err	72
11.27.2.275Bave_cos_err	72
11.27.2.275Bave_sine	72
11.27.2.280Bave_sine_dm	72
11.27.2.280Bave_sine_dm_err	72
11.27.2.282Bave_sine_err	72
11.27.2.283BaveScale	72
11.27.2.284BwhiteNoiseModelFile	73
11.28sample Struct Reference	73
11.28.1 Member Data Documentation	73
11.28.1.1 actual	73

11.28.1.2 e	73
11.28.1.3 pred	73
11.28.1.4 x	73
11.28.1.5 y	73
11.29storePrecision Struct Reference	73
11.29.1 Member Data Documentation	73
11.29.1.1 comment	73
11.29.1.2 minPrec	73
11.29.1.3 routine	73
11.30T1Polyco Struct Reference	74
11.30.1 Member Data Documentation	74
11.30.1.1 binary_frequency	74
11.30.1.2 binary_phase	74
11.30.1.3 coeff	74
11.30.1.4 date_string	74
11.30.1.5 dm	74
11.30.1.6 doppler	74
11.30.1.7 frequency_obs	74
11.30.1.8 frequency_psr_0	74
11.30.1.9 log10rms	74
11.30.1.10mjd_mid	74
11.30.1.11ncoeff	74
11.30.1.12psrname	74
11.30.1.13reference_phase	74
11.30.1.14sitename	75
11.30.1.15span	75
11.30.1.16utc_string	75
11.31T1PolycoSet Struct Reference	75
11.31.1 Member Data Documentation	75
11.31.1.1 nsegments	75
11.31.1.2 segments	75
11.32T2Predictor Struct Reference	75
11.32.1 Member Data Documentation	76
11.32.1.1 cheby	76
11.32.1.2 kind	76
11.32.1.3 modelset	76
11.32.1.4 t1	76
11.33TabulatedFunction Struct Reference	77
11.33.1 Member Data Documentation	77
11.33.1.1 fileName	77

11.33.1.2 header_line	77
11.33.1.3 samples	77
11.34 TabulatedFunctionSample Struct Reference	77
11.34.1 Member Data Documentation	77
11.34.1.1 x	77
11.34.1.2 y	78
11.35 XY Struct Reference	78
11.35.1 Member Data Documentation	78
11.35.1.1 x	78
11.35.1.2 y	78
12 File Documentation	79
12.1 age.C File Reference	79
12.1.1 Function Documentation	79
12.1.1.1 calc_age	79
12.1.1.2 derivs	79
12.1.1.3 rk4rms	79
12.2 bootstrap.C File Reference	80
12.2.1 Macro Definition Documentation	80
12.2.1.1 MAX_ITER	80
12.2.2 Function Documentation	80
12.2.2.1 bootstrap	80
12.2.2.2 random	80
12.3 BTJmodel.C File Reference	80
12.3.1 Function Documentation	81
12.3.1.1 BTJmodel	81
12.3.1.2 updateBTJ	81
12.4 BTmodel.C File Reference	81
12.4.1 Function Documentation	82
12.4.1.1 BTmodel	82
12.4.1.2 updateBT	82
12.5 BTXmodel.C File Reference	82
12.5.1 Function Documentation	83
12.5.1.1 BTXmodel	83
12.5.1.2 updateBTX	83
12.6 calculate_bclt.C File Reference	83
12.6.1 Function Documentation	84
12.6.1.1 calculate_bclt	84
12.7 cheby2d.c File Reference	84
12.7.1 Macro Definition Documentation	86

12.7.1.1	M_PII	86
12.7.2	Function Documentation	86
12.7.2.1	Cheby2D_Construct	86
12.7.2.2	Cheby2D_Construct_x_Derivative	86
12.7.2.3	Cheby2D_Copy	86
12.7.2.4	Cheby2D_Destroy	86
12.7.2.5	Cheby2D_Evaluate	86
12.7.2.6	Cheby2D_Init	86
12.7.2.7	Cheby2D_Test	86
12.7.2.8	ChebyModel_Copy	86
12.7.2.9	ChebyModel_Destroy	86
12.7.2.10	ChebyModel_GetFrequency	86
12.7.2.11	ChebyModel_GetPhase	86
12.7.2.12	ChebyModel_Init	86
12.7.2.13	ChebyModel_Read	86
12.7.2.14	ChebyModel_Write	86
12.7.2.15	ChebyModelSet_Destroy	86
12.7.2.16	ChebyModelSet_GetFrequency	86
12.7.2.17	ChebyModelSet_GetNearest	86
12.7.2.18	ChebyModelSet_GetNearestIndex	86
12.7.2.19	ChebyModelSet_GetPhase	86
12.7.2.20	ChebyModelSet_Init	86
12.7.2.21	ChebyModelSet_Insert	86
12.7.2.22	ChebyModelSet_Keep	86
12.7.2.23	ChebyModelSet_Read	87
12.7.2.24	ChebyModelSet_Write	87
12.7.2.25	testCheby2D	87
12.7.2.26	testFunc	87
12.7.3	Variable Documentation	87
12.7.3.1	ChebyModelSet_OutOfRange	87
12.8	cheby2d_int.C File Reference	87
12.8.1	Function Documentation	88
12.8.1.1	ChebyModel_Construct	88
12.8.1.2	ChebyModel_Test	88
12.8.1.3	chebyModelFunc	88
12.8.1.4	ChebyModelSet_Construct	88
12.8.1.5	ChebyModelSet_Test	88
12.9	cholesky.C File Reference	88
12.9.1	Macro Definition Documentation	89
12.9.1.1	LINE_LENGTH	89

12.9.2	Function Documentation	89
12.9.2.1	addCovar	89
12.9.2.2	cholesky_covarFunc2matrix	89
12.9.2.3	cholesky_dmModel	89
12.9.2.4	cholesky_dmModelCovarParam	89
12.9.2.5	cholesky_ecm	89
12.9.2.6	cholesky_formUinv	89
12.9.2.7	cholesky_powerlawModel	90
12.9.2.8	cholesky_powerlawModel_withBeta	90
12.9.2.9	cholesky_readFromCovarianceFunction	90
12.9.2.10	cholesky_readT2CholModel	90
12.9.2.11	cholesky_readT2CholModel_R	90
12.9.2.12	cholesky_readT2Model1	90
12.9.2.13	cholesky_readT2Model2	90
12.9.2.14	getCholeskyDiagonals	90
12.9.2.15	getCholeskyMatrix	90
12.10	cholesky.h File Reference	90
12.10.1	Function Documentation	91
12.10.1.1	cholesky_covarFunc2matrix	91
12.10.1.2	cholesky_dmModel	91
12.10.1.3	cholesky_dmModelCovarParam	91
12.10.1.4	cholesky_ecm	91
12.10.1.5	cholesky_formUinv	91
12.10.1.6	cholesky_powerlawModel	91
12.10.1.7	cholesky_powerlawModel_withBeta	91
12.10.1.8	cholesky_readFromCovarianceFunction	91
12.11	choleskyAutomatic.C File Reference	91
12.11.1	Function Documentation	92
12.11.1.1	T2get_covFunc_automatic	92
12.12	choleskyRoutines.C File Reference	92
12.12.1	Function Documentation	93
12.12.1.1	T2calculateCholesky	93
12.12.1.2	T2calculateCovarFunc	93
12.12.1.3	T2calculateDailyCovariance	94
12.12.1.4	T2calculateSpectra	94
12.12.1.5	T2cholDecomposition	94
12.12.1.6	T2cubicFit	94
12.12.1.7	T2findSmoothCurve	94
12.12.1.8	T2fitSpectra	94
12.12.1.9	T2getHighFreqRes	94

12.12.1.10T2getWhiteNoiseLevel	94
12.12.1.11T2getWhiteRes	94
12.12.1.12T2guess_vals	94
12.12.1.13T2interpolate	94
12.12.1.14T2obtainTimingResiduals	94
12.12.1.15T2writeCovarFuncModel	94
12.13choleskyRoutines.h File Reference	94
12.13.1 Function Documentation	96
12.13.1.1 T2calculateCholesky	96
12.13.1.2 T2calculateCovarFunc	96
12.13.1.3 T2calculateDailyCovariance	96
12.13.1.4 T2calculateSpectra	96
12.13.1.5 T2cholDecomposition	96
12.13.1.6 T2cubicFit	96
12.13.1.7 T2findSmoothCurve	96
12.13.1.8 T2fitSpectra	96
12.13.1.9 T2get_covFunc_automatic	96
12.13.1.10T2getHighFreqRes	96
12.13.1.11T2getWhiteNoiseLevel	96
12.13.1.12T2getWhiteRes	96
12.13.1.13T2guess_vals	97
12.13.1.14T2interpolate	97
12.13.1.15T2obtainTimingResiduals	97
12.13.1.16T2writeCovarFuncModel	97
12.13.2 Variable Documentation	97
12.13.2.1 EXPSMOOTH	97
12.13.2.2 FCALPHA	97
12.13.2.3 FCFINAL	97
12.13.2.4 NFIT	97
12.13.2.5 UPW	97
12.13.2.6 WNLEVEL	97
12.14clkcorr.C File Reference	97
12.14.1 Function Documentation	98
12.14.1.1 ClockCorrectionFunction_getCorrection	98
12.14.1.2 ClockCorrectionFunction_getEndMJD	98
12.14.1.3 ClockCorrectionFunction_getStartMJD	98
12.14.1.4 ClockCorrectionFunction_load	98
12.14.1.5 ClockCorrectionSequence_getEndMJD	98
12.14.1.6 ClockCorrectionSequence_getStartMJD	98
12.14.1.7 defineClockCorrectionSequence	98

12.14.1.8	getClockCorrections	98
12.14.1.9	getClockCorrectionSequence	98
12.14.1.10	getCorrection	98
12.14.1.11	getCorrectionTT	98
12.14.1.12	initialize_ClockCorrections	98
12.14.1.13	makeClockCorrectionSequence	98
12.14.2	Variable Documentation	99
12.14.2.1	clockCorrectionFunctions	99
12.14.2.2	clockCorrectionSequences	99
12.15	config.h File Reference	99
12.15.1	Macro Definition Documentation	100
12.15.1.1	_DARWIN_USE_64_BIT_INODE	100
12.15.1.2	F77_FUNC	100
12.15.1.3	F77_FUNC_	100
12.15.1.4	HAVE_BLAS	100
12.15.1.5	HAVE_DLERROR	100
12.15.1.6	HAVE_DLFCN_H	100
12.15.1.7	HAVE_FFTW3	100
12.15.1.8	HAVE_INTTYPES_H	100
12.15.1.9	HAVE_LAPACK	100
12.15.1.10	HAVE_LIBDL	100
12.15.1.11	HAVE_LIBDLLOADER	100
12.15.1.12	HAVE_LIBM	100
12.15.1.13	HAVE_MEMORY_H	100
12.15.1.14	HAVE_PGPLOT	100
12.15.1.15	HAVE_PTHREAD	100
12.15.1.16	HAVE_STDINT_H	100
12.15.1.17	HAVE_STDLIB_H	100
12.15.1.18	HAVE_STRING_H	100
12.15.1.19	HAVE_STRINGS_H	100
12.15.1.20	HAVE_SYS_STAT_H	100
12.15.1.21	HAVE_SYS_TYPES_H	100
12.15.1.22	HAVE_UNISTD_H	100
12.15.1.23	LT_OBJDIR	100
12.15.1.24	PACKAGE	100
12.15.1.25	PACKAGE_BUGREPORT	101
12.15.1.26	PACKAGE_NAME	101
12.15.1.27	PACKAGE_STRING	101
12.15.1.28	PACKAGE_TARNAME	101
12.15.1.29	PACKAGE_URL	101

12.15.1.30	PACKAGE_VERSION	101
12.15.1.31	STDC_HEADERS	101
12.15.1.32	TEMPO2_ARCH	101
12.15.1.33	VERSION	101
12.15.1.34	X_DISPLAY_MISSING	101
12.16	constraints.C File Reference	101
12.16.1	Function Documentation	102
12.16.1.1	autoConstraints	102
12.16.1.2	autosetDMCM	102
12.16.1.3	computeConstraintWeights	102
12.16.1.4	consFunc_dmmodel_cw	102
12.16.1.5	consFunc_dmmodel_cw_year	102
12.16.1.6	consFunc_dmmodel_dm1	102
12.16.1.7	consFunc_dmmodel_mean	102
12.16.1.8	consFunc_ifunc	102
12.16.1.9	consFunc_ifunc_year	102
12.16.1.10	consFunc_qifunc_c_year	102
12.16.1.11	consFunc_qifunc_p_year	102
12.16.1.12	consFunc_quad_ifunc_c	102
12.16.1.13	consFunc_quad_ifunc_p	102
12.16.1.14	consFunc_tel_dx	102
12.16.1.15	consFunc_tel_dy	102
12.16.1.16	consFunc_tel_dz	102
12.16.1.17	CONSTRAINTfuncs	102
12.16.1.18	get_constraint_name	102
12.16.1.19	get_constraint_name	103
12.16.1.20	getConstraintDeriv	103
12.16.1.21	matrixDMConstraintWeights	103
12.16.1.22	standardConstraintFunctions	103
12.17	constraints.h File Reference	103
12.17.1	Function Documentation	104
12.17.1.1	autosetDMCM	104
12.17.1.2	computeConstraintWeights	104
12.17.1.3	consFunc_dmmodel_cw	104
12.17.1.4	consFunc_dmmodel_cw_year	104
12.17.1.5	consFunc_dmmodel_dm1	104
12.17.1.6	consFunc_dmmodel_mean	104
12.17.1.7	consFunc_ifunc	104
12.17.1.8	consFunc_ifunc_year	104
12.17.1.9	consFunc_qifunc_c_year	104

12.17.1.10	consFunc_qifunc_p_year	104
12.17.1.11	consFunc_quad_ifunc_c	104
12.17.1.12	consFunc_quad_ifunc_p	104
12.17.1.13	consFunc_tel_dx	104
12.17.1.14	consFunc_tel_dy	104
12.17.1.15	consFunc_tel_dz	104
12.17.1.16	CONSTRAINTfuncs	104
12.17.1.17	get_constraint_name	104
12.17.1.18	standardConstraintFunctions	104
12.18	DDGRmodel.C File Reference	104
12.18.1	Function Documentation	105
12.18.1.1	DDGRmodel	105
12.18.1.2	mass2dd	105
12.18.1.3	updateDDGR	105
12.19	DDHmodel.C File Reference	105
12.19.1	Function Documentation	106
12.19.1.1	DDHmodel	106
12.19.1.2	updateDDH	106
12.20	DDKmodel.C File Reference	106
12.20.1	Function Documentation	107
12.20.1.1	DDKmodel	107
12.20.1.2	updateDDK	107
12.21	DDmodel.C File Reference	107
12.21.1	Function Documentation	108
12.21.1.1	DDmodel	108
12.21.1.2	updateDD	108
12.22	DDSmodel.C File Reference	108
12.22.1	Function Documentation	109
12.22.1.1	DDSmodel	109
12.22.1.2	updateDDS	109
12.23	displayParameters.C File Reference	109
12.23.1	Function Documentation	110
12.23.1.1	displayParameters	110
12.24	dm_delays.C File Reference	110
12.24.1	Function Documentation	111
12.24.1.1	dm_delays	111
12.24.1.2	solarWindModel	111
12.25	documentation/DEVELOPER_GUIDE.md File Reference	111
12.26	documentation/developers.md File Reference	111
12.27	documentation/directories.md File Reference	111

12.28documentation/USER_GUIDE.md File Reference	111
12.29doFit.C File Reference	111
12.29.1 Function Documentation	113
12.29.1.1 dgemm	113
12.29.1.2 dgemm_	113
12.29.1.3 dgemm_ctof	113
12.29.1.4 dgemm_ftoc	113
12.29.1.5 dgemv	113
12.29.1.6 dgemv_	113
12.29.1.7 dgemv_ctof	113
12.29.1.8 dgemv_ftoc	113
12.29.1.9 dgesvd	113
12.29.1.10dgesvd_	113
12.29.1.11dgesvd_ctof	113
12.29.1.12dgesvd_ftoc	113
12.29.1.13doFit	113
12.29.1.14doFitAll	113
12.29.1.15doFitDCM	113
12.29.1.16doFitOLD	113
12.29.1.17dpotrf	114
12.29.1.18dpotrf_	114
12.29.1.19dpotrf_ctof	114
12.29.1.20dpotrf_ftoc	114
12.29.1.21dpotri	114
12.29.1.22dpotri_	114
12.29.1.23dpotri_ctof	114
12.29.1.24dpotri_ftoc	114
12.29.1.25FITfuncs	114
12.29.1.26getConstraintDeriv	114
12.29.1.27getNglobal	114
12.29.1.28getNparams	114
12.29.1.29getParamDeriv	114
12.29.1.30getTempoNestMaxLike	114
12.29.1.31globalFITfuncs	114
12.29.1.32bthpl	114
12.29.1.33updateGlobalParameters	114
12.29.1.34updateParameters	114
12.30dynarr.C File Reference	114
12.30.1 Function Documentation	115
12.30.1.1 DynamicArray_free	115

12.30.1.2 DynamicArray_init	115
12.30.1.3 DynamicArray_push_back	115
12.30.1.4 DynamicArray_resize	115
12.31 dynarr.h File Reference	116
12.31.1 Function Documentation	116
12.31.1.1 DynamicArray_free	116
12.31.1.2 DynamicArray_init	116
12.31.1.3 DynamicArray_push_back	116
12.31.1.4 DynamicArray_resize	116
12.32 ELL1Hmodel.C File Reference	117
12.32.1 Function Documentation	117
12.32.1.1 ELL1Hmodel	117
12.32.1.2 updateELL1H	117
12.33 ELL1model.C File Reference	117
12.33.1 Function Documentation	118
12.33.1.1 ELL1model	118
12.33.1.2 updateELL1	118
12.34 eop.C File Reference	118
12.34.1 Function Documentation	119
12.34.1.1 get_EOP	119
12.34.1.2 load_EOP	119
12.35 formBats.C File Reference	119
12.35.1 Function Documentation	120
12.35.1.1 formBats	120
12.36 formResiduals.C File Reference	120
12.36.1 Function Documentation	121
12.36.1.1 averageResiduals	121
12.36.1.2 formResiduals	121
12.36.1.3 residualTracking	121
12.37 get_obsCoord.C File Reference	121
12.37.1 Function Documentation	122
12.37.1.1 ang	122
12.37.1.2 get_obsCoord	122
12.37.1.3 get_obsCoord_IAU2000B	122
12.37.1.4 get_precessionMatrix	122
12.37.1.5 iau_c2t00b_	122
12.37.1.6 iau_cp_	123
12.37.1.7 iau_pom00_	123
12.37.1.8 iau_pxp_	123
12.37.1.9 iau_rxp_	123

12.37.1.10	<code>au_sxp_</code>	123
12.37.1.11	<code>iau_trxp_</code>	123
12.37.1.12	<code>au_trxpv_</code>	123
12.37.1.13	<code>mst</code>	123
12.37.1.14	<code>remove_white</code>	123
12.38	<code>getInputs.C</code> File Reference	123
12.38.1	Function Documentation	123
12.38.1.1	<code>getInputs</code>	124
12.38.1.2	<code>printplugs</code>	124
12.38.1.3	<code>setPlugPath</code>	124
12.39	<code>getPeriod.C</code> File Reference	124
12.39.1	Function Documentation	124
12.39.1.1	<code>main</code>	124
12.40	<code>global.C</code> File Reference	124
12.40.1	Macro Definition Documentation	126
12.40.1.1	<code>MAX_FUNCTIONS</code>	126
12.40.2	Function Documentation	126
12.40.2.1	<code>clock_corrections</code>	126
12.40.2.2	<code>CVSdisplayVersion</code>	126
12.40.2.3	<code>ephemeris_routines</code>	126
12.40.2.4	<code>extra_delays</code>	126
12.40.2.5	<code>formBatsAll</code>	126
12.40.2.6	<code>updateBatsAll</code>	126
12.40.3	Variable Documentation	126
12.40.3.1	<code>covarFuncFile</code>	126
12.40.3.2	<code>dcmFile</code>	126
12.40.3.3	<code>displayCVSversion</code>	126
12.40.3.4	<code>ECLIPTIC_OBLIQUITY</code>	126
12.40.3.5	<code>EXPSMOOTH</code>	126
12.40.3.6	<code>FCALPHA</code>	126
12.40.3.7	<code>FCFINAL</code>	126
12.40.3.8	<code>forceGlobalFit</code>	126
12.40.3.9	<code>MAX_OBSN</code>	126
12.40.3.10	<code>MAX_PSR</code>	126
12.40.3.11	<code>NEWFIT</code>	127
12.40.3.12	<code>NFIT</code>	127
12.40.3.13	<code>TEMPO2_ENVIRON</code>	127
12.40.3.14	<code>TEMPO2_ERROR</code>	127
12.40.3.15	<code>tempo2_plug_path</code>	127
12.40.3.16	<code>tempo2_plug_path_len</code>	127

12.40.3.17tempo2MachineType	127
12.40.3.18JPW	127
12.40.3.19veryFast	127
12.40.3.20WNLEVEL	127
12.41GWsim.C File Reference	127
12.41.1 Function Documentation	129
12.41.1.1 calculateResidualgeneralGW	129
12.41.1.2 calculateResidualGW	129
12.41.1.3 dadt	129
12.41.1.4 dedt	129
12.41.1.5 dotProduct	129
12.41.1.6 dtdt	129
12.41.1.7 eccRes	129
12.41.1.8 eccResWithEnergy	129
12.41.1.9 Fe	129
12.41.1.10Findphi	129
12.41.1.11GWanisotropicbackground	129
12.41.1.12GWbackground	129
12.41.1.13GWbackground_read	129
12.41.1.14GWbackground_write	129
12.41.1.15GWdipolebackground	129
12.41.1.16GWgeneralanisotropicbackground	129
12.41.1.17GWgeneralbackground	129
12.41.1.18GWgeneralbackground_read	129
12.41.1.19GWgeneralbackground_write	130
12.41.1.20matrixMult	130
12.41.1.21psrangle	130
12.41.1.22Rs	130
12.41.1.23setupgeneralGW	130
12.41.1.24setupGW	130
12.41.1.25setupPulsar_GWsim	130
12.41.1.26spharm	130
12.41.2 Variable Documentation	130
12.41.2.1 gwsim_Ngrid	130
12.42GWsim.h File Reference	130
12.42.1 Typedef Documentation	132
12.42.1.1 gwgeneralSrc	132
12.42.1.2 gwgenSpec	132
12.42.1.3 gwSrc	132
12.42.2 Function Documentation	132

12.42.2.1 calculateResidualgeneralGW	132
12.42.2.2 calculateResidualGW	132
12.42.2.3 dadt	132
12.42.2.4 dedt	132
12.42.2.5 dotProduct	132
12.42.2.6 dtdt	132
12.42.2.7 eccRes	132
12.42.2.8 eccResWithEnergy	132
12.42.2.9 Fe	132
12.42.2.10Findphi	132
12.42.2.11GWanisotropicbackground	132
12.42.2.12GWbackground	132
12.42.2.13GWbackground_read	132
12.42.2.14GWbackground_write	132
12.42.2.15GWdipolebackground	132
12.42.2.16GWgeneralanisotropicbackground	132
12.42.2.17GWgeneralbackground	132
12.42.2.18GWgeneralbackground_read	132
12.42.2.19GWgeneralbackground_write	132
12.42.2.20matrixMult	133
12.42.2.21psrange	133
12.42.2.22Rs	133
12.42.2.23setupgeneralGW	133
12.42.2.24setupGW	133
12.42.2.25setupPulsar_GWsim	133
12.42.2.26spharm	133
12.43ifteph.C File Reference	133
12.43.1 Function Documentation	134
12.43.1.1 IFTE_close_file	134
12.43.1.2 IFTE_DeltaT	134
12.43.1.3 IFTE_DeltaTDot	134
12.43.1.4 IFTE_get_DeltaT_DeltaTDot	134
12.43.1.5 IFTE_get_Vals	134
12.43.1.6 IFTE_get_vE	134
12.43.1.7 IFTE_get_vE_vEDot	134
12.43.1.8 IFTE_get_vEDot	134
12.43.1.9 IFTE_init	134
12.43.1.10FTswap4	134
12.43.1.11FTswap8	134
12.43.1.12FTswap8N	134

12.43.1.13	FTswapDouble	134
12.43.1.14	FTswapDoubles	134
12.43.1.15	FTswapInt	134
12.43.1.16	FTswapInts	134
12.44	ifteph.h File Reference	135
12.44.1	Macro Definition Documentation	136
12.44.1.1	IFTE_JD0	136
12.44.1.2	IFTE_K	136
12.44.1.3	IFTE_KM1	136
12.44.1.4	IFTE_LC	136
12.44.1.5	IFTE_MJD0	136
12.44.1.6	IFTE_TEPH0	136
12.44.2	Function Documentation	136
12.44.2.1	IFTE_close_file	136
12.44.2.2	IFTE_DeltaT	136
12.44.2.3	IFTE_DeltaTDot	136
12.44.2.4	IFTE_get_DeltaT_DeltaTDot	136
12.44.2.5	IFTE_get_vE	136
12.44.2.6	IFTE_get_vE_vEDot	136
12.44.2.7	IFTE_get_vEDot	136
12.44.2.8	IFTE_init	136
12.45	initialise.C File Reference	136
12.45.1	Function Documentation	137
12.45.1.1	allocateMemory	137
12.45.1.2	destroyMemory	137
12.45.1.3	destroyOne	137
12.45.1.4	initialise	137
12.45.1.5	initialiseOne	137
12.46	jpl_int.h File Reference	138
12.46.1	Macro Definition Documentation	138
12.46.1.1	JPL_HEADER_SIZE	138
12.46.1.2	MAX_KERNEL_SIZE	138
12.46.2	Typedef Documentation	138
12.46.2.1	JPLlong	138
12.47	jpleph.c File Reference	138
12.47.1	Macro Definition Documentation	139
12.47.1.1	FALSE	139
12.47.1.2	SWAP_MACRO	139
12.47.1.3	TRUE	139
12.47.2	Function Documentation	139

12.47.2.1 jpl_close_ephemeris	139
12.47.2.2 jpl_get_double	139
12.47.2.3 jpl_get_long	139
12.47.2.4 jpl_init_ephemeris	139
12.47.2.5 jpl_pleph	139
12.47.2.6 jpl_state	139
12.48jpleph.h File Reference	140
12.48.1 Macro Definition Documentation	140
12.48.1.1 DLL_FUNC	140
12.48.1.2 JPL_EPHEM_AU_IN_KM	140
12.48.1.3 JPL_EPHEM_EARTH_MOON_RATIO	140
12.48.1.4 JPL_EPHEM_END_JD	141
12.48.1.5 JPL_EPHEM_EPHEMERIS_VERSION	141
12.48.1.6 JPL_EPHEM_KERNEL_NCOEFF	141
12.48.1.7 JPL_EPHEM_KERNEL_RECORD_SIZE	141
12.48.1.8 JPL_EPHEM_KERNEL_SIZE	141
12.48.1.9 JPL_EPHEM_KERNEL_SWAP_BYTES	141
12.48.1.10JPL_EPHEM_N_CONSTANTS	141
12.48.1.11JPL_EPHEM_START_JD	141
12.48.1.12JPL_EPHEM_STEP	141
12.48.2 Function Documentation	141
12.48.2.1 jpl_close_ephemeris	141
12.48.2.2 jpl_get_double	141
12.48.2.3 jpl_get_long	141
12.48.2.4 jpl_init_ephemeris	141
12.48.2.5 jpl_pleph	141
12.48.2.6 jpl_state	141
12.48.2.7 make_sub_ephem	141
12.49MSSmodel.C File Reference	141
12.49.1 Function Documentation	142
12.49.1.1 MSSmodel	142
12.49.1.2 updateMSS	142
12.50observatory.C File Reference	142
12.50.1 Macro Definition Documentation	143
12.50.1.1 GRS80_A	143
12.50.1.2 GRS80_F	143
12.50.2 Function Documentation	143
12.50.2.1 fang	143
12.50.2.2 getObservatory	143
12.50.2.3 GRS80_to_ITRF	143

12.50.2.4 initObservatories	143
12.50.2.5 ITRF_to_GRS80	143
12.50.2.6 lookup_observatory_alias	144
12.50.2.7 readAliases	144
12.50.2.8 readObservatoryFile	144
12.51plugin/add_pulseNumber_plug.C File Reference	144
12.51.1 Function Documentation	144
12.51.1.1 tempoOutput	144
12.52plugin/addRed_plug.C File Reference	144
12.52.1 Function Documentation	145
12.52.1.1 graphicalInterface	145
12.52.1.2 help	145
12.52.2 Variable Documentation	145
12.52.2.1 plugVersionCheck	145
12.53plugin/analyticChol_plug.C File Reference	146
12.53.1 Function Documentation	146
12.53.1.1 graphicalInterface	146
12.53.1.2 help	146
12.53.2 Variable Documentation	146
12.53.2.1 plugVersionCheck	146
12.54plugin/angle_plug.C File Reference	147
12.54.1 Function Documentation	147
12.54.1.1 graphicalInterface	147
12.54.1.2 help	147
12.54.1.3 psrangle	147
12.55plugin/applet_plug.C File Reference	147
12.55.1 Function Documentation	148
12.55.1.1 fortranMod	148
12.55.1.2 graphicalInterface	148
12.55.1.3 help	148
12.55.1.4 nint_derived	148
12.55.1.5 parseLine	148
12.55.1.6 rnd8	148
12.55.2 Variable Documentation	149
12.55.2.1 plugVersionCheck	149
12.56plugin/autoDM_plug.C File Reference	149
12.56.1 Function Documentation	149
12.56.1.1 graphicalInterface	149
12.56.1.2 help	150
12.56.2 Variable Documentation	150

12.56.2.1 plugVersionCheck	150
12.57plugin/autoSpectralFit_plug.C File Reference	150
12.57.1 Macro Definition Documentation	150
12.57.1.1 MAX_FREQ	150
12.57.2 Function Documentation	150
12.57.2.1 graphicalInterface	150
12.57.2.2 help	150
12.58plugin/averageData_plug.C File Reference	151
12.58.1 Macro Definition Documentation	151
12.58.1.1 MAX_TIMES	151
12.58.2 Function Documentation	151
12.58.2.1 graphicalInterface	151
12.58.2.2 help	152
12.58.3 Variable Documentation	152
12.58.3.1 plugVersionCheck	152
12.59plugin/bary_plug.C File Reference	152
12.59.1 Macro Definition Documentation	152
12.59.1.1 GRS80_A	152
12.59.1.2 GRS80_F	152
12.59.2 Function Documentation	153
12.59.2.1 graphicalInterface	153
12.59.2.2 help	153
12.59.2.3 ITRF_to_GRS80	153
12.59.3 Variable Documentation	153
12.59.3.1 plugVersionCheck	153
12.60plugin/basic_plug.C File Reference	153
12.60.1 Function Documentation	153
12.60.1.1 callFit	154
12.60.1.2 graphicalInterface	154
12.60.1.3 help	154
12.60.1.4 plot_ppdot	154
12.60.2 Variable Documentation	154
12.60.2.1 plugVersionCheck	154
12.61plugin/calcDMe_plug.C File Reference	154
12.61.1 Function Documentation	156
12.61.1.1 callFit	156
12.61.1.2 describe	156
12.61.1.3 display	156
12.61.1.4 findFirst	156
12.61.1.5 findMean	156

12.61.1.6 findSessions	156
12.61.1.7 get_binObs	156
12.61.1.8 graphicalInterface	156
12.61.1.9 handleFreqPoints	156
12.61.1.10help	156
12.61.1.11init	156
12.61.1.12InterpolateSplineSmooth	156
12.61.1.13InterpolateWeightedSmooth	156
12.61.1.14output	156
12.61.1.15resetDMandF0	156
12.61.1.16setAllDeleted	157
12.61.1.17setFitParams	157
12.61.2 Variable Documentation	157
12.61.2.1 allParTim	157
12.61.2.2 ascii	157
12.61.2.3 bin_dmCount	157
12.61.2.4 bin_dmCount_inc	157
12.61.2.5 bin_fitCount	157
12.61.2.6 bin_fitCount_inc	157
12.61.2.7 binObs	157
12.61.2.8 binSizeDays	157
12.61.2.9 binStart	157
12.61.2.10bcmFile	157
12.61.2.11ddm	157
12.61.2.12ddmCount	157
12.61.2.13ddmErr	157
12.61.2.14ddmMJD	157
12.61.2.15dm0	157
12.61.2.16dm0_err	157
12.61.2.17dmCount	157
12.61.2.18dmObs	157
12.61.2.19doDisplay	157
12.61.2.20f0_0	157
12.61.2.21f0_0_err	157
12.61.2.22f0fit	157
12.61.2.23finish_sessions	157
12.61.2.24fitCount	157
12.61.2.25fitObs	157
12.61.2.26freq1f	158
12.61.2.27freq2f	158

12.61.2.28freqArray	158
12.61.2.29freqOffset	158
12.61.2.30gotOut	158
12.61.2.31gr	158
12.61.2.32hardcopy	158
12.61.2.33header	158
12.61.2.34impCount	158
12.61.2.35impObs	158
12.61.2.36lastUsedSession	158
12.61.2.37mean	158
12.61.2.38meanMJD	158
12.61.2.39meanMJDval	158
12.61.2.40meanVal	158
12.61.2.41nf	158
12.61.2.42nSessions	158
12.61.2.43outDM	158
12.61.2.44outFileName	158
12.61.2.45outInterpCount	158
12.61.2.46outSmoothCount	158
12.61.2.47outX	158
12.61.2.48outY	158
12.61.2.49parFile	158
12.61.2.50plugVersionCheck	158
12.61.2.51rawOut	158
12.61.2.52sessionSeparation	158
12.61.2.53smoothWidth	158
12.61.2.54splineOut	159
12.61.2.55start_sessions	159
12.61.2.56timFile	159
12.61.2.57title	159
12.61.2.58valID	159
12.61.2.59xlab	159
12.61.2.60ylab	159
12.62plugin/checkWhite_plug.C File Reference	159
12.62.1 Macro Definition Documentation	160
12.62.1.1 MAX_POLY	160
12.62.2 Function Documentation	160
12.62.2.1 average	160
12.62.2.2 calcStat	160
12.62.2.3 corr2pt	160

12.62.2.4 graphicalInterface	160
12.62.2.5 help	160
12.62.2.6 lombScargle	160
12.62.2.7 plotHistogram	160
12.62.2.8 plotResiduals	160
12.62.2.9 shuffle	160
12.62.2.10 shufflePoints	160
12.62.3 Variable Documentation	160
12.62.3.1 plugVersionCheck	160
12.63 plugin/cholSpectra_plug.C File Reference	160
12.63.1 Function Documentation	161
12.63.1.1 calculateSpectrum	161
12.63.1.2 graphicalInterface	161
12.63.1.3 help	161
12.63.2 Variable Documentation	161
12.63.2.1 OMEGA0	161
12.63.2.2 plugVersionCheck	161
12.63.2.3 toffset	161
12.64 plugin/clock_plug.C File Reference	162
12.64.1 Function Documentation	162
12.64.1.1 graphicalInterface	162
12.64.1.2 help	162
12.64.1.3 mjd2year	162
12.64.1.4 slaCalyd	162
12.64.1.5 slaClyd	162
12.64.2 Variable Documentation	162
12.64.2.1 cholmode	162
12.64.2.2 covarFuncFile	163
12.64.2.3 plugVersionCheck	163
12.65 plugin/compareBackends_plug.C File Reference	163
12.65.1 Function Documentation	163
12.65.1.1 graphicalInterface	163
12.65.1.2 help	163
12.65.1.3 runPlugin	163
12.65.2 Variable Documentation	163
12.65.2.1 plugVersionCheck	164
12.66 plugin/compareDsets_plug.C File Reference	164
12.66.1 Function Documentation	164
12.66.1.1 checkSecondComparison	164
12.66.1.2 compareDatasets	165

12.66.1.3 findOverlap	165
12.66.1.4 graphicalInterface	165
12.66.1.5 help	165
12.66.1.6 idPoint	165
12.66.1.7 idPoint2	165
12.66.2 Variable Documentation	165
12.66.2.1 plugVersionCheck	165
12.67plugin/delays_plug.C File Reference	165
12.67.1 Macro Definition Documentation	166
12.67.1.1 MAX_HIGHLIGHT	166
12.67.2 Function Documentation	166
12.67.2.1 callFit	166
12.67.2.2 createNewArrivalTimes	166
12.67.2.3 deletePoint	166
12.67.2.4 doPlot	166
12.67.2.5 findMax	166
12.67.2.6 findMean	166
12.67.2.7 findMin	166
12.67.2.8 fortranMod	166
12.67.2.9 graphicalInterface	166
12.67.2.10help	166
12.67.2.11idPoint	166
12.67.3 Variable Documentation	166
12.67.3.1 plugVersionCheck	166
12.68plugin/designmatrix_plug.C File Reference	166
12.68.1 Function Documentation	167
12.68.1.1 graphicalInterface	167
12.68.1.2 help	167
12.68.1.3 ProcessTempo2Objects	167
12.68.1.4 tempo2_GetNumberOfParameters	167
12.68.1.5 WriteDesignMatrix	167
12.69plugin/detectGWB_plug.C File Reference	168
12.69.1 Function Documentation	169
12.69.1.1 calcSpectra_plugin	169
12.69.1.2 cosineFunc	169
12.69.1.3 formCholeskyMatrixPlugin	169
12.69.1.4 getSpectrum	169
12.69.1.5 graphicalInterface	169
12.69.1.6 hdfunc	169
12.69.1.7 hdfunc_cosineSub	169

12.69.1.8 hdfunc_meanSub	169
12.69.1.9 hdfunc_offs	169
12.69.1.10 hdfunc_removeCosine	169
12.69.1.11 help	169
12.69.1.12 psrangle	169
12.69.2 Variable Documentation	169
12.69.2.1 GLOBAL_COSVAL	169
12.69.2.2 GLOBAL_MEANSUB	169
12.69.2.3 OMEGA0	169
12.69.2.4 write_debug_files	169
12.70 plugin/detectGWBnew_plug.C File Reference	169
12.70.1 Function Documentation	170
12.70.1.1 calcSpectra_plugin	170
12.70.1.2 fitMeanSineFunc	170
12.70.1.3 fitPolyFunc	170
12.70.1.4 formCholeskyMatrixPlugin	170
12.70.1.5 getSpectrum	171
12.70.1.6 graphicalInterface	171
12.70.1.7 hdfunc	171
12.70.1.8 help	171
12.70.1.9 offsetToCM	171
12.70.1.10 psrangle	171
12.70.2 Variable Documentation	171
12.70.2.1 notim	171
12.70.2.2 OMEGA0	171
12.70.2.3 write_debug_files	171
12.70.2.4 write_python_files	171
12.71 plugin/dm_plug.C File Reference	171
12.71.1 Macro Definition Documentation	172
12.71.1.1 MAX_TIMES	172
12.71.2 Function Documentation	172
12.71.2.1 doPlot	172
12.71.2.2 graphicalInterface	172
12.71.2.3 help	172
12.71.2.4 mjd2year	172
12.71.2.5 selectData	172
12.71.2.6 slaCalyd	172
12.71.2.7 slaClyd	172
12.71.3 Variable Documentation	172
12.71.3.1 plugVersionCheck	172

12.72plugin/dmmodel_fitFunc_plug.C File Reference	172
12.72.1 Function Documentation	173
12.72.1.1 getFitLabels	173
12.72.1.2 pluginFitFunc	173
12.72.1.3 updateDMvals	173
12.73plugin/efacEquad_plug.C File Reference	173
12.73.1 Macro Definition Documentation	175
12.73.1.1 EPS	175
12.73.1.2 EPS1	175
12.73.1.3 EPS2	175
12.73.1.4 FMAX	175
12.73.1.5 FPMIN	175
12.73.1.6 FREE_ARG	175
12.73.1.7 ITMAX	175
12.73.1.8 M	175
12.73.1.9 NR_END	175
12.73.1.10NSTACK	175
12.73.1.11SWAP	175
12.73.2 Function Documentation	175
12.73.2.1 calcEfacEquad	175
12.73.2.2 calcEfacEquad	175
12.73.2.3 calcEfacEquad2	175
12.73.2.4 erff	175
12.73.2.5 free_ivector	175
12.73.2.6 gammln	175
12.73.2.7 gammp	175
12.73.2.8 gaussFunc	175
12.73.2.9 gcf	175
12.73.2.10graphicalInterface	175
12.73.2.11gser	175
12.73.2.12help	176
12.73.2.13vector	176
12.73.2.14ksone	176
12.73.2.15kstwo	176
12.73.2.16nerror	176
12.73.2.17probks	176
12.73.2.18sort	176
12.73.3 Variable Documentation	176
12.73.3.1 plugVersionCheck	176
12.74plugin/fake_plug.C File Reference	176

12.74.1 Function Documentation	177
12.74.1.1 callFit	177
12.74.1.2 graphicalInterface	177
12.74.2 Variable Documentation	177
12.74.2.1 plugVersionCheck	177
12.75plugin/fermi_plug.C File Reference	177
12.75.1 Macro Definition Documentation	178
12.75.1.1 SECDAY	178
12.75.2 Function Documentation	178
12.75.2.1 clock_corrections_fermi	178
12.75.2.2 cpgpt	178
12.75.2.3 ephemeris_routines_fermi	178
12.75.2.4 extra_delays_fermi	178
12.75.2.5 formBatsAll_fermi	178
12.75.2.6 graphicalInterface	178
12.75.2.7 HTest	178
12.75.2.8 inner_product	178
12.75.2.9 met2mjd	178
12.75.2.10mjd2met	178
12.75.2.11outer_product	178
12.75.3 Variable Documentation	178
12.75.3.1 plugVersionCheck	178
12.76plugin/findCW_plug.C File Reference	178
12.76.1 Function Documentation	179
12.76.1.1 graphicalInterface	179
12.76.1.2 help	179
12.76.2 Variable Documentation	179
12.76.2.1 plugVersionCheck	179
12.77plugin/findCWs_plug.C File Reference	179
12.77.1 Function Documentation	180
12.77.1.1 graphicalInterface	180
12.77.1.2 help	180
12.78plugin/fixData_plug.C File Reference	180
12.78.1 Function Documentation	181
12.78.1.1 determine1dStructureFunction	181
12.78.1.2 doPlugin1	181
12.78.1.3 doPlugin2	181
12.78.1.4 doPlugin3	181
12.78.1.5 doSummary	181
12.78.1.6 graphicalInterface	181

12.78.1.7 help	181
12.78.1.8 plotHistogram	181
12.78.2 Variable Documentation	181
12.78.2.1 alpha	181
12.78.2.2 dayGap	181
12.78.2.3 gwamp	181
12.78.2.4 nit	181
12.78.2.5 plotout	181
12.78.2.6 plotoutSet	182
12.78.2.7 plugVersionCheck	182
12.78.2.8 script	182
12.79plugin/general2_plug.C File Reference	182
12.79.1 Function Documentation	182
12.79.1.1 fortranMod	182
12.79.1.2 nint_derived	183
12.79.1.3 parseLine	183
12.79.1.4 rnd8	183
12.79.1.5 tempoOutput	183
12.79.2 Variable Documentation	183
12.79.2.1 plugVersionCheck	183
12.80plugin/general_plug.C File Reference	183
12.80.1 Function Documentation	184
12.80.1.1 nint_derived	184
12.80.1.2 parseLine	184
12.80.1.3 rnd8	184
12.80.1.4 tempoOutput	184
12.80.2 Variable Documentation	184
12.80.2.1 plugVersionCheck	184
12.81plugin/glast_plug.C File Reference	184
12.81.1 Macro Definition Documentation	185
12.81.1.1 FREE_ARG	185
12.81.1.2 M	185
12.81.1.3 NR_END	185
12.81.1.4 NRANSI	185
12.81.1.5 NSTACK	185
12.81.1.6 SWAP	185
12.81.2 Function Documentation	185
12.81.2.1 fitwave_function	185
12.81.2.2 free_ivector	185
12.81.2.3 getParameter	185

12.81.2.4 graphicalInterface	185
12.81.2.5 help	185
12.81.2.6 indexx_patrick	185
12.81.2.7 ivector	185
12.81.2.8 nrerror	185
12.81.2.9 sla_CALDJ	185
12.81.2.10sla_CLDJ	185
12.81.2.11slaCalyd	185
12.81.2.12slaClyd	185
12.82plugin/glitch_plug.C File Reference	186
12.82.1 Macro Definition Documentation	188
12.82.1.1 LM_DWARF	188
12.82.1.2 LM_MACHEP	188
12.82.1.3 LM_SQRT_DWARF	188
12.82.1.4 LM_SQRT_GIANT	188
12.82.1.5 LM_USERTOL	188
12.82.1.6 MAX	188
12.82.1.7 MAX_TIMES	188
12.82.1.8 MIN	188
12.82.1.9 SQR	188
12.82.2 Typedef Documentation	188
12.82.2.1 glitchS	188
12.82.3 Function Documentation	188
12.82.3.1 changeFit	188
12.82.3.2 checkMenu	188
12.82.3.3 defineGlitchVal	188
12.82.3.4 doPlot	188
12.82.3.5 drawMenu	188
12.82.3.6 fitFuncs	189
12.82.3.7 graphicalInterface	189
12.82.3.8 help	189
12.82.3.9 interactivePlot	189
12.82.3.10m_enorm	189
12.82.3.11lm_lmdif	189
12.82.3.12m_lmpar	189
12.82.3.13m_printout_std	189
12.82.3.14m_qrfac	189
12.82.3.15m_qrsolv	189
12.82.3.16mcurve_evaluate	190
12.82.3.17mcurve_fit	190

12.82.3.18	mmmin	190
12.82.3.19	nonlinearFunc	190
12.82.3.20	plot1	190
12.82.3.21	plot2	190
12.82.3.22	plot3	190
12.82.3.23	plot4	190
12.82.3.24	plot5	190
12.82.3.25	plot6	190
12.82.3.26	plot7	190
12.82.3.27	plot8	190
12.82.3.28	plot9	190
12.82.4	Variable Documentation	190
12.82.4.1	global_fit0	190
12.82.4.2	global_fit1	190
12.82.4.3	global_footer	190
12.82.4.4	global_glitch	190
12.82.4.5	global_header	190
12.82.4.6	global_nglt	190
12.82.4.7	global_valf0	190
12.82.4.8	global_valf1	191
12.82.4.9	lm_control_double	191
12.82.4.10	lm_control_float	191
12.82.4.11	lm_infmsg	191
12.82.4.12	lm_shortmsg	191
12.82.4.13	plugVersionCheck	191
12.83	plugin/global_fitFunc_plug.C File Reference	191
12.83.1	Function Documentation	192
12.83.1.1	globalFITfuncs	192
12.83.1.2	pluginFitFunc	192
12.83.2	Variable Documentation	192
12.83.2.1	gnpsr	192
12.83.2.2	plugVersionCheck	192
12.84	plugin/globalDCM_fitFunc_plug.C File Reference	192
12.84.1	Function Documentation	193
12.84.1.1	formCholeskyMatrix2	193
12.84.1.2	globalFITfuncs	193
12.84.1.3	multMatrix2	193
12.84.1.4	multMatrixVec2	193
12.84.1.5	pluginFitFunc	193
12.84.1.6	readUinv	193

12.84.1.7 TKbacksubstitution_svd2	193
12.84.1.8 TKbidiagonal2	193
12.84.1.9 TKpythag2	193
12.84.1.10TKsingularValueDecomposition_Isq2	193
12.84.2 Variable Documentation	193
12.84.2.1 gnpsr	193
12.84.2.2 plugVersionCheck	193
12.85plugin/grTemplate_plug.C File Reference	193
12.85.1 Function Documentation	194
12.85.1.1 graphicalInterface	194
12.85.1.2 help	194
12.85.2 Variable Documentation	194
12.85.2.1 plugVersionCheck	194
12.86plugin/GWanisobkgrd_plug.C File Reference	194
12.86.1 Function Documentation	195
12.86.1.1 convertXY_celestial	195
12.86.1.2 doPlot	195
12.86.1.3 draw_grid	195
12.86.1.4 getTspan	196
12.86.1.5 graphicalInterface	196
12.86.1.6 help	196
12.86.1.7 plotPosn	196
12.86.1.8 plotResiduals	196
12.86.1.9 plotSpectrum	196
12.86.2 Variable Documentation	196
12.86.2.1 plugVersionCheck	196
12.87plugin/GWbkgrd_plug.C File Reference	196
12.87.1 Function Documentation	197
12.87.1.1 convertXY_celestial	197
12.87.1.2 doPlot	197
12.87.1.3 draw_grid	197
12.87.1.4 getTspan	197
12.87.1.5 graphicalInterface	197
12.87.1.6 help	197
12.87.1.7 plotPosn	197
12.87.1.8 plotResiduals	197
12.87.1.9 plotSpectrum	197
12.87.2 Variable Documentation	197
12.87.2.1 plugVersionCheck	197
12.88plugin/GWbkgrdfromfile_plug.C File Reference	197

12.88.1 Function Documentation	198
12.88.1.1 convertXY_celestial	198
12.88.1.2 doGenPlot	198
12.88.1.3 doPlot	198
12.88.1.4 draw_grid	198
12.88.1.5 getTspan	199
12.88.1.6 graphicalInterface	199
12.88.1.7 help	199
12.88.1.8 plotGenPosn	199
12.88.1.9 plotGenSpectrum	199
12.88.1.10 plotPosn	199
12.88.1.11 plotResiduals	199
12.88.1.12 plotSpectrum	199
12.88.2 Variable Documentation	199
12.88.2.1 NGWmax	199
12.88.2.2 plugVersionCheck	199
12.89 plugin/GWdetect_plug.C File Reference	199
12.89.1 Function Documentation	200
12.89.1.1 graphicalInterface	200
12.89.1.2 help	200
12.89.1.3 searchGridPos	200
12.90 plugin/GWdipolebkgd_plug.C File Reference	200
12.90.1 Function Documentation	201
12.90.1.1 convertXY_celestial	201
12.90.1.2 doPlot	201
12.90.1.3 draw_grid	201
12.90.1.4 getTspan	201
12.90.1.5 graphicalInterface	201
12.90.1.6 help	201
12.90.1.7 plotPosn	201
12.90.1.8 plotResiduals	201
12.90.1.9 plotSpectrum	201
12.90.2 Variable Documentation	201
12.90.2.1 plugVersionCheck	201
12.91 plugin/GWevolve_plug.C File Reference	201
12.91.1 Macro Definition Documentation	203
12.91.1.1 BIG_G	203
12.91.1.2 ERRCON	203
12.91.1.3 FMAX	203
12.91.1.4 FREE_ARG	203

12.91.1.5 MAX_VAL	203
12.91.1.6 MAXSTP	203
12.91.1.7 NR_END	203
12.91.1.8 PCM	203
12.91.1.9 PGROW	203
12.91.1.10PSHRNK	203
12.91.1.11SAFETY	203
12.91.1.12SIGN	203
12.91.1.13SOLAR_MASS	204
12.91.1.14SPEED_LIGHT	204
12.91.1.15TINY	204
12.91.2 Function Documentation	204
12.91.2.1 calcAmp	204
12.91.2.2 free_vector	204
12.91.2.3 graphicalInterface	204
12.91.2.4 help	204
12.91.2.5 nrerror	204
12.91.2.6 ode	204
12.91.2.7 ode	204
12.91.2.8 psrangle	204
12.91.2.9 RungeKuttaCashKarp	204
12.91.2.10RungeKuttaStep	204
12.91.2.11setup3C66B	204
12.91.2.12setupTest	204
12.91.2.13ThetaEderivs	204
12.91.2.14vector	204
12.91.3 Variable Documentation	204
12.91.3.1 const2	204
12.91.3.2 constA0	204
12.91.3.3 dxsav	204
12.91.3.4 kmax	204
12.91.3.5 kount	204
12.91.3.6 plugVersionCheck	205
12.91.3.7 xp	205
12.91.3.8 yp	205
12.92plugin/GWgeneralanisobkgrd_plug.C File Reference	205
12.92.1 Function Documentation	206
12.92.1.1 convertXY_celestial	206
12.92.1.2 doPlot	206
12.92.1.3 draw_grid	206

12.92.1.4	getTspan	206
12.92.1.5	graphicalInterface	206
12.92.1.6	help	206
12.92.1.7	plotPosn	206
12.92.1.8	plotResiduals	206
12.92.1.9	plotSpectrum	206
12.92.2	Variable Documentation	206
12.92.2.1	plugVersionCheck	206
12.93	plugin/GWgeneralbkgrd_plug.C File Reference	206
12.93.1	Function Documentation	207
12.93.1.1	convertXY_celestial	207
12.93.1.2	doPlot	207
12.93.1.3	draw_grid	207
12.93.1.4	getTspan	207
12.93.1.5	graphicalInterface	207
12.93.1.6	help	207
12.93.1.7	plotPosn	207
12.93.1.8	plotResiduals	207
12.93.1.9	plotSpectrum	207
12.93.2	Variable Documentation	207
12.93.2.1	plugVersionCheck	207
12.94	plugin/gwm_plug.C File Reference	207
12.94.1	Function Documentation	208
12.94.1.1	graphicalInterface	208
12.94.1.2	help	208
12.94.2	Variable Documentation	208
12.94.2.1	plugVersionCheck	208
12.95	plugin/gwmStats_plug.C File Reference	208
12.95.1	Macro Definition Documentation	209
12.95.1.1	MAX_CORR	209
12.95.2	Function Documentation	209
12.95.2.1	calculateAngularFactors	209
12.95.2.2	calculateD	209
12.95.2.3	graphicalInterface	209
12.95.2.4	help	209
12.95.3	Variable Documentation	209
12.95.3.1	plugVersionCheck	209
12.96	plugin/GWsens_plug.C File Reference	209
12.96.1	Function Documentation	210
12.96.1.1	detectSource	210

12.96.1.2 doPlugin	210
12.96.1.3 getSensCurv	210
12.96.1.4 graphicalInterface	210
12.96.1.5 help	211
12.96.2 Variable Documentation	211
12.96.2.1 plugVersionCheck	211
12.97plugin/GWsingle_plug.C File Reference	211
12.97.1 Function Documentation	212
12.97.1.1 convertXY_celestial	212
12.97.1.2 doPlot	212
12.97.1.3 draw_grid	212
12.97.1.4 getTspan	212
12.97.1.5 graphicalInterface	212
12.97.1.6 help	212
12.97.1.7 plotPosn	212
12.97.1.8 plotResiduals	212
12.97.1.9 plotSpectrum	212
12.97.2 Variable Documentation	212
12.97.2.1 plugVersionCheck	212
12.98plugin/GWwhiteLimit_plug.C File Reference	212
12.98.1 Macro Definition Documentation	213
12.98.1.1 MAX_FLAG	213
12.98.1.2 MAX_FREQ	213
12.98.1.3 MAX_ITERATION	213
12.98.1.4 MAX_POLY	213
12.98.1.5 SIGN	213
12.98.2 Function Documentation	213
12.98.2.1 checkReal	213
12.98.2.2 cumulativeHistogram	213
12.98.2.3 cumulativeHistogram2	214
12.98.2.4 getLimits	214
12.98.2.5 getThreshold	214
12.98.2.6 GramSchmidt	214
12.98.2.7 GramSchmidt	214
12.98.2.8 graphicalInterface	214
12.98.2.9 help	214
12.98.2.10setupPulsar	214
12.98.2.11shuffle	214
12.98.2.12sortit	214
12.98.2.13writeCommands	214

12.98.3 Variable Documentation	214
12.98.3.1 plugVersionCheck	214
12.98.3.2 storeVal	214
12.99plugin/icLimit_plug.C File Reference	214
12.99.1 Function Documentation	215
12.99.1.1 calculateGWCholesky	215
12.99.1.2 calculateStatistic	215
12.99.1.3 calculateWeighting	215
12.99.1.4 createGWcovarianceFunction	215
12.99.1.5 formCholeskyMatrixPlugin	215
12.99.1.6 getSpectra	215
12.99.1.7 getStatPS	216
12.99.1.8 getTspan	216
12.99.1.9 graphicalInterface	216
12.99.1.10help	216
12.99.2 Variable Documentation	216
12.99.2.1 plugVersionCheck	216
12.100plugin/interpolate_plug.C File Reference	216
12.100.1 Macro Definition Documentation	217
12.100.1.1FREE_ARG	217
12.100.1.2MAX_SAMPLES	217
12.100.1.3NR_END	217
12.100.1.4NRANSI	217
12.100.1.5TINY	217
12.100.2 Typedef Documentation	217
12.100.2.1sample	217
12.100.3 Function Documentation	217
12.100.3.1choldc	217
12.100.3.2free_vector	217
12.100.3.3getPowerSpectra	217
12.100.3.4graphicalInterface	218
12.100.3.5help	218
12.100.3.6ubksb	218
12.100.3.7udcmp	218
12.100.3.8matrixMult	218
12.100.3.9nrerror	218
12.100.3.10lotModel	218
12.100.3.11lotResiduals	218
12.100.3.12ortSamples	218
12.100.3.13ector	218

12.100.4/variable Documentation	218
12.100.4.1plugVersionCheck	218
12.100.4/plugin/matrix_plug.C File Reference	218
12.101.1/Function Documentation	219
12.101.1.1getLabel	219
12.101.1.2tempoOutput	219
12.101.2/variable Documentation	219
12.101.2.1plugVersionCheck	219
12.101.2/plugin/mjk_plug.C File Reference	219
12.102.1/Macro Definition Documentation	220
12.102.1.1NIT	220
12.102.2/Function Documentation	220
12.102.2.1_itt	220
12.102.2.2graphicalInterface	220
12.102.2.3help	220
12.102.2.4iterativeFit	220
12.102.2.5saveparams	220
12.102.2/plugin/photons_plug.C File Reference	220
12.103.1/Macro Definition Documentation	221
12.103.1.1IAU_K	221
12.103.1.2AU_KINV	221
12.103.1.3AU_TEPH0	221
12.103.2/Function Documentation	221
12.103.2.1check_barycentered	221
12.103.2.2find_event_hdu	221
12.103.2.3get_mjdref	221
12.103.2.4graphicalInterface	221
12.103.2.5cb2tdb	221
12.103.2.6db2tcb	221
12.103.2/plugin/planet_plug.C File Reference	221
12.104.1/Function Documentation	223
12.104.1.1calculateCholeskyCovarFunc	223
12.104.1.2calculateDailyCovariance	223
12.104.1.3calculateSpectra	223
12.104.1.4doPlugin	223
12.104.1.5fileOutput2	223
12.104.1.6fileOutput3	223
12.104.1.7findSmoothCurve	223
12.104.1.8fitExponential	223
12.104.1.9fitSineFunc	223

12.104.1.10	getHighFreqCovar	223
12.104.1.11	getHighFreqRes	223
12.104.1.12	graphicalInterface	223
12.104.1.13	help	223
12.104.1.14	modelFcn	224
12.104.1.15	obtainTimingResiduals	224
12.104.1.16	outputCovarianceFunction	224
12.104.1.17	outputMatrix	224
12.104.1.18	plot1	224
12.104.1.19	plot2	224
12.104.1.20	plot3	224
12.104.1.21	plot3a	224
12.104.1.22	plot4	224
12.104.1.23	plot5	224
12.104.1.24	plot6	224
12.104.1.25	removeMean	224
12.104.1.26	fitSpectraRMS	224
12.104.2	variable Documentation	224
12.104.2.1	G_OMEGA	224
12.104.2.2	pgdevice	224
12.104.2.3	plugVersionCheck	224
12.104.2.4	skipprocess	224
12.104.2.5	skipstep2	224
12.104.2.6	writeFiles	224
12.105	plugin/plk_plug.C File Reference	225
12.105	Function Documentation	226
12.105.1.1	averagePts	226
12.105.1.2	binResiduals	226
12.105.1.3	callFit	226
12.105.1.4	changeFitParameters	226
12.105.1.5	changeParameters	226
12.105.1.6	checkMenu	227
12.105.1.7	checkMenu3	227
12.105.1.8	deletePoint	227
12.105.1.9	displayStatistics	227
12.105.1.10	doPlot	227
12.105.1.11	drawAxisSel	227
12.105.1.12	drawMenu	227
12.105.1.13	drawMenu3	227
12.105.1.14	drawMenu3_2	227

12.105.1.15	DrawOption	227
12.105.1.16	FindMaxY	227
12.105.1.17	FindMean	227
12.105.1.18	FindMeanD	227
12.105.1.19	FindMinY	227
12.105.1.20	TranMod	227
12.105.1.21	GraphicalInterface	227
12.105.1.22	Help	227
12.105.1.23	Point	227
12.105.1.24	Inst2	227
12.105.1.25	NewTim	227
12.105.1.26	VerPlotN	227
12.105.1.27	VerPlotShapiro	227
12.105.1.28	FitWAVES_spec	228
12.105.1.29	Fit	228
12.105.1.30	SetLabel	228
12.105.1.31	SetPlot	228
12.105.1.32	CalcYd	228
12.105.1.33	CalcYd	228
12.105.1.34	Sort	228
12.105.1.35	SwapFit	228
12.105.1.36	NewModels	228
12.105.2	Variable Documentation	228
12.105.2.1	cholmode	228
12.105.2.2	covarFuncFile	228
12.105.2.3	cdmFile	228
12.105.2.4	FITWAVES_harmonicStep	228
12.105.2.5	FITWAVES_n	228
12.105.2.6	FITWAVES_omega	228
12.105.2.7	FITWAVES_par	228
12.105.2.8	flagStore	228
12.105.2.9	plugVersionCheck	228
12.106	plugin/plotMany_plug.C File Reference	228
12.106.1	Function Documentation	229
12.106.1.1	calcYr	229
12.106.1.2	callFit	229
12.106.1.3	doPlot	229
12.106.1.4	findMax	229
12.106.1.5	findMaxVal	230
12.106.1.6	findMean	230

12.106.1.7findMin	230
12.106.1.8findMinVal	230
12.106.1.9fortranMod	230
12.106.1.10graphicalInterface	230
12.106.1.11help	230
12.106.1.12inst2	230
12.106.1.13aCalyd	230
12.106.1.14aClyd	230
12.106.2Variable Documentation	230
12.106.2.1covarFuncFile2	230
12.107plugin/ppta_splug.C File Reference	230
12.107.Function Documentation	231
12.107.1.selectInterface	231
12.108plugin/publish_plug.C File Reference	231
12.108.Function Documentation	232
12.108.1.1dispParameter	232
12.108.1.2fixDec	232
12.108.1.3fixRA	232
12.108.1.4nint_derived	232
12.108.1.5parseExp	232
12.108.1.6parseMinus	232
12.108.1.7rnd8	232
12.108.1.8tempoOutput	232
12.108.2Variable Documentation	232
12.108.2.1plugVersionCheck	232
12.109plugin/sigmaz_plug.C File Reference	232
12.109.1Macro Definition Documentation	234
12.109.1.1MAX_GWS	234
12.109.2Typedef Documentation	234
12.109.2.1XY	234
12.109.3Function Documentation	234
12.109.3.1calcSigmaz	234
12.109.3.2calcSpline	234
12.109.3.3calculateGWlim	234
12.109.3.4convert_gravWaveBackground_fit	235
12.109.3.5convert_gravWaveBackground_noFit	235
12.109.3.6doplot	235
12.109.3.7it4	235
12.109.3.8itv	235
12.109.3.9getprtj	235

12.109.3.100	getweights	235
12.109.3.101	graphicalInterface	235
12.109.3.102	help	235
12.109.3.103	index8	235
12.109.3.104	mat20	235
12.109.3.105	plotA_g	235
12.109.3.106	plotOmega_g	235
12.109.3.107	readin	235
12.109.3.108	shufflePts	235
12.109.3.109	rmWhiteFunc	235
12.109.3.110	sortTimes	235
12.109.3.111	splineBlend	235
12.109.3.112	splineCurve	235
12.109.3.113	splineKnots	235
12.109.3.114	splinePoint	235
12.109.4	Variable Documentation	235
12.109.4.1	addvar	235
12.109.4.2	data	235
12.109.4.3	ndx	235
12.109.4.4	infile	236
12.109.4.5	nbintype	236
12.109.4.6	ncubic	236
12.109.4.7	ncubics	236
12.109.4.8	ndim	236
12.109.4.9	nformat	236
12.109.4.10	opt	236
12.109.4.11	opt1last	236
12.109.4.12	opt2last	236
12.109.4.13	tau	236
12.109.4.14	units	236
12.109.4.15	usewt	236
12.109.4.16	writeres	236
12.109.4.17	xunits	236
12.109.4.18	ymax	236
12.109.4.19	bugVersionCheck	236
12.109.4.20	util	236
12.109.4.21	bot2	236
12.109.4.22	ecyear	236
12.109.4.23	gmai	236
12.109.4.24	uday	236

12.109.4.25	ensure	236
12.109.4.26	log	236
12.109.4.27	amax	236
12.109.4.28	amin	236
12.109.4.29	usec	236
12.109.4.30	year	236
12.109.4.31	diffmin	236
12.109.4.32	max	237
12.109.4.33	min	237
12.109.4.34	first	237
12.109.4.35	jd	237
12.109.4.36	jd1	237
12.109.4.37	jd2	237
12.109.4.38	jdlast	237
12.109.4.39	last	237
12.109.4.40	mean	237
12.109.4.41	max	237
12.109.4.42	min	237
12.110	plugin/simRedNoise_plug.C File Reference	237
12.110.1	Function Documentation	238
12.110.1.1	doPlugin	238
12.110.1.2	getRedNoiseRealisation	238
12.110.1.3	graphicalInterface	238
12.110.1.4	help	238
12.110.2	Variable Documentation	238
12.110.2.1	plugVersionCheck	238
12.111	plugin/simulDM_plug.C File Reference	238
12.111.1	Macro Definition Documentation	239
12.111.1.1	MAX_DM	239
12.111.2	Function Documentation	239
12.111.2.1	doPlot	239
12.111.2.2	doplugin	239
12.111.2.3	graphicalInterface	239
12.111.2.4	help	239
12.111.3	Variable Documentation	239
12.111.3.1	plugVersionCheck	239
12.112	plugin/spectralModel_plug.C File Reference	239
12.112.1	Function Documentation	241
12.112.1.1	calculateCholeskyCovarFunc	241
12.112.1.2	calculateDailyCovariance	241

12.112.1.3	calculateSpectra	241
12.112.1.4	doPlugin	241
12.112.1.5	fileOutput2	241
12.112.1.6	fileOutput3	241
12.112.1.7	findSmoothCurve	241
12.112.1.8	fitExponential	241
12.112.1.9	fitSineFunc	241
12.112.1.10	getHighFreqCovar	241
12.112.1.11	getHighFreqRes	241
12.112.1.12	graphicalInterface	241
12.112.1.13	help	241
12.112.1.14	obtainTimingResiduals	241
12.112.1.15	outputCovarianceFunction	242
12.112.1.16	outputMatrix	242
12.112.1.17	plot1	242
12.112.1.18	plot2	242
12.112.1.19	plot3	242
12.112.1.20	plot3a	242
12.112.1.21	plot4	242
12.112.1.22	plot5	242
12.112.1.23	plot6	242
12.112.1.24	removeMean	242
12.112.2	Variable Documentation	242
12.112.2.1	G_OMEGA	242
12.112.2.2	pgdevice	242
12.112.2.3	plugVersionCheck	242
12.112.2.4	skipprocess	242
12.112.2.5	skipstep2	242
12.112.2.6	writeFiles	242
12.113	plugin/spectrum_plug.C File Reference	243
12.113.1	Macro Definition Documentation	244
12.113.1.1	MAX_ID	244
12.113.2	Function Documentation	244
12.113.2.1	checkMenu	244
12.113.2.2	doPlugin	244
12.113.2.3	drawMenu	244
12.113.2.4	drawOption	244
12.113.2.5	graphicalInterface	244
12.113.2.6	help	244
12.113.2.7	identify	244

12.113.2.8model	244
12.113.3/variable Documentation	244
12.113.3.1covarFuncFile	244
12.113.3.2dcmFile	244
12.113.3.3plugVersionCheck	244
12.114.plugin/splk_plug.C File Reference	244
12.114.1Function Documentation	245
12.114.1.1callFit	245
12.114.1.2deletePoint	245
12.114.1.3doPlot	245
12.114.1.4findMax	245
12.114.1.5findMean	245
12.114.1.6findMin	245
12.114.1.7ortranMod	245
12.114.1.8graphicalInterface	246
12.114.1.9dPoint	246
12.114.2/variable Documentation	246
12.114.2.1covarFuncFile	246
12.114.2.2dcmFile	246
12.114.2.3plugVersionCheck	246
12.115.plugin/transform_plug.C File Reference	246
12.115.1Function Documentation	246
12.115.1.1graphicalInterface	247
12.115.1.2help	247
12.115.2/variable Documentation	247
12.115.2.1plugVersionCheck	247
12.116.plugin/polyco.C File Reference	247
12.116.1Function Documentation	247
12.116.1.1atimfake	247
12.116.1.2hebpc	247
12.116.1.3pcshft	248
12.116.1.4polyco	248
12.116.1.5zFit	248
12.117.plugin/preProcess.C File Reference	248
12.117.1Function Documentation	248
12.117.1.1logicFlag	248
12.117.1.2preProcess	248
12.117.1.3processFlag	248
12.117.1.4processSimultaneous	249
12.117.1.5readWhiteNoiseModelFile	249

12.117.1.6useSelectFile	249
12.118.1preProcessSimple.C File Reference	249
12.118.1Function Documentation	249
12.118.1.1preProcessSimple	249
12.118.1.2preProcessSimple1	249
12.118.1.3preProcessSimple2	249
12.118.1.4preProcessSimple3	249
12.119.1read_fortran.h File Reference	250
12.119.1Function Documentation	251
12.119.1.1close_file	251
12.119.1.2open_file	251
12.119.1.3read_char	251
12.119.1.4read_character	251
12.119.1.5read_double	251
12.119.1.6read_float	251
12.119.1.7read_int	251
12.119.1.8read_record_int	251
12.119.2Variable Documentation	251
12.119.2.1c_fileptr	251
12.119.2.2swapByte	251
12.120.1read_fortran2.h File Reference	251
12.120.1Function Documentation	252
12.120.1.1close_file2	252
12.120.1.2open_file2	252
12.120.1.3read_character2	252
12.120.1.4read_double2	252
12.120.1.5read_float2	252
12.120.1.6read_int2	252
12.120.1.7read_record_int2	252
12.120.2Variable Documentation	252
12.120.2.1c_fileptr2	252
12.120.2.2swapByte2	252
12.121.1readEphemeris.C File Reference	253
12.121.1Macro Definition Documentation	253
12.121.1.1MAX_SHOTS	253
12.121.2Function Documentation	253
12.121.2.1gasdev	253
12.121.2.2random2	253
12.121.2.3readEphemeris	254
12.122.1readEphemeris_calceph.C File Reference	254

12.122. Function Documentation	254
12.122.1.1convertUnits	254
12.122.1.2readEphemeris_calceph	254
12.123.readJBO_bat.C File Reference	254
12.123. Function Documentation	255
12.123.1.1date2mjd	255
12.123.1.2makechars	255
12.123.1.3readJBO_bat	255
12.123.1.4swap4	255
12.123.1.5swap8	255
12.123.1.6swap8	255
12.124.readParfile.C File Reference	255
12.124. Function Documentation	256
12.124.1.1checkAllSet	256
12.124.1.2checkLine	256
12.124.1.3getValue	256
12.124.1.4readParfile	256
12.124.1.5readParfileGlobal	256
12.124.1.6readSimpleParfile	256
12.124.1.7readValue	256
12.124.1.8removeCR	256
12.124.1.9setupParameterFileDefaults	256
12.125.readTimfile.C File Reference	256
12.125. Function Documentation	257
12.125.1.1readTim	257
12.125.1.2readTimfile	257
12.125.1.3removeCR2	257
12.125.1.4writeTim	257
12.126.ecularMotion.C File Reference	257
12.126. Function Documentation	258
12.126.1.1secularMotion	258
12.127.shapiro_delay.C File Reference	258
12.127. Function Documentation	259
12.127.1.1shapiro_delay	259
12.128.sortToAs.C File Reference	259
12.128. Function Documentation	260
12.128.1.1compareObs	260
12.128.1.2sortToAs	260
12.129.storePrecision.C File Reference	260
12.129. Function Documentation	261

12.129.1.1recordPrecision	261
12.130.1.1w_delay.C File Reference	261
12.130.1.1Macro Definition Documentation	262
12.130.1.1.1MAX_CURRENT	262
12.130.2.1Function Documentation	262
12.130.2.1.1amod	262
12.130.2.2.1calcRotN	262
12.130.2.3.1convertEcliptic	262
12.130.2.4.1elsun2	262
12.130.2.5.1findAngle	262
12.130.2.6.1mcl2	263
12.130.2.7.1mjd2date	263
12.130.2.8.1outputResults	263
12.130.2.9.1readCurrentSheet	263
12.130.2.10.1SolarWindModel	263
12.131.1.1polyco.c File Reference	263
12.131.1.1Function Documentation	264
12.131.1.1.1T1P_grabInt	264
12.131.1.1.2T1P_grabLongDouble	264
12.131.1.1.3T1P_grabString	264
12.131.1.1.4T1Polyco_GetFrequency	264
12.131.1.1.5T1Polyco_GetPhase	264
12.131.1.1.6T1Polyco_Read	264
12.131.1.1.7T1Polyco_Read_NewFormat	264
12.131.1.1.8T1Polyco_Write	264
12.131.1.1.9T1PolycoSet_Destroy	264
12.131.1.1.10T1PolycoSet_GetFrequency	264
12.131.1.1.11T1PolycoSet_GetNearest	264
12.131.1.1.12T1PolycoSet_GetPhase	264
12.131.1.1.13T1PolycoSet_Read	264
12.131.1.1.14T1PolycoSet_Write	264
12.132.1.1T2-PTAmodel.C File Reference	264
12.132.1.1Function Documentation	265
12.132.1.1.1computeU	265
12.132.1.1.2T2_PTAmodel	265
12.132.1.1.3updateT2_PTA	265
12.133.1.1T2accel.C File Reference	265
12.133.1.1Macro Definition Documentation	266
12.133.1.1.1F77_dgels	266
12.133.1.1.2F77_dgemm	266

12.133.1.3F77_dgemv	267
12.133.1.4F77_dpotf2	267
12.133.1.5F77_dtptri	267
12.133.1.6F77_dtrmm	267
12.133.2Function Documentation	267
12.133.2.1accel_lsqr	267
12.133.2.2accel_multMatrix	267
12.133.2.3accel_multMatrixVec	267
12.133.2.4accel_uinv	267
12.133.2.5F77_dgels	267
12.133.2.6F77_dgemm	267
12.133.2.7F77_dgemv	267
12.133.2.8F77_dpotf2	267
12.133.2.9F77_dtptri	267
12.133.2.10F77_dtrmm	267
12.133.3Variable Documentation	267
12.133.3.1useT2accel	267
12.1342accel.h File Reference	268
12.134.1Macro Definition Documentation	268
12.134.1.1ACCEL_LSQ	268
12.134.1.2ACCEL_MULTMATRIX	269
12.134.1.3ACCEL_UINV	269
12.134.2Function Documentation	269
12.134.2.1accel_lsqr	269
12.134.2.2accel_multMatrix	269
12.134.2.3accel_multMatrixVec	269
12.134.2.4accel_uinv	269
12.134.3Variable Documentation	269
12.134.3.1useT2accel	269
12.1352fit.C File Reference	269
12.135.1Macro Definition Documentation	270
12.135.1.1T2_SVD_TOL	270
12.135.2Function Documentation	270
12.135.2.1t2Fit	270
12.135.2.2t2Fit_buildConstraintsMatrix	270
12.135.2.3t2Fit_buildDesignMatrix	270
12.135.2.4t2Fit_fillFitInfo	270
12.135.2.5t2Fit_fillFitInfo_INNER	270
12.135.2.6t2Fit_fillGlobalFitInfo	271
12.135.2.7t2Fit_getFitData	271

12.135.2.2Fit_updateParameters	271
12.136.2fit.h File Reference	271
12.136.1Function Documentation	272
12.136.1.1t2Fit	272
12.136.1.22Fit_buildConstraintsMatrix	272
12.136.1.32Fit_buildDesignMatrix	272
12.136.1.42Fit_fillFitInfo	272
12.136.1.52Fit_fillGlobalFitInfo	272
12.136.1.62Fit_getFitData	272
12.136.1.72Fit_updateParameters	272
12.137.2fit_dmmodel.C File Reference	272
12.137.1Function Documentation	273
12.137.1.1t2FitFunc_dmmodelCM	273
12.137.1.22FitFunc_dmmodelDM	273
12.137.1.32UpdateFunc_dmmodelCM	273
12.137.1.42UpdateFunc_dmmodelDM	273
12.138.2fit_dmmodel.h File Reference	273
12.138.1Function Documentation	274
12.138.1.1t2FitFunc_dmmodelCM	275
12.138.1.22FitFunc_dmmodelDM	275
12.138.1.32UpdateFunc_dmmodelCM	275
12.138.1.42UpdateFunc_dmmodelDM	275
12.139.2fit_fitwaves.C File Reference	275
12.139.1Function Documentation	275
12.139.1.1t2FitFunc_fitwaves	275
12.139.1.22UpdateFunc_fitwaves	275
12.140.2fit_fitwaves.h File Reference	276
12.140.1Function Documentation	276
12.140.1.1t2FitFunc_fitwaves	276
12.140.1.22UpdateFunc_fitwaves	276
12.141.2fit_glitch.C File Reference	276
12.141.1Function Documentation	277
12.141.1.1t2FitFunc_stdGlitch	277
12.141.1.22UpdateFunc_stdGlitch	277
12.142.2fit_glitch.h File Reference	278
12.142.1Function Documentation	278
12.142.1.1t2FitFunc_stdGlitch	278
12.142.1.22UpdateFunc_stdGlitch	278
12.143.2fit_ifunc.C File Reference	278
12.143.1Function Documentation	279

12.143.1.1	func	279
12.143.1.2	sinfunc	279
12.143.1.3	FitFunc_ifunc	279
12.143.1.4	FitFunc_sifunc	279
12.143.1.5	UpdateFunc_ifunc	279
12.144	fit_ifunc.h File Reference	280
12.144	Function Documentation	281
12.144.1.1	func	281
12.144.1.2	sinfunc	281
12.144.1.3	FitFunc_ifunc	281
12.144.1.4	FitFunc_sifunc	281
12.144.1.5	UpdateFunc_ifunc	281
12.145	fit_position.C File Reference	281
12.145	Function Documentation	281
12.145.1.1	FitFunc_stdPosition	281
12.145.1.2	UpdateFunc_stdPosition	281
12.146	fit_position.h File Reference	281
12.146	Function Documentation	282
12.146.1.1	FitFunc_stdPosition	282
12.146.1.2	UpdateFunc_stdPosition	283
12.147	fit_stdFitFuncs.C File Reference	283
12.147	Function Documentation	284
12.147.1.1	FitFunc_binaryModels	284
12.147.1.2	FitFunc_jump	284
12.147.1.3	FitFunc_miscDm	284
12.147.1.4	FitFunc_planet	284
12.147.1.5	FitFunc_stdDm	284
12.147.1.6	FitFunc_stdFreq	284
12.147.1.7	FitFunc_stdGravWav	284
12.147.1.8	FitFunc_telPos	284
12.147.1.9	FitFunc_zero	284
12.147.1.10	UpdateFunc_binaryModels	284
12.147.1.11	UpdateFunc_jump	284
12.147.1.12	UpdateFunc_miscDm	284
12.147.1.13	UpdateFunc_planet	284
12.147.1.14	UpdateFunc_simpleAdd	284
12.147.1.15	UpdateFunc_simpleMinus	284
12.147.1.16	UpdateFunc_stdFreq	284
12.147.1.17	UpdateFunc_stdGravWav	284
12.147.1.18	UpdateFunc_telPos	284

12.147.1.112	UpdateFunc_zero	284
12.148	fit_stdFitFuncs.h File Reference	284
12.148	Function Documentation	286
12.148.1.112	FitFunc_binaryModels	286
12.148.1.112	FitFunc_ifunc	286
12.148.1.112	FitFunc_jump	286
12.148.1.112	FitFunc_miscDm	286
12.148.1.112	FitFunc_planet	286
12.148.1.112	FitFunc_stdDm	286
12.148.1.112	FitFunc_stdFreq	286
12.148.1.112	FitFunc_stdGravWav	286
12.148.1.112	FitFunc_telPos	286
12.148.1.112	FitFunc_zero	286
12.148.1.112	UpdateFunc_binaryModels	286
12.148.1.112	UpdateFunc_ifunc	286
12.148.1.112	UpdateFunc_jump	286
12.148.1.112	UpdateFunc_miscDm	286
12.148.1.112	UpdateFunc_planet	286
12.148.1.112	UpdateFunc_simpleAdd	286
12.148.1.112	UpdateFunc_simpleMinus	286
12.148.1.112	UpdateFunc_stdFreq	286
12.148.1.112	UpdateFunc_stdGravWav	286
12.148.1.112	UpdateFunc_telPos	286
12.148.1.112	UpdateFunc_zero	287
12.149	2model.C File Reference	287
12.149	Function Documentation	288
12.149.1.1	addKeplerianJumps	288
12.149.1.2	calcGR	288
12.149.1.3	computeU	288
12.149.1.4	deriveKeplerian	288
12.149.1.5	derivePostKeplerian	288
12.149.1.6	getKeplerian	288
12.149.1.7	getParameter	288
12.149.1.8	getPostKeplerian	288
12.149.1.9	KopeikinTerms	288
12.149.1.10	2model	288
12.149.1.11	updateParameters	288
12.149.1.12	updateT2	288
12.150	2toolkit.C File Reference	288
12.150	Macro Definition Documentation	290

12.150.1.1RAND_M	290
12.150.1.2RAND_N	290
12.150.2Function Documentation	290
12.150.2.1genrand_int32	290
12.150.2.2genrand_real1	290
12.150.2.3nit_genrand	290
12.150.2.4TKconvertFloat1	290
12.150.2.5TKconvertFloat2	290
12.150.2.6TKfindMax_d	290
12.150.2.7TKfindMax_f	290
12.150.2.8TKfindMedian_d	290
12.150.2.9TKfindMedian_f	290
12.150.2.10TKfindMin_d	290
12.150.2.11TKfindMin_f	290
12.150.2.12TKfindRMS_d	290
12.150.2.13TKfindRMS_f	290
12.150.2.14TKfindRMSweight_d	290
12.150.2.15TKgaussDev	290
12.150.2.16TKmean_d	290
12.150.2.17TKmean_f	290
12.150.2.18TKranDev	290
12.150.2.19TKrange_d	290
12.150.2.20TKrange_f	290
12.150.2.21TKretMax_d	290
12.150.2.22TKretMax_f	290
12.150.2.23TKretMin_d	290
12.150.2.24TKretMin_f	290
12.150.2.25TKretMin_i	291
12.150.2.26TKsetSeed	291
12.150.2.27TKsign_d	291
12.150.2.28TKsort_2f	291
12.150.2.29TKsort_3d	291
12.150.2.30TKsort_d	291
12.150.2.31TKsort_f	291
12.150.2.32TKvariance_d	291
12.150.2.33TKzeromean_d	291
12.1512 toolkit.h File Reference	291
12.151.1Detailed Description	292
12.151.2Function Documentation	292
12.151.2.1genrand_int32	292

12.151.2.2	genrand_real1	292
12.151.2.3	nit_genrand	292
12.151.2.4	KconvertFloat1	292
12.151.2.5	KconvertFloat2	292
12.151.2.6	KfindMax_d	292
12.151.2.7	KfindMax_f	292
12.151.2.8	KfindMedian_d	292
12.151.2.9	KfindMedian_f	292
12.151.2.10	KfindMin_d	292
12.151.2.11	KfindMin_f	292
12.151.2.12	KfindRMS_d	292
12.151.2.13	KfindRMS_f	292
12.151.2.14	KfindRMSweight_d	292
12.151.2.15	KgaussDev	292
12.151.2.16	Kmean_d	292
12.151.2.17	Kmean_f	292
12.151.2.18	KranDev	292
12.151.2.19	Krange_d	293
12.151.2.20	Krange_f	293
12.151.2.21	KretMax_d	293
12.151.2.22	KretMax_f	293
12.151.2.23	KretMin_d	293
12.151.2.24	KretMin_f	293
12.151.2.25	KretMin_i	293
12.151.2.26	KsetSeed	293
12.151.2.27	Ksign_d	293
12.151.2.28	Ksort_2f	293
12.151.2.29	Ksort_3d	293
12.151.2.30	Ksort_d	293
12.151.2.31	Ksort_f	293
12.151.2.32	Kvariance_d	293
12.151.2.33	Kzeromean_d	293
12.152	toolkit_global.C File Reference	293
12.152.1	Variable Documentation	294
12.152.1.1	MAX_OBSN	294
12.153	tabulatedfunction.C File Reference	294
12.153.1	Function Documentation	295
12.153.1.1	TabulatedFunction_getEndX	295
12.153.1.2	TabulatedFunction_getStartX	295
12.153.1.3	TabulatedFunction_getValue	295

12.153.1.4	TabulatedFunction_load	295
12.154	tabulatedfunction.h File Reference	296
12.154.1	Function Documentation	297
12.154.1.1	TabulatedFunction_getEndX	297
12.154.1.2	TabulatedFunction_getStartX	297
12.154.1.3	TabulatedFunction_getValue	297
12.154.1.4	TabulatedFunction_load	297
12.155	ai2tt.C File Reference	297
12.156	ai2ut1.C File Reference	297
12.156.1	Function Documentation	298
12.156.1.1	ai2ut1	298
12.156.1.2	ut1red	298
12.157	Tempo2.C File Reference	298
12.157.1	Function Documentation	299
12.157.1.1	clock_corrections	299
12.157.1.2	ephemeris_routines	299
12.157.1.3	extra_delays	299
12.157.1.4	main	299
12.157.1.5	shwart_annoying_dynamic_library_stuff	299
12.158	Tempo2.h File Reference	299
12.158.1	Detailed Description	306
12.158.2	Macro Definition Documentation	307
12.158.2.1	AU_DIST	307
12.158.2.2	AULTSC	307
12.158.2.3	BIG_G	307
12.158.2.4	DM_CONST	307
12.158.2.5	DM_CONST_SI	307
12.158.2.6	ECLIPTIC_OBLIQUITY_VAL	307
12.158.2.7	FB90_TIMEEPH	307
12.158.2.8	GM	307
12.158.2.9	GM_C3	307
12.158.2.10	GMJ_C3	307
12.158.2.11	GMN_C3	307
12.158.2.12	GMS_C3	307
12.158.2.13	GMU_C3	308
12.158.2.14	GMV_C3	308
12.158.2.15	HAVE_GWSIM_H	308
12.158.2.16	F99_TIMEEPH	308
12.158.2.17	FTEPH_FILE	308
12.158.2.18	EAPSECOND_FILE	308

12.158.2.10	ASYR2RADS	308
12.158.2.20	MAX_BPJ_JUMPS	308
12.158.2.21	MAX_CLK_CORR	308
12.158.2.22	MAX_CLKCORR	308
12.158.2.23	MAX_COEFF	308
12.158.2.24	MAX_COMPANIONS	308
12.158.2.25	MAX_DM_DERIVATIVES	308
12.158.2.26	MAX_DMx	309
12.158.2.27	MAX_FILELEN	309
12.158.2.28	MAX_FIT	309
12.158.2.29	MAX_FLAG_LEN	309
12.158.2.30	MAX_FLAGS	309
12.158.2.31	MAX_FREQ_DERIVATIVES	309
12.158.2.32	MAX_IFUNC	309
12.158.2.33	MAX_JUMPS	309
12.158.2.34	MAX_LEAPSEC	309
12.158.2.35	MAX_MSG	309
12.158.2.36	MAX_OBSN_VAL	309
12.158.2.37	MAX_PARAMS	309
12.158.2.38	MAX_PSR_VAL	310
12.158.2.39	MAX_QUAD	310
12.158.2.40	MAX_SITE	310
12.158.2.41	MAX_STOREPRECISION	310
12.158.2.42	MAX_STRLEN	310
12.158.2.43	MAX_T2EFAC	310
12.158.2.44	MAX_T2EQUAD	310
12.158.2.45	MAX_TEL_CLK_OFFS	310
12.158.2.46	MAX_TEL_DX	310
12.158.2.47	MAX_TEL_DY	310
12.158.2.48	MAX_TEL_DZ	310
12.158.2.49	MAX_TNBN	310
12.158.2.50	MAX_TNDMEv	310
12.158.2.51	MAX_TNECORR	311
12.158.2.52	MAX_TNEF	311
12.158.2.53	MAX_TNEQ	311
12.158.2.54	MAX_TNGN	311
12.158.2.55	MAX_TNSQ	311
12.158.2.56	MAX_TOFFSET	311
12.158.2.57	MAX_WHITE	311
12.158.2.58	E_SW_DEFAULT	311

12.158.2.59BLQ	311
12.158.2.60BSSYS_FILE	311
12.158.2.61CM	311
12.158.2.62ECDAY	311
12.158.2.63ECDAYI	312
12.158.2.64I_UNITS	312
12.158.2.65SOLAR_MASS	312
12.158.2.66SOLAR_RADIUS	312
12.158.2.67PEED_LIGHT	312
12.158.2.68C_IAU2000B	312
12.158.2.69C_TEMPO	312
12.158.2.70DB_UNITS	312
12.158.2.71DBTDT_FILE	312
12.158.2.72EMPO2_h_HASH	312
12.158.2.73EMPO2_h_MAJOR_VER	312
12.158.2.74EMPO2_h_MINOR_VER	312
12.158.2.75EMPO2_h_VER	312
12.158.2.76SUN	312
12.158.2.77T1_FILE	312
12.158.3Typedef Documentation	313
12.158.3.1constraint_label	313
12.158.3.2constraintDerivFunc	313
12.158.3.3FitInfo	313
12.158.3.4observation	313
12.158.3.5param_label	313
12.158.3.6paramDerivFunc	313
12.158.3.7parameter	313
12.158.3.8paramUpdateFunc	313
12.158.3.9pulsar	313
12.158.3.10orePrecision	314
12.158.4Enumeration Type Documentation	314
12.158.4.1constraint	314
12.158.4.2abel	315
12.158.5Function Documentation	318
12.158.5.1allocateMemory	318
12.158.5.2autoConstraints	318
12.158.5.3bootstrap	318
12.158.5.4BTJmodel	318
12.158.5.5BTmodel	318
12.158.5.6BTXmodel	318

12.158.5.7calcRMS	318
12.158.5.8calculate_bclt	318
12.158.5.9compute_tropospheric_delays	318
12.158.5.10copyParam	318
12.158.5.11copyPSR	318
12.158.5.12VSdisplayVersion	318
12.158.5.13DGModel	318
12.158.5.14DHmodel	318
12.158.5.15DKmodel	318
12.158.5.16DDmodel	318
12.158.5.17DSmodel	318
12.158.5.18defineClockCorrectionSequence	318
12.158.5.19destroyMemory	318
12.158.5.20destroyOne	318
12.158.5.21displayMsg	318
12.158.5.22displayParameters	318
12.158.5.23m_delays	319
12.158.5.24ms_turn	319
12.158.5.25oFit	319
12.158.5.26oFitAll	319
12.158.5.27oFitDCM	319
12.158.5.28oFitGlobal	319
12.158.5.29otproduct	319
12.158.5.30LL1Hmodel	319
12.158.5.31LL1model	319
12.158.5.32qu2ecl	319
12.158.5.33ITfuncs	319
12.158.5.34rmBats	319
12.158.5.35rmBatsAll	319
12.158.5.36rmResiduals	319
12.158.5.37rtran_mod	319
12.158.5.38rtran_nint	319
12.158.5.39rtran_nlong	319
12.158.5.40et_EOP	319
12.158.5.41et_obsCoord	319
12.158.5.42et_obsCoord_IAU2000B	319
12.158.5.43et_OneobsCoord	319
12.158.5.44etCholeskyMatrix	319
12.158.5.45etClockCorrections	319
12.158.5.46etCorrection	319

12.158.5.47	getCorrectionTT	319
12.158.5.48	getInputs	320
12.158.5.49	getObservatory	320
12.158.5.50	getParamDeriv	320
12.158.5.51	getParameterValue	320
12.158.5.52	ms_turn	320
12.158.5.53	residual	320
12.158.5.54	initialise	320
12.158.5.55	initialiseOne	320
12.158.5.56	model	320
12.158.5.57	logicFlag	320
12.158.5.58	lookup_observatory_alias	320
12.158.5.59	SSmodel	320
12.158.5.60	polyco	320
12.158.5.61	preProcess	320
12.158.5.62	preProcessSimple	320
12.158.5.63	preProcessSimple1	320
12.158.5.64	preProcessSimple2	320
12.158.5.65	preProcessSimple3	320
12.158.5.66	processFlag	320
12.158.5.67	processSimultaneous	320
12.158.5.68	readEphemeris	320
12.158.5.69	readEphemeris_calceph	320
12.158.5.70	readJBO_bat	320
12.158.5.71	readObsFile	321
12.158.5.72	readOneEphemeris	321
12.158.5.73	readParfile	321
12.158.5.74	readParfileGlobal	321
12.158.5.75	readSimpleParfile	321
12.158.5.76	readTimfile	321
12.158.5.77	recordPrecision	321
12.158.5.78	secularMotion	321
12.158.5.79	setPlugPath	321
12.158.5.80	setStart	321
12.158.5.81	setupParameterFileDefaults	321
12.158.5.82	shapiro_delay	321
12.158.5.83	simplePlot	321
12.158.5.84	solarWindModel	321
12.158.5.85	sortToAs	321
12.158.5.86	PTAmodel	321

12.158.5.872model	321
12.158.5.881tt	321
12.158.5.891ut1	321
12.158.5.90xtOutput	321
12.158.5.91a2utc	321
12.158.5.92ansform_units	321
12.158.5.93tb	321
12.158.5.94rn_deg	321
12.158.5.95rn_dms	322
12.158.5.96rn_hms	322
12.158.5.97updateBatsAll	322
12.158.5.98updateBT	322
12.158.5.99updateBTJ	322
12.158.5.100updateBTX	322
12.158.5.101updateDD	322
12.158.5.102updateDDGR	322
12.158.5.103updateDDH	322
12.158.5.104updateDDK	322
12.158.5.105updateDDS	322
12.158.5.106updateELL1	322
12.158.5.107updateELL1H	322
12.158.5.108updateJV	322
12.158.5.109updateMSS	322
12.158.5.110updateParameters	322
12.158.5.111updateT2	322
12.158.5.112updateT2_PTA	322
12.158.5.113selectFile	322
12.158.5.1142tai	322
12.158.5.115vectorPulsar	322
12.158.5.116ctorscale	322
12.158.5.117ctorsum	322
12.158.5.118iteTim	322
12.158.5.119om_graphics	322
12.158.6variable Documentation	322
12.158.6.1covarFuncFile	322
12.158.6.2dcmFile	322
12.158.6.3displayCVSversion	323
12.158.6.4ECLIPTIC_OBLIQUITY	323
12.158.6.5forceGlobalFit	323
12.158.6.6MAX_OBSN	323

12.158.6.7MAX_PSR	323
12.158.6.8NEWFIT	323
12.158.6.9TEMPO2_ENVIRON	323
12.158.6.10TEMPO2_ERROR	323
12.158.6.11tempo2_plug_path	323
12.158.6.12tempo2_plug_path_len	323
12.158.6.13tempo2MachineType	323
12.158.6.14VeryFast	323
12.159tempo2pred.c File Reference	324
12.159.1Function Documentation	325
12.159.1.1T2Predictor_Copy	325
12.159.1.2T2Predictor_Destroy	325
12.159.1.3T2Predictor_FRead	325
12.159.1.4T2Predictor_FWrite	325
12.159.1.5T2Predictor_GetEndFreq	325
12.159.1.6T2Predictor_GetEndMJD	325
12.159.1.7T2Predictor_GetFrequency	325
12.159.1.8T2Predictor_GetPhase	325
12.159.1.9T2Predictor_GetPlan	325
12.159.1.10T2Predictor_GetPlan_Ext	325
12.159.1.11T2Predictor_GetPSRName	325
12.159.1.12T2Predictor_GetSiteName	325
12.159.1.13T2Predictor_GetStartFreq	325
12.159.1.14T2Predictor_GetStartMJD	325
12.159.1.15T2Predictor_Init	325
12.159.1.16T2Predictor_Insert	325
12.159.1.17T2Predictor_Keep	325
12.159.1.18T2Predictor_Kind	325
12.159.1.19T2Predictor_Read	325
12.159.1.20T2Predictor_Write	325
12.159.2Variable Documentation	325
12.159.2.1tempo2_verbose	325
12.160tempo2pred.h File Reference	326
12.160.1Enumeration Type Documentation	327
12.160.1.1T2PredictorKind	327
12.160.2Function Documentation	327
12.160.2.1T2Predictor_Copy	327
12.160.2.2T2Predictor_Destroy	327
12.160.2.3T2Predictor_FRead	327
12.160.2.4T2Predictor_FWrite	327

12.160.2.5T2Predictor_GetEndFreq	327
12.160.2.6T2Predictor_GetEndMJD	327
12.160.2.7T2Predictor_GetFrequency	327
12.160.2.8T2Predictor_GetPhase	327
12.160.2.9T2Predictor_GetPlan	327
12.160.2.10T2Predictor_GetPlan_Ext	328
12.160.2.11T2Predictor_GetPSRName	328
12.160.2.12T2Predictor_GetSiteName	328
12.160.2.13T2Predictor_GetStartFreq	328
12.160.2.14T2Predictor_GetStartMJD	328
12.160.2.15T2Predictor_Init	328
12.160.2.16T2Predictor_Insert	328
12.160.2.17T2Predictor_Keep	328
12.160.2.18T2Predictor_Kind	328
12.160.2.19T2Predictor_Read	328
12.160.2.20T2Predictor_Write	328
12.160.3/variable Documentation	328
12.160.3.1ChebyModelSet_OutOfRange	328
12.161tempo2pred_int.h File Reference	328
12.161.1Function Documentation	330
12.161.1.1Cheby2D_Construct	330
12.161.1.2Cheby2D_Construct_x_Derivative	330
12.161.1.3Cheby2D_Test	330
12.161.1.4ChebyModel_Construct	330
12.161.1.5ChebyModel_Copy	330
12.161.1.6ChebyModel_Destroy	330
12.161.1.7ChebyModel_GetFrequency	330
12.161.1.8ChebyModel_GetPhase	330
12.161.1.9ChebyModel_Init	330
12.161.1.10ChebyModel_Read	330
12.161.1.11ChebyModel_Test	330
12.161.1.12ChebyModel_Write	330
12.161.1.13ChebyModelSet_Construct	330
12.161.1.14ChebyModelSet_Destroy	330
12.161.1.15ChebyModelSet_GetFrequency	330
12.161.1.16ChebyModelSet_GetNearest	330
12.161.1.17ChebyModelSet_GetPhase	330
12.161.1.18ChebyModelSet_Init	330
12.161.1.19ChebyModelSet_Insert	330
12.161.1.20ChebyModelSet_Keep	330

12.161.1.20	chebyModelSet_Read	330
12.161.1.22	chebyModelSet_Test	330
12.161.1.23	chebyModelSet_Write	330
12.161.1.24	Polyco_GetFrequency	330
12.161.1.25	Polyco_GetPhase	331
12.161.1.26	Polyco_Read	331
12.161.1.27	Polyco_Write	331
12.161.1.28	PolycoSet_Destroy	331
12.161.1.29	PolycoSet_GetFrequency	331
12.161.1.30	PolycoSet_GetNearest	331
12.161.1.31	PolycoSet_GetPhase	331
12.161.1.32	PolycoSet_Read	331
12.161.1.33	PolycoSet_Write	331
12.162	Tempo2Util.C File Reference	331
12.162	Function Documentation	332
12.162.1.1	copyParam	332
12.162.1.2	copyPSR	332
12.162.1.3	displayMsg	332
12.162.1.4	dms_turn	332
12.162.1.5	dotproduct	332
12.162.1.6	equ2ecl	332
12.162.1.7	fortran_mod	332
12.162.1.8	fortran_mod	332
12.162.1.9	fortran_nint	332
12.162.1.10	fortran_nlong	332
12.162.1.11	getParameterValue	332
12.162.1.12	hms_turn	332
12.162.1.13	turn_deg	332
12.162.1.14	turn_dms	332
12.162.1.15	turn_hms	332
12.162.1.16	vectorscale	332
12.162.1.17	vectorsum	332
12.163	Tempo2Util.h File Reference	333
12.163	Function Documentation	333
12.163.1.1	dms_turn	333
12.163.1.2	hms_turn	333
12.163.1.3	turn_deg	333
12.164	TextOutput.C File Reference	333
12.164	Function Documentation	334
12.164.1.1	calcRMS	334

12.164.1.2dglep	334
12.164.1.3m2	334
12.164.1.4printGlitch	334
12.164.1.5extOutput	334
12.165Kcholesky.h File Reference	334
12.165.Function Documentation	335
12.165.1.1cholesky_covarFunc2matrix	335
12.165.1.2cholesky_dmModel	335
12.165.1.3cholesky_dmModelCovarParam	335
12.165.1.4cholesky_ecm	335
12.165.1.5cholesky_formUinv	335
12.165.1.6cholesky_powerlawModel	335
12.165.1.7cholesky_powerlawModel_withBeta	335
12.165.1.8cholesky_readFromCovarianceFunction	335
12.166Kfit.C File Reference	335
12.166.Function Documentation	337
12.166.1.1TKconstrainedLeastSquares	337
12.166.1.2TKfindMax_Ld	337
12.166.1.3TKfindPoly_d	337
12.166.1.4TKfit_getPulsarDesignMatrix	337
12.166.1.5TKfitPoly	337
12.166.1.6TKleastSquares	337
12.166.1.7TKleastSquares_global_pulsar	337
12.166.1.8TKleastSquares_single_pulsar	337
12.166.1.9TKleastSquares_svd	337
12.166.1.10TKleastSquares_svd_noErr	337
12.166.1.11TKleastSquares_svd_passN	337
12.166.1.12TKleastSquares_svd_psr	337
12.166.1.13TKleastSquares_svd_psr_dcm	337
12.166.1.14TKremovePoly_d	337
12.166.1.15TKremovePoly_f	337
12.166.1.16TKrobustConstrainedLeastSquares	338
12.166.1.17TKrobustLeastSquares	338
12.167Kfit.h File Reference	338
12.167.Function Documentation	339
12.167.1.1TKconstrainedLeastSquares	339
12.167.1.2TKfindPoly_d	339
12.167.1.3TKfitPoly	339
12.167.1.4TKleastSquares	339
12.167.1.5TKleastSquares_svd	339

12.167.1.6TKleastSquares_svd_noErr	339
12.167.1.7TKremovePoly_d	339
12.167.1.8TKremovePoly_f	339
12.167.1.9TKrobustConstrainedLeastSquares	339
12.167.1.10TKrobustLeastSquares	339
12.168TKlog.C File Reference	339
12.168TK Function Documentation	340
12.168.1.1TKchklog	340
12.168.1.2logerr_check	340
12.168.2TK Variable Documentation	340
12.168.2.1debugFlag	340
12.168.2.2check	340
12.168.2.3timer_clk	340
12.168.2.4TK_errorCount	340
12.168.2.5TK_errorlog	341
12.168.2.6TK_warnCount	341
12.168.2.7TK_warnlog	341
12.168.2.8writeResiduals	341
12.169TKlog.h File Reference	341
12.169TK Macro Definition Documentation	342
12.169.1.1LOG	342
12.169.1.2BOLDCOLOR	342
12.169.1.3DEPRECATED	342
12.169.1.4ENDERR	342
12.169.1.5ENDL	342
12.169.1.6ERRORCOLOR	342
12.169.1.7LOG_OUTFILE	342
12.169.1.8logdbg	342
12.169.1.9logerr	342
12.169.1.10logmsg	342
12.169.1.11logtchk	342
12.169.1.12logwarn	343
12.169.1.13RESETCOLOR	343
12.169.1.14TK_MAX_ERROR_LEN	343
12.169.1.15TK_MAX_ERRORS	343
12.169.1.16TK_STORE_ERROR	343
12.169.1.17TK_STORE_WARNING	343
12.169.1.18WARNCOLOR	343
12.169.1.19WHEREARG	343
12.169.1.20WHEREERR	343

12.169.1.201	WHERESTR	343
12.169.1.202	WHERECHK	343
12.169.1.203	HEREWARN	343
12.169.2	Function Documentation	343
12.169.2.1	_TKchklog	343
12.169.2.2	logerr_check	343
12.169.3	Variable Documentation	343
12.169.3.1	debugFlag	343
12.169.3.2	check	343
12.169.3.3	timer_clk	343
12.169.3.4	TK_errorCount	343
12.169.3.5	TK_errorlog	343
12.169.3.6	TK_warnCount	343
12.169.3.7	TK_warnlog	343
12.169.3.8	writeResiduals	343
12.170	longdouble.C File Reference	343
12.170.1	Macro Definition Documentation	344
12.170.1.1	BUFSIZE	344
12.170.2	Function Documentation	344
12.170.2.1	ld_fprintf	344
12.170.2.2	ld_printf	344
12.170.2.3	ld_sprintf	344
12.170.2.4	ld_vsprintf	344
12.170.2.5	parse_longdouble	344
12.170.2.6	print_longdouble	344
12.171	longdouble.float128.h File Reference	345
12.171.1	Macro Definition Documentation	345
12.171.1.1	cosl	345
12.171.1.2	fabsl	345
12.171.1.3	floorl	345
12.171.1.4	FMT_LD	346
12.171.1.5	LD_PI	346
12.171.1.6	longdouble	346
12.171.1.7	LONGDOUBLE_IS_FLOAT128	346
12.171.1.8	LONGDOUBLE_ONE	346
12.171.1.9	sinl	346
12.171.1.10	USE_BUILTIN_LONGDOUBLE	346
12.171.2	Typedef Documentation	346
12.171.2.1	longdouble	346
12.171.3	Function Documentation	346

12.171.3.1ld_fprintf	346
12.171.3.2d_printf	346
12.171.3.3d_sprintf	346
12.171.3.4parse_longdouble	346
12.172Klongdouble.h File Reference	346
12.172.1Macro Definition Documentation	347
12.172.1.1cosl	347
12.172.1.2absl	347
12.172.1.3floorl	347
12.172.1.4FMT_LD	347
12.172.1.5D_PI	347
12.172.1.6ongdouble	347
12.172.1.7LONGDOUBLE_IS_FLOAT128	347
12.172.1.8LONGDOUBLE_ONE	347
12.172.1.9sinl	347
12.172.1.10USE_BUILTIN_LONGDOUBLE	347
12.172.2Typedef Documentation	347
12.172.2.1longdouble	347
12.172.3Function Documentation	347
12.172.3.1ld_fprintf	347
12.172.3.2d_printf	348
12.172.3.3d_sprintf	348
12.172.3.4parse_longdouble	348
12.173Klongdouble.ld.h File Reference	348
12.173.1Macro Definition Documentation	348
12.173.1.1ld_fprintf	348
12.173.1.2D_PI	348
12.173.1.3d_printf	348
12.173.1.4d_sprintf	349
12.173.1.5ongdouble	349
12.173.1.6LONGDOUBLE_IS_IEEE754	349
12.173.1.7LONGDOUBLE_ONE	349
12.173.1.8USE_BUILTIN_LONGDOUBLE	349
12.173.2Typedef Documentation	349
12.173.2.1longdouble	349
12.173.3Function Documentation	349
12.173.3.1parse_longdouble	349
12.174Kmatrix.C File Reference	349
12.174.1Function Documentation	350
12.174.1.1free_2df	350

12.174.1.2	free_2dLL	350
12.174.1.3	free_blas	350
12.174.1.4	free_uinv	350
12.174.1.5	get_blas_cols	350
12.174.1.6	get_blas_rows	350
12.174.1.7	malloc_2df	350
12.174.1.8	malloc_2dLL	350
12.174.1.9	malloc_blas	350
12.174.1.10	malloc_uinv	350
12.174.1.11	KmultMatrix	350
12.174.1.12	KmultMatrix_sq	350
12.174.1.13	KmultMatrixVec	350
12.174.1.14	KmultMatrixVec_sq	350
12.175	Kmatrix.h File Reference	350
12.175	Function Documentation	351
12.175.1.1	free_2df	351
12.175.1.2	free_blas	351
12.175.1.3	free_uinv	351
12.175.1.4	get_blas_cols	351
12.175.1.5	get_blas_rows	351
12.175.1.6	malloc_2df	351
12.175.1.7	malloc_blas	351
12.175.1.8	malloc_uinv	351
12.175.1.9	KmultMatrix	351
12.175.1.10	KmultMatrix_sq	351
12.175.1.11	KmultMatrixVec	351
12.175.1.12	KmultMatrixVec_sq	351
12.176	Kspectrum.C File Reference	351
12.176	Macro Definition Documentation	354
12.176.1.1	ABS	354
12.176.1.2	MAX	354
12.176.1.3	MIN	354
12.176	Function Documentation	354
12.176.2.1	calcSpectra	354
12.176.2.2	calcSpectra_ri	354
12.176.2.3	calcSpectra_ri_T	354
12.176.2.4	calcSpectraErr	354
12.176.2.5	fit4	354
12.176.2.6	fitCosSineFunc	354
12.176.2.7	fitMeanSineFunc	354

12.176.2.8fitMeanSineFunc_IFUNC	354
12.176.2.9getprtj	354
12.176.2.10getweights	354
12.176.2.11indexx8	354
12.176.2.12at20	354
12.176.2.13adin	354
12.176.2.14neFunc	354
12.176.2.15K_dft	354
12.176.2.16K_fft	354
12.176.2.17K_fitSinusoids	354
12.176.2.18K_weightLS	355
12.176.2.19KaveragePts	355
12.176.2.20Kboxcar	355
12.176.2.21KcalcSigmaz	355
12.176.2.22Kcmonot	355
12.176.2.23KfirstDifference	355
12.176.2.24Kkhanh	355
12.176.2.25KinterpolateSplineSmoothFixedXPts	355
12.176.2.26Klomb_d	355
12.176.2.27Ksortit	355
12.176.2.28Kspectrum	355
12.176.2.29Kspline_interpolate	355
12.176.3Variable Documentation	355
12.176.3.1addvar	355
12.176.3.2data	355
12.176.3.3GLOBAL_OMEGA	355
12.176.3.4globalOmega	355
12.176.3.5ndx	355
12.176.3.6infile	355
12.176.3.7nbintype	355
12.176.3.8ncubic	355
12.176.3.9ncubics	355
12.176.3.10dim	356
12.176.3.11format	356
12.176.3.12pt	356
12.176.3.13pt1last	356
12.176.3.14pt2last	356
12.176.3.15tau	356
12.176.3.16units	356
12.176.3.17usewt	356

12.176.3.18 writers	356
12.176.3.19 units	356
12.176.3.20 permax	356
12.176.3.21 rti	356
12.176.3.22 tot2	356
12.176.3.23 cyear	356
12.176.3.24 gmai	356
12.176.3.25 auday	356
12.176.3.26 uensure	356
12.176.3.27 aulog	356
12.176.3.28 umax	356
12.176.3.29 umin	356
12.176.3.30 usec	356
12.176.3.31 uyear	356
12.176.3.32 iffmin	356
12.176.3.33 max	356
12.176.3.34 min	356
12.176.3.35 first	356
12.176.3.36 jd	356
12.176.3.37 jd1	356
12.176.3.38 jd2	357
12.176.3.39 jdlast	357
12.176.3.40 last	357
12.176.3.41 tmean	357
12.176.3.42 erbose_calc_spectra	357
12.176.3.43 max	357
12.176.3.44 min	357
12.177 Kspectrum.h File Reference	357
12.177.1 Macro Definition Documentation	359
12.177.1.1 ABS	359
12.177.1.2 MAX	359
12.177.1.3 MIN	359
12.177.2 Typedef Documentation	359
12.177.2.1 complexVal	359
12.177.3 Function Documentation	359
12.177.3.1 calcSpectra	359
12.177.3.2 calcSpectra_ri	359
12.177.3.3 calcSpectra_ri_T	359
12.177.3.4 calcSpectraErr	359
12.177.3.5 fit4	359

12.177.3.6fitCosSineFunc	359
12.177.3.7fitCosSineFunc	359
12.177.3.8fitMeanSineFunc	359
12.177.3.9fitMeanSineFunc_IFUNC	359
12.177.3.10getprtj	359
12.177.3.11getweights	359
12.177.3.12index8	359
12.177.3.13mat20	359
12.177.3.14readin	359
12.177.3.15sineFunc	359
12.177.3.16K_dft	359
12.177.3.17K_fft	360
12.177.3.18K_fitSine	360
12.177.3.19K_fitSinusoids	360
12.177.3.20K_weightLS	360
12.177.3.21KaveragePts	360
12.177.3.22Kboxcar	360
12.177.3.23KcalcSigmaz	360
12.177.3.24Kcmonot	360
12.177.3.25KfirstDifference	360
12.177.3.26Kkhann	360
12.177.3.27KinterpolateSplineSmoothFixedXPts	360
12.177.3.28Klomb_d	360
12.177.3.29Ksortit	360
12.177.3.30Kspectrum	360
12.177.3.31Kspline_interpolate	360
12.177.4Variable Documentation	360
12.177.4.1GLOBAL_OMEGA	360
12.177.4.2verbose_calc_spectra	360
12.178Ksvd.C File Reference	360
12.178Function Documentation	361
12.178.1.1TKbacksubstitution_svd	361
12.178.1.2TKbidiagonal	361
12.178.1.3TKpythag	361
12.178.1.4TKsingularValueDecomposition_Isq	361
12.179Ksvd.h File Reference	362
12.179Function Documentation	362
12.179.1.1TKbacksubstitution_svd	362
12.179.1.2TKbidiagonal	362
12.179.1.3TKpythag	362

12.179.1.4TKsingularValueDecomposition_Isq	362
12.180a2utc.C File Reference	362
12.180.1Macro Definition Documentation	363
12.180.1.1USE_NEW_CLK_CORR	363
12.180.2Function Documentation	363
12.180.2.1convertTOA	363
12.180.2.2linearInterpolate	363
12.180.2.3toa2utc	363
12.180.2.4toa2utc_nist	363
12.181tropo.C File Reference	363
12.181.1Function Documentation	364
12.181.1.1compute_tropospheric_delays	364
12.181.1.2getMeteorologicalValue	364
12.181.1.3getSurfaceAtmosphericPressure	364
12.181.1.4getZenithWetDelay	364
12.181.1.5initialize_meteorology_table	364
12.181.1.6initialize_meteorology_tables	365
12.181.1.7MeteorologyFunction_getEndMJD	365
12.181.1.8MeteorologyFunction_getStartMJD	365
12.181.1.9MeteorologyFunction_getValue	365
12.181.1.10MeteorologyFunction_load	365
12.181.1.11MF_hydrostatic	365
12.181.1.12MF_wet	365
12.181.2Variable Documentation	365
12.181.2.1surfaceAtmosphericPressureTables	365
12.181.2.2zenithWetDelayTables	365
12.1822tdb.C File Reference	365
12.182.1Function Documentation	366
12.182.1.1FB_deltaT	366
12.182.1.2F_deltaT	366
12.182.1.3nit_ifte	366
12.182.1.4t2tb	366
12.183nits.C File Reference	366
12.183.1Function Documentation	367
12.183.1.1scale_param	367
12.183.1.2transform_units	367
12.183.1.3form_mjd	367
12.184tc2tai.C File Reference	367
12.185ectorPulsar.C File Reference	367
12.185.1Function Documentation	368

12.185.1.1vectorPulsar	368
Index	369

Chapter 1

Main Page

- [User Guide](#)
- [Developer Guide](#)
- [Directory structure](#)

Chapter 2

Developer Guide

A brief description of the function

2.1 Tempo2 Developer Guide

2.1.1 About this guide

This guide has been developed to encourage development of tempo2, and to improve the consistency between developers. The majority of this guide has been written by [MJK](#), although all are welcome to contribute.

2.1.2 General code guidelines

Tempo2 is, for historical reasons, mostly written in C but compiled using a C++ compiler. However, be aware that a few parts of tempo2 use C++ classes or other C++ extensions. There is no particular C or C++ version in use, but for now assume that we are using C++98 with GNU extensions (i.e. `-std=gnu++98`)

Todo determine if we should migrate to C++ 11. It has lots of good features, but we need to check that all compilers support it.

Core tempo2 code

As a general rule, we try to minimise the libraries needed to build the core of tempo2 (not plugins). This means you can't link against `libfftw`, `libpgplot`, etc. from the core code. Some linear algebra features from BLAS/LAPACK are made available to the code code via the T2toolkit, and fallback routines have been generated to ensure that the code still works without BLAS/LAPACK. These routines are being expanded all the time.

plugins

For plugins, the rules are much less strict. Currently we compile plugins with links to `cfitsio`, `fftw` and `pgplot` as part of the main plugin distribution.

libt2toolkit

[MJK](#) is attempting to introduce a little more rigour in the coding standards for the code that makes up libt2toolkit, but in general this is treated exactly the same as code tempo2.

2.1.3 Development workflow

Recommended workflow

The recommended workflow is as follows.

Step 1: create a new branch:

```
git checkout -b myfeature
```

Step 2: Make and commit your changes to that branch

```
git commit -a
```

Step 3: Build, test, run your code.

```
make
make check
```

Step 4: If the new features seem good, promote them to the "master" branch.

```
# if the first time
git push --set-upstream origin docs
# otherwise
git push origin
```

and go to <https://bitbucket.org/mkeith/tempo2/pull-requests/new> to make a new pull request. The code will be reviewed by the core developers to check that the changes do not break any important features. If the modification is accepted (almost always) then it will be merged.

Alternative workflow

If you can't be bothered with branches, you can simply work directly on the "dev" branch:

```
git checkout dev
```

And commit as you want.

```
git commit -a && git push origin
```

The dev branch will be merged into master, after code review, as and when required. The drawbacks of this method are that you have to deal with conflicts yourself.

2.1.4 Coding style

Tempo2 does not have a strict coding style. However, it is recommended to adopt the following practice, as illustrated by the snippet below:

```
// copyright statement up here.
#ifdef HAVE_CONFIG_H
#include <config.h> // make sure to include config.h
#endif

#include <stdint>    // standard libries are included first
#include <fftw.h>    // then external libraries
#include "TKlog.h"   // then internal libraries

// functions are preferably camelCase with small first letter.
// strings should be declared as const char* (or std::string) as they are immutable.
void myFunction(int anInt, const char *str, double **matrix) {
    // indent is 4 spaces.

    // use stdint types where possible to avoid confusion on 32-bit vs 64-bit machines.
    // use unsigned types whre sutable
```

```
// use const when a variable will not change
const uint64_t myconst = 1024;

// keywords have a space before parenthesis (e.g. if, for, while).
if (anInt < 10) { // always use braces, even if one line!
    // use TKlog for logging debug messages and warnings.

    // debug for statements that are to be printed when debug flag is set
    logdbg("anInt = %d", anInt);

    // warnings when problem might be an issue but can continue
    logwarn("anInt should be less than 10"); // adds a message to the warning stack

    // messages always appear
    logmsg("Print to terminal")

    // errors for when the operation is likely to fail.
    logerr("aborting because anInt was too large (%d)", anInt);

    // prefer to return on error rather than exit
    return;
}

// best to declare variables in for loops, but give them a proper name (not i, j, k) if possible.
for (size_t iVal = 0; iVal < myconst; iVal++) {
    // ...
}
```

Headers should declare the functions and have documentation! Please avoid globals as much as possible, but sometimes they are required. Use any doxygen markup required to document the interface, ESPECIALLY if it is to be called from outside tempo2.

```
#ifndef myHeader_h // use defines to prevent double declaration
#define myHeader_h

void myFunction(int anInt, const char* str, double** matrix);
#endif
```

Note

Core tempo2 code should be copyright George Hobbs and Russell Edwards until we decide to change this.

Chapter 3

Core Developers

Tempo2 development team

Tempo2 was originally written by George Hobbs and Russell Edwards.

Core package maintainers

- George Hobbs [GH]george.hobbs@csiro.au
 - Core tempo2 development.
 - Gravitational wave codes.
 - Binary models.
- Michael Keith [MK]mkeith@pulsarastronomy.net
 - C++ code maintainence.
 - Linear algebra and least-squares algorithms.
 - Build system maintainence.
 - Unit testing.

Active contributors

- Joris Verbiest
- Lindley Lentati
- Ryan Shannon
- Paul Demorest
- Lucas Guillemot
- Stefan Osłowski
- Willem van Straten
- Rutger van Haasteren
- Anne Archibald

Past Contributors

- Russell Edwards
- Aiden Hotan
- Ankur Chaudhary
- Ingrid Stairs

Chapter 4

Directory structure

The tempo2 directory structure:

```
.
+-- autoconf.boot
+-- documentation
+-- mpack_lite
+-- plugin
+-- sofa
+-- t2runtime
+-- tests
    +-- gtest-1.7.0
    +-- test_data
+-- unsupported_plugins
```

autoconf.boot

This directory contains the .m4 files used by autoconf to build the configure script. It is copied to autoconf/ by the bootstrap script.

documentation

Includes this documentation

mpack_lite

Source code for multi-precision lapack/blas. This is a subset of the mplapack package from <http://mplapack.sourceforge.net/>

plugin

Source code for plugins

sofa

Source code for the 3rd party fortran SOFA library.

T2runtime

This directory contains the runtime files for tempo2, i.e. the contents of this directory should be reached at \$TEMPPO2 This includes the clock correction files, observatory parameters and earth ephemerides, etc.

tests

Source code for the unit tests, and the gtest library. Also contains a number of data files in the test_data subdirectory used by the tests.

unsupported_plugins

Source code for other plugins that are for whatever reason not part of the main distribution.

Chapter 5

User Guide

Chapter 6

Todo List

Page [Developer Guide](#)

determine if we should migrate to C++ 11. It has lots of good features, but we need to check that all compilers support it.

Chapter 7

Module Index

7.1 Modules

Here is a list of all modules:

libt2toolkit API	23
libtempo2 External API	24

Chapter 8

Class Index

8.1 Class List

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

Cheby2D	25
ChebyModel	25
ChebyModelInfo	27
ChebyModelSet	27
clock_correction	28
ClockCorrectionFunction	29
complexVal	30
DynamicArray	30
EOPSample	30
FitInfo	
Details of the fit	31
glitchS	32
gwgeneralSrc	33
gwgenSpec	35
gwSrc	35
IFTE_interpolation_info	36
IFTEphemeris	37
interpolation_info	38
jpl_eph_data	39
lm_control_struct	40
lm_status_struct	41
lmcurve_data_struct	41
MeteorologyFunction	42
observation	
A struct containing the details of a single obesrvation	42
observatory	50
ObservatoryAliasList	51
parameter	
Holds the values for a parameter	52
pulsar	
Details for a single pulsar	53
sample	73
storePrecision	73
T1Polyco	74
T1PolycoSet	75
T2Predictor	75
TabulatedFunction	77
TabulatedFunctionSample	77

XY	78
----------	----

Chapter 9

File Index

9.1 File List

Here is a list of all files with brief descriptions:

age.C	79
bootstrap.C	80
BTJmodel.C	80
BTmodel.C	81
BTXmodel.C	82
calculate_bclt.C	83
cheby2d.c	84
cheby2d_int.C	87
cholesky.C	88
cholesky.h	90
choleskyAutomatic.C	91
choleskyRoutines.C	92
choleskyRoutines.h	94
clkcorr.C	97
config.h	99
constraints.C	101
constraints.h	103
DDGRmodel.C	104
DDHmodel.C	105
DDKmodel.C	106
DDmodel.C	107
DDSmodel.C	108
displayParameters.C	109
dm_delays.C	110
doFit.C	111
dynarr.C	114
dynarr.h	116
ELL1Hmodel.C	117
ELL1model.C	117
eop.C	118
formBats.C	119
formResiduals.C	120
get_obsCoord.C	121
getInputs.C	123
getPeriod.C	124
global.C	124
GWsim.C	127
GWsim.h	130

ifteph.C	133
ifteph.h	135
initialise.C	136
jpl_int.h	138
jpleph.c	138
jpleph.h	140
MSSmodel.C	141
observatory.C	142
polyco.C	247
preProcess.C	248
preProcessSimple.C	249
read_fortran.h	250
read_fortran2.h	251
readEphemeris.C	253
readEphemeris_calceph.C	254
readJBO_bat.C	254
readParfile.C	255
readTimfile.C	256
secularMotion.C	257
shapiro_delay.C	258
sortToAs.C	259
storePrecision.C	260
sw_delay.C	261
t1polyco.c	263
T2-PTAmodel.C	264
T2accel.C	265
T2accel.h	268
t2fit.C	269
t2fit.h	271
t2fit_dmmodel.C	272
t2fit_dmmodel.h	273
t2fit_fitwaves.C	275
t2fit_fitwaves.h	276
t2fit_glitch.C	276
t2fit_glitch.h	278
t2fit_ifunc.C	278
t2fit_ifunc.h	280
t2fit_position.C	281
t2fit_position.h	281
t2fit_stdFitFuncs.C	283
t2fit_stdFitFuncs.h	284
T2model.C	287
T2toolkit.C	288
T2toolkit.h	
Set of routines that are commonly used in tempo2 and/or its plugins	291
t2toolkit_global.C	293
tabulatedfunction.C	294
tabulatedfunction.h	296
tai2tt.C	297
tai2ut1.C	297
tempo2.C	298
tempo2.h	
Main interface to libtempo2	299
tempo2pred.c	324
tempo2pred.h	326
tempo2pred_int.h	328
tempo2Util.C	331
tempo2Util.h	333

textOutput.C	333
TKcholesky.h	334
TKfit.C	335
TKfit.h	338
TKlog.C	339
TKlog.h	341
TKlongdouble.C	343
TKlongdouble.float128.h	345
TKlongdouble.h	346
TKlongdouble.ld.h	348
TKmatrix.C	349
TKmatrix.h	350
TKspectrum.C	351
TKspectrum.h	357
TKsvd.C	360
TKsvd.h	362
toa2utc.C	362
tropo.C	363
tt2tdb.C	365
units.C	366
utc2tai.C	367
vectorPulsar.C	367
plugin/add_pulseNumber_plug.C	144
plugin/addRed_plug.C	144
plugin/analyticChol_plug.C	146
plugin/angle_plug.C	147
plugin/applet_plug.C	147
plugin/autoDM_plug.C	149
plugin/autoSpectralFit_plug.C	150
plugin/averageData_plug.C	151
plugin/bary_plug.C	152
plugin/basic_plug.C	153
plugin/calcDMe_plug.C	154
plugin/checkWhite_plug.C	159
plugin/cholSpectra_plug.C	160
plugin/clock_plug.C	162
plugin/compareBackends_plug.C	163
plugin/compareDsets_plug.C	164
plugin/delays_plug.C	165
plugin/designmatrix_plug.C	166
plugin/detectGWB_plug.C	168
plugin/detectGWBnew_plug.C	169
plugin/dm_plug.C	171
plugin/dmmodel_fitFunc_plug.C	172
plugin/efacEquad_plug.C	173
plugin/fake_plug.C	176
plugin/fermi_plug.C	177
plugin/findCW_plug.C	178
plugin/findCWs_plug.C	179
plugin/fixData_plug.C	180
plugin/general2_plug.C	182
plugin/general_plug.C	183
plugin/glast_plug.C	184
plugin/glitch_plug.C	186
plugin/global_fitFunc_plug.C	191
plugin/globalDCM_fitFunc_plug.C	192
plugin/grTemplate_plug.C	193
plugin/GWanisobkgd_plug.C	194

plugin/GWbkgrd_plug.C	196
plugin/GWbkgrdfromfile_plug.C	197
plugin/GWdetect_plug.C	199
plugin/GWdipolebkgrd_plug.C	200
plugin/GWevolve_plug.C	201
plugin/GWgeneralanisobkgrd_plug.C	205
plugin/GWgeneralbkgrd_plug.C	206
plugin/gwm_plug.C	207
plugin/gwmStats_plug.C	208
plugin/GWsens_plug.C	209
plugin/GWsingle_plug.C	211
plugin/GWwhiteLimit_plug.C	212
plugin/icLimit_plug.C	214
plugin/interpolate_plug.C	216
plugin/matrix_plug.C	218
plugin/mjk_plug.C	219
plugin/photons_plug.C	220
plugin/planet_plug.C	221
plugin/plk_plug.C	225
plugin/plotMany_plug.C	228
plugin/ppta_splug.C	230
plugin/publish_plug.C	231
plugin/sigmaz_plug.C	232
plugin/simRedNoise_plug.C	237
plugin/simulDM_plug.C	238
plugin/spectralModel_plug.C	239
plugin/spectrum_plug.C	243
plugin/splk_plug.C	244
plugin/transform_plug.C	246

Chapter 10

Module Documentation

10.1 libt2toolkit API

Files

- file [T2toolkit.h](#)

Set of routines that are commonly used in tempo2 and/or its plugins.

10.1.1 Detailed Description

10.2 libtempo2 External API

Files

- file [tempo2.h](#)
contains the main interface to libtempo2.

10.2.1 Detailed Description

Chapter 11

Class Documentation

11.1 Cheby2D Struct Reference

```
#include <tempo2pred.h>
```

Public Attributes

- int [nx](#)
- int [ny](#)
- long double * [coeff](#)

11.1.1 Member Data Documentation

11.1.1.1 long double* [Cheby2D::coeff](#)

11.1.1.2 int [Cheby2D::nx](#)

11.1.1.3 int [Cheby2D::ny](#)

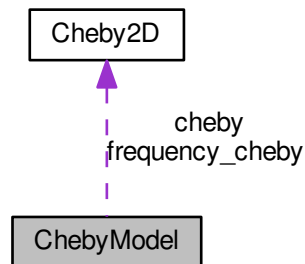
The documentation for this struct was generated from the following file:

- [tempo2pred.h](#)

11.2 ChebyModel Struct Reference

```
#include <tempo2pred.h>
```

Collaboration diagram for ChebyModel:



Public Attributes

- char [psrname](#) [64]
- char [sitename](#) [64]
- long double [mjd_start](#)
- long double [mjd_end](#)
- long double [freq_start](#)
- long double [freq_end](#)
- long double [dispersion_constant](#)
- [Cheby2D](#) [cheby](#)
- [Cheby2D](#) [frequency_cheby](#)

11.2.1 Member Data Documentation

11.2.1.1 **Cheby2D** [ChebyModel::cheby](#)

11.2.1.2 long double [ChebyModel::dispersion_constant](#)

11.2.1.3 long double [ChebyModel::freq_end](#)

11.2.1.4 long double [ChebyModel::freq_start](#)

11.2.1.5 **Cheby2D** [ChebyModel::frequency_cheby](#)

11.2.1.6 long double [ChebyModel::mjd_end](#)

11.2.1.7 long double [ChebyModel::mjd_start](#)

11.2.1.8 char [ChebyModel::psrname](#)[64]

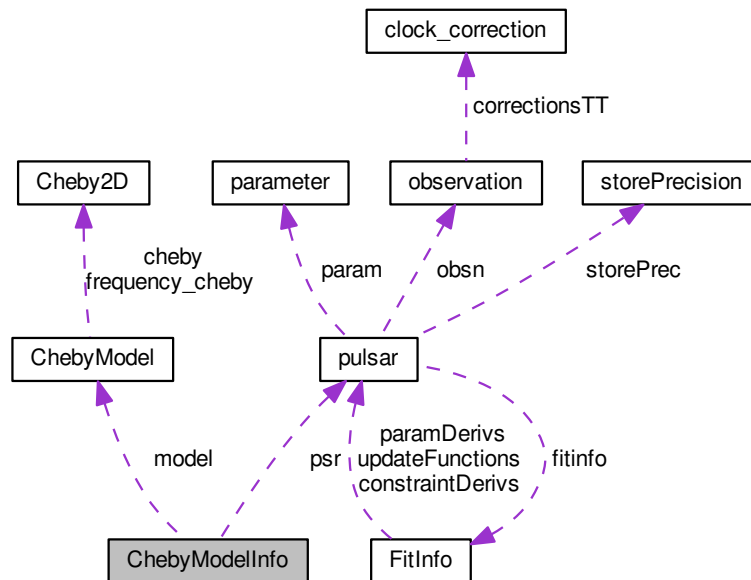
11.2.1.9 char [ChebyModel::sitename](#)[64]

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

- [tempo2pred.h](#)

11.3 ChebyModelInfo Struct Reference

Collaboration diagram for ChebyModelInfo:



Public Attributes

- [ChebyModel](#) * `model`
- const [pulsar](#) * `psr`
- bool `compute_dispersion_constant`

11.3.1 Member Data Documentation

11.3.1.1 bool `ChebyModelInfo::compute_dispersion_constant`

11.3.1.2 [ChebyModel](#)* `ChebyModelInfo::model`

11.3.1.3 const [pulsar](#)* `ChebyModelInfo::psr`

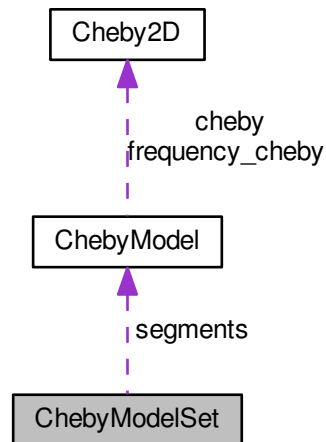
The documentation for this struct was generated from the following file:

- [cheby2d_int.C](#)

11.4 ChebyModelSet Struct Reference

```
#include <tempo2pred.h>
```

Collaboration diagram for ChebyModelSet:



Public Attributes

- [ChebyModel](#) * [segments](#)
- int [nsegments](#)

11.4.1 Member Data Documentation

11.4.1.1 int [ChebyModelSet::nsegments](#)

11.4.1.2 [ChebyModel](#)* [ChebyModelSet::segments](#)

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

- [tempo2pred.h](#)

11.5 clock_correction Struct Reference

```
#include <tempo2.h>
```

Public Attributes

- double [correction](#)
- char [corrects_to](#) [32]

11.5.1 Detailed Description

[observation](#) contains an array of these, which [getClockCorrections\(\)](#) fills in

11.5.2 Member Data Documentation

11.5.2.1 `double clock_correction::correction`

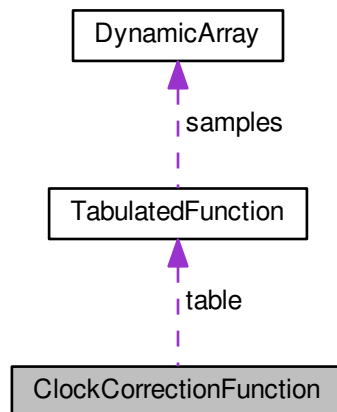
11.5.2.2 `char clock_correction::corrects_to[32]`

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

- [tempo2.h](#)

11.6 ClockCorrectionFunction Struct Reference

Collaboration diagram for ClockCorrectionFunction:



Public Attributes

- [TabulatedFunction table](#)
- `char clockFrom [16]`
- `char clockTo [16]`
- `float badness`

11.6.1 Member Data Documentation

11.6.1.1 `float ClockCorrectionFunction::badness`

11.6.1.2 `char ClockCorrectionFunction::clockFrom[16]`

11.6.1.3 `char ClockCorrectionFunction::clockTo[16]`

11.6.1.4 `TabulatedFunction ClockCorrectionFunction::table`

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

- [clkcorr.C](#)

11.7 complexVal Struct Reference

```
#include <TKspectrum.h>
```

Public Attributes

- double [real](#)
- double [imag](#)

11.7.1 Member Data Documentation

11.7.1.1 double [complexVal::imag](#)

11.7.1.2 double [complexVal::real](#)

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

- [TKspectrum.h](#)

11.8 DynamicArray Struct Reference

```
#include <dynarr.h>
```

Public Attributes

- void * [data](#)
- size_t [nelem](#)
- size_t [elem_size](#)
- size_t [nallocated](#)

11.8.1 Member Data Documentation

11.8.1.1 void* [DynamicArray::data](#)

11.8.1.2 size_t [DynamicArray::elem_size](#)

11.8.1.3 size_t [DynamicArray::nallocated](#)

11.8.1.4 size_t [DynamicArray::nelem](#)

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

- [dynarr.h](#)

11.9 EOPSample Struct Reference

Public Attributes

- double [mjd](#)
- double [xp](#)

- double [yp](#)
- double [dut1](#)

11.9.1 Member Data Documentation

11.9.1.1 double EOPSample::dut1

11.9.1.2 double EOPSample::mjd

11.9.1.3 double EOPSample::xp

11.9.1.4 double EOPSample::yp

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

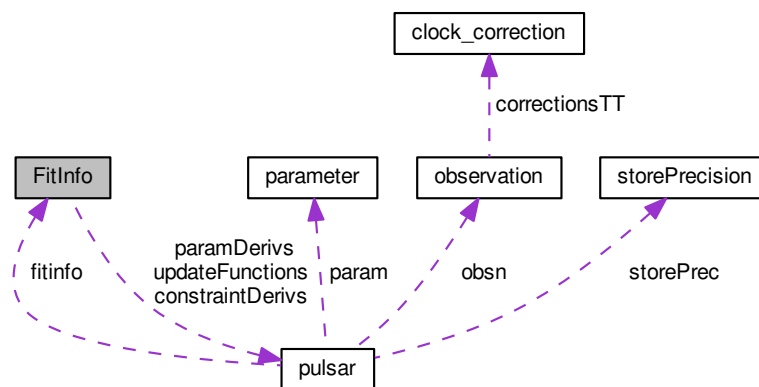
- [eop.C](#)

11.10 FitInfo Struct Reference

contains details of the fit

```
#include <tempo2.h>
```

Collaboration diagram for FitInfo:



Public Attributes

- unsigned [nParams](#)
- unsigned [nConstraints](#)
- [param_label](#) [paramIndex](#) [[MAX_FIT](#)]
- [constraint_label](#) [constraintIndex](#) [[MAX_FIT](#)]
- int [paramCounters](#) [[MAX_FIT](#)]
- int [constraintCounters](#) [[MAX_FIT](#)]
- [paramDerivFunc](#) [paramDerivs](#) [[MAX_FIT](#)]
- [constraintDerivFunc](#) [constraintDerivs](#) [[MAX_FIT](#)]
- [paramUpdateFunc](#) [updateFunctions](#) [[MAX_FIT](#)]

11.10.1 Detailed Description

contains details of the fit

Holds references to the fit functions, as well as references linking the index in the derivative matrix to the actual parameter fit for.

11.10.2 Member Data Documentation

11.10.2.1 `int FitInfo::constraintCounters[MAX_FIT]`

11.10.2.2 `constraintDerivFunc FitInfo::constraintDerivs[MAX_FIT]`

11.10.2.3 `constraint_label FitInfo::constraintIndex[MAX_FIT]`

11.10.2.4 `unsigned FitInfo::nConstraints`

11.10.2.5 `unsigned FitInfo::nParams`

11.10.2.6 `int FitInfo::paramCounters[MAX_FIT]`

11.10.2.7 `paramDerivFunc FitInfo::paramDerivs[MAX_FIT]`

11.10.2.8 `param_label FitInfo::paramIndex[MAX_FIT]`

11.10.2.9 `paramUpdateFunc FitInfo::updateFunctions[MAX_FIT]`

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

- [tempo2.h](#)

11.11 glitchS Struct Reference

Public Attributes

- double [glep](#)
- double [glph](#)
- double [glf0](#)
- double [glf1](#)
- double [glf0d](#)
- double [gltd](#)
- int [fitph](#)
- int [fitt0](#)
- int [fitt1](#)
- int [fitt0d](#)
- int [fittid](#)

11.11.1 Member Data Documentation

11.11.1.1 `int glitchS::fitt0`

11.11.1.2 `int glitchS::fitt0d`

11.11.1.3 int glitchS::fitf1

11.11.1.4 int glitchS::fitph

11.11.1.5 int glitchS::fitted

11.11.1.6 double glitchS::glep

11.11.1.7 double glitchS::glf0

11.11.1.8 double glitchS::glf0d

11.11.1.9 double glitchS::glf1

11.11.1.10 double glitchS::glph

11.11.1.11 double glitchS::gltd

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

- [plugin/glitch_plug.C](#)

11.12 gwgeneralSrc Struct Reference

```
#include <GWsim.h>
```

Public Attributes

- [longdouble theta_g](#)
- [longdouble phi_g](#)
- [longdouble omega_g](#)
- [longdouble phi_polar_g](#)
- [longdouble phase_g](#)
- [longdouble aplus_g](#)
- [longdouble aplus_im_g](#)
- [longdouble across_g](#)
- [longdouble across_im_g](#)
- [longdouble ast_g](#)
- [longdouble ast_im_g](#)
- [longdouble asl_g](#)
- [longdouble asl_im_g](#)
- [longdouble avx_g](#)
- [longdouble avx_im_g](#)
- [longdouble avy_g](#)
- [longdouble avy_im_g](#)
- [longdouble phi_bin](#)
- [longdouble theta_bin](#)
- [longdouble inc_bin](#)
- [longdouble dist_bin](#)
- [longdouble h \[3\]\[3\]](#)
- [longdouble h_im \[3\]\[3\]](#)
- [longdouble kg \[3\]](#)

11.12.1 Member Data Documentation

- 11.12.1.1 `longdouble gwgeneralSrc::across_g`
- 11.12.1.2 `longdouble gwgeneralSrc::across_im_g`
- 11.12.1.3 `longdouble gwgeneralSrc::aplus_g`
- 11.12.1.4 `longdouble gwgeneralSrc::aplus_im_g`
- 11.12.1.5 `longdouble gwgeneralSrc::asl_g`
- 11.12.1.6 `longdouble gwgeneralSrc::asl_im_g`
- 11.12.1.7 `longdouble gwgeneralSrc::ast_g`
- 11.12.1.8 `longdouble gwgeneralSrc::ast_im_g`
- 11.12.1.9 `longdouble gwgeneralSrc::avx_g`
- 11.12.1.10 `longdouble gwgeneralSrc::avx_im_g`
- 11.12.1.11 `longdouble gwgeneralSrc::avy_g`
- 11.12.1.12 `longdouble gwgeneralSrc::avy_im_g`
- 11.12.1.13 `longdouble gwgeneralSrc::dist_bin`
- 11.12.1.14 `longdouble gwgeneralSrc::h[3][3]`
- 11.12.1.15 `longdouble gwgeneralSrc::h_im[3][3]`
- 11.12.1.16 `longdouble gwgeneralSrc::inc_bin`
- 11.12.1.17 `longdouble gwgeneralSrc::kg[3]`
- 11.12.1.18 `longdouble gwgeneralSrc::omega_g`
- 11.12.1.19 `longdouble gwgeneralSrc::phase_g`
- 11.12.1.20 `longdouble gwgeneralSrc::phi_bin`
- 11.12.1.21 `longdouble gwgeneralSrc::phi_g`
- 11.12.1.22 `longdouble gwgeneralSrc::phi_polar_g`
- 11.12.1.23 `longdouble gwgeneralSrc::theta_bin`
- 11.12.1.24 `longdouble gwgeneralSrc::theta_g`

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

- [GWsim.h](#)

11.13 gwgenSpec Struct Reference

```
#include <GWsim.h>
```

Public Attributes

- double [tensor_amp](#)
- double [st_amp](#)
- double [sl_amp](#)
- double [vl_amp](#)
- double [tensor_alpha](#)
- double [st_alpha](#)
- double [sl_alpha](#)
- double [vl_alpha](#)

11.13.1 Member Data Documentation

11.13.1.1 double `gwgenSpec::sl_alpha`

11.13.1.2 double `gwgenSpec::sl_amp`

11.13.1.3 double `gwgenSpec::st_alpha`

11.13.1.4 double `gwgenSpec::st_amp`

11.13.1.5 double `gwgenSpec::tensor_alpha`

11.13.1.6 double `gwgenSpec::tensor_amp`

11.13.1.7 double `gwgenSpec::vl_alpha`

11.13.1.8 double `gwgenSpec::vl_amp`

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

- [GWsim.h](#)

11.14 gwSrc Struct Reference

```
#include <GWsim.h>
```

Public Attributes

- [longdouble theta_g](#)
- [longdouble phi_g](#)
- [longdouble omega_g](#)
- [longdouble phi_polar_g](#)
- [longdouble phase_g](#)
- [longdouble aplus_g](#)
- [longdouble aplus_im_g](#)
- [longdouble across_g](#)
- [longdouble across_im_g](#)

- [longdouble phi_bin](#)
- [longdouble theta_bin](#)
- [longdouble inc_bin](#)
- [longdouble dist_bin](#)
- [longdouble h \[3\]\[3\]](#)
- [longdouble h_im \[3\]\[3\]](#)
- [longdouble kg \[3\]](#)

11.14.1 Member Data Documentation

11.14.1.1 [longdouble gwSrc::across_g](#)

11.14.1.2 [longdouble gwSrc::across_im_g](#)

11.14.1.3 [longdouble gwSrc::aplus_g](#)

11.14.1.4 [longdouble gwSrc::aplus_im_g](#)

11.14.1.5 [longdouble gwSrc::dist_bin](#)

11.14.1.6 [longdouble gwSrc::h\[3\]\[3\]](#)

11.14.1.7 [longdouble gwSrc::h_im\[3\]\[3\]](#)

11.14.1.8 [longdouble gwSrc::inc_bin](#)

11.14.1.9 [longdouble gwSrc::kg\[3\]](#)

11.14.1.10 [longdouble gwSrc::omega_g](#)

11.14.1.11 [longdouble gwSrc::phase_g](#)

11.14.1.12 [longdouble gwSrc::phi_bin](#)

11.14.1.13 [longdouble gwSrc::phi_g](#)

11.14.1.14 [longdouble gwSrc::phi_polar_g](#)

11.14.1.15 [longdouble gwSrc::theta_bin](#)

11.14.1.16 [longdouble gwSrc::theta_g](#)

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

- [GWsim.h](#)

11.15 IFTE_interpolation_info Struct Reference

Public Attributes

- double [pc](#) [18]
- double [vc](#) [18]
- double [twot](#)
- int [np](#)
- int [nv](#)

11.15.1 Member Data Documentation

11.15.1.1 `int IFTE_interpolation_info::np`

11.15.1.2 `int IFTE_interpolation_info::nv`

11.15.1.3 `double IFTE_interpolation_info::pc[18]`

11.15.1.4 `double IFTE_interpolation_info::twot`

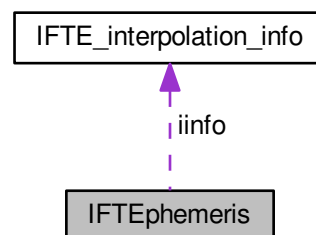
11.15.1.5 `double IFTE_interpolation_info::vc[18]`

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

- [ifteph.C](#)

11.16 IFTEphemeris Struct Reference

Collaboration diagram for IFTEphemeris:



Public Attributes

- `char title [256]`
- `double startJD`
- `double endJD`
- `double stepJD`
- `int ephver`
- `double L_C`
- `int swap_endian`
- `int reclen`
- `int irec`
- `double buf [322]`
- `FILE * f`
- `struct IFTE_interpolation_info iinfo`
- `int ipt [2][3]`

11.16.1 Member Data Documentation

- 11.16.1.1 `double IFTEphemeris::buf[322]`
- 11.16.1.2 `double IFTEphemeris::endJD`
- 11.16.1.3 `int IFTEphemeris::ephver`
- 11.16.1.4 `FILE* IFTEphemeris::f`
- 11.16.1.5 `struct IFTE_interpolation_info IFTEphemeris::iinfo`
- 11.16.1.6 `int IFTEphemeris::ipt[2][3]`
- 11.16.1.7 `int IFTEphemeris::irec`
- 11.16.1.8 `double IFTEphemeris::L_C`
- 11.16.1.9 `int IFTEphemeris::reclen`
- 11.16.1.10 `double IFTEphemeris::startJD`
- 11.16.1.11 `double IFTEphemeris::stepJD`
- 11.16.1.12 `int IFTEphemeris::swap_endian`
- 11.16.1.13 `char IFTEphemeris::title[256]`

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

- [ifteph.C](#)

11.17 interpolation_info Struct Reference

```
#include <jpl_int.h>
```

Public Attributes

- `double pc[18]`
- `double vc[18]`
- `double twot`
- `int np`
- `int nv`

11.17.1 Member Data Documentation

- 11.17.1.1 `int interpolation_info::np`
- 11.17.1.2 `int interpolation_info::nv`
- 11.17.1.3 `double interpolation_info::pc[18]`
- 11.17.1.4 `double interpolation_info::twot`

11.17.1.5 double interpolation_info::vc[18]

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

- [jpl_int.h](#)

11.18 jpl_eph_data Struct Reference

```
#include <jpl_int.h>
```

Public Attributes

- double [ephem_start](#)
- double [ephem_end](#)
- double [ephem_step](#)
- [JPLlong](#) [ncon](#)
- double [au](#)
- double [emrat](#)
- [JPLlong](#) [ipt](#) [13][3]
- [JPLlong](#) [ephemeris_version](#)
- [JPLlong](#) [kernel_size](#)
- [JPLlong](#) [recsize](#)
- [JPLlong](#) [ncoeff](#)
- [JPLlong](#) [swap_bytes](#)
- [JPLlong](#) [curr_cache_loc](#)
- double [pvsun](#) [6]
- double * [cache](#)
- void * [iinfo](#)
- FILE * [ifile](#)

11.18.1 Member Data Documentation

11.18.1.1 double [jpl_eph_data::au](#)

11.18.1.2 double* [jpl_eph_data::cache](#)

11.18.1.3 [JPLlong](#) [jpl_eph_data::curr_cache_loc](#)

11.18.1.4 double [jpl_eph_data::emrat](#)

11.18.1.5 double [jpl_eph_data::ephem_end](#)

11.18.1.6 double [jpl_eph_data::ephem_start](#)

11.18.1.7 double [jpl_eph_data::ephem_step](#)

11.18.1.8 [JPLlong](#) [jpl_eph_data::ephemeris_version](#)

11.18.1.9 FILE* [jpl_eph_data::ifile](#)

11.18.1.10 void* [jpl_eph_data::iinfo](#)

11.18.1.11 **JPLlong** jpl_eph_data::ipt[13][3]

11.18.1.12 **JPLlong** jpl_eph_data::kernel_size

11.18.1.13 **JPLlong** jpl_eph_data::ncoeff

11.18.1.14 **JPLlong** jpl_eph_data::ncon

11.18.1.15 **double** jpl_eph_data::pvsun[6]

11.18.1.16 **JPLlong** jpl_eph_data::recsize

11.18.1.17 **JPLlong** jpl_eph_data::swap_bytes

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

- [jpl_int.h](#)

11.19 Im_control_struct Struct Reference

Public Attributes

- **double** [ftol](#)
- **double** [xtol](#)
- **double** [gtol](#)
- **double** [epsilon](#)
- **double** [stepbound](#)
- **int** [maxcall](#)
- **int** [scale_diag](#)
- **int** [printflags](#)

11.19.1 Member Data Documentation

11.19.1.1 **double** Im_control_struct::epsilon

11.19.1.2 **double** Im_control_struct::ftol

11.19.1.3 **double** Im_control_struct::gtol

11.19.1.4 **int** Im_control_struct::maxcall

11.19.1.5 **int** Im_control_struct::printflags

11.19.1.6 **int** Im_control_struct::scale_diag

11.19.1.7 **double** Im_control_struct::stepbound

11.19.1.8 **double** Im_control_struct::xtol

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

- [plugin/glitch_plug.C](#)

11.20 lm_status_struct Struct Reference

Public Attributes

- double [fnorm](#)
- int [nfev](#)
- int [info](#)

11.20.1 Member Data Documentation

11.20.1.1 double lm_status_struct::fnorm

11.20.1.2 int lm_status_struct::info

11.20.1.3 int lm_status_struct::nfev

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

- [plugin/glitch_plug.C](#)

11.21 lmcurve_data_struct Struct Reference

Public Attributes

- const double * [t](#)
- const double * [y](#)
- double(* [f](#))(double [t](#), const double *par, int obsNum)

11.21.1 Member Data Documentation

11.21.1.1 double(* lmcurve_data_struct::f)(double [t](#), const double *par, int obsNum)

11.21.1.2 const double* lmcurve_data_struct::t

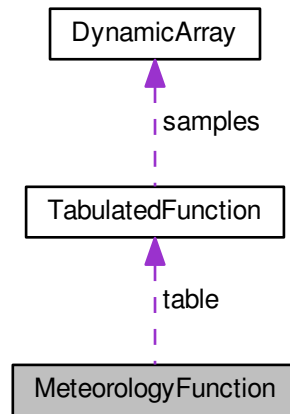
11.21.1.3 const double* lmcurve_data_struct::y

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

- [plugin/glitch_plug.C](#)

11.22 MeteorologyFunction Struct Reference

Collaboration diagram for MeteorologyFunction:



Public Attributes

- [TabulatedFunction table](#)
- char [siteName](#) [256]

11.22.1 Member Data Documentation

11.22.1.1 char `MeteorologyFunction::siteName`[256]

11.22.1.2 `TabulatedFunction MeteorologyFunction::table`

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

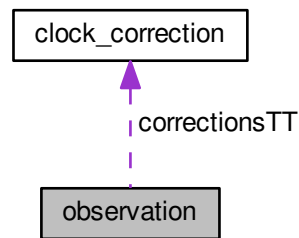
- [tropo.C](#)

11.23 observation Struct Reference

A struct containing the details of a single obesrvation.

```
#include <tempo2.h>
```

Collaboration diagram for observation:



Public Attributes

- [longdouble](#) `sat`
- [longdouble](#) `origsat`
- [longdouble](#) `sat_day`
- [longdouble](#) `sat_sec`
- [longdouble](#) `bat`
- [longdouble](#) `batCorr`
- [longdouble](#) `bbat`
- [longdouble](#) `pet`
- [int](#) `clockCorr`
- [int](#) `delayCorr`
- [int](#) `deleted`
- [longdouble](#) `prefitResidual`
- [longdouble](#) `residual`
- [double](#) `addedNoise`
- [double](#) `TNRedSignal`
- [double](#) `TNRedErr`
- [double](#) `TNDMSignal`
- [double](#) `TNDMErr`
- [double](#) `TNGroupSignal`
- [double](#) `TNGroupErr`
- [double](#) `freq`
- [double](#) `freqSSB`
- [double](#) `toaErr`
- [double](#) `toaDMErr`
- [double](#) `origErr`
- [double](#) `phaseOffset`
- [double](#) `averagebat`
- [double](#) `averageres`
- [double](#) `averageerr`
- [char](#) `fname` [`MAX_FILELEN`]
- [char](#) `telID` [100]
- [clock_correction](#) `correctionsTT` [`MAX_CLK_CORR`]
- [int](#) `nclock_correction`
- [longdouble](#) `correctionTT_TB`
- [double](#) `einsteinRate`

- [longdouble correctionTT_Teph](#)
- [longdouble correctionUT1](#)
- [double sun_ssb](#) [6]
- [double sun_earth](#) [6]
- [double planet_ssb](#) [9][6]
- [double jupiter_earth](#) [6]
- [double saturn_earth](#) [6]
- [double venus_earth](#) [6]
- [double uranus_earth](#) [6]
- [double neptune_earth](#) [6]
- [double earthMoonBary_ssb](#) [6]
- [double earthMoonBary_earth](#) [6]
- [double earth_ssb](#) [6]
- [double observatory_earth](#) [6]
- [double psrPos](#) [3]
- [double zenith](#) [3]
- [double nutations](#) [6]
- [double siteVel](#) [3]
- [longdouble shklovskii](#)
- [double shapiroDelaySun](#)
- [double shapiroDelayJupiter](#)
- [double shapiroDelaySaturn](#)
- [double shapiroDelayVenus](#)
- [double shapiroDelayUranus](#)
- [double shapiroDelayNeptune](#)
- [double troposphericDelay](#)
- [double tdis1](#)
- [double tdis2](#)
- [longdouble roemer](#)
- [longdouble torb](#)
- [longdouble nphase](#)
- [longdouble phase](#)
- [long long pulseN](#)
- [char flagID](#) [MAX_FLAGS][MAX_FLAG_LEN]
- [char flagVal](#) [MAX_FLAGS][MAX_FLAG_LEN]
- [int nFlags](#)
- [int jump](#) [MAX_FLAGS]
- [int obsNjump](#)
- [double efac](#)
- [double equad](#)

11.23.1 Detailed Description

A struct containing the details of a single obesrvation.

11.23.2 Member Data Documentation

11.23.2.1 [double observation::addedNoise](#)

11.23.2.2 [double observation::averagebat](#)

11.23.2.3 [double observation::averageerr](#)

11.23.2.4 `double observation::averageres`

11.23.2.5 `longdouble observation::bat`

Infinite frequency barycentric arrival time

11.23.2.6 `longdouble observation::batCorr`

11.23.2.7 `longdouble observation::bbat`

Arrival time at binary barycentre

11.23.2.8 `int observation::clockCorr`

= 1 for clock corrections to be applied, = 0 for BAT

11.23.2.9 `clock_correction observation::correctionsTT[MAX_CLK_CORR]`

chain of corrections from site TOA to chosen realisation of TT

11.23.2.10 `longdouble observation::correctionTT_TB`

Correction to TDB/TCB

11.23.2.11 `longdouble observation::correctionTT_Teph`

Correction to Teph

11.23.2.12 `longdouble observation::correctionUT1`

Correction from site TOA to UT1

11.23.2.13 `int observation::delayCorr`

= 1 for time delay corrections to be applied, = 0 for BAT

11.23.2.14 `int observation::deleted`

= 1 if observation has been deleted, = -1 if not included in fit

11.23.2.15 `double observation::earth_ssb[6]`

Centre of Earth w.r.t. SSB

11.23.2.16 `double observation::earthMoonBary_earth[6]`

Position of Earth-Moon barycentre with respect to Earth (sec) (RBE)

11.23.2.17 `double observation::earthMoonBary_ssb[6]`

Ephem values for Earth-Moon barycentre wrt SSB (sec) (RCB)

11.23.2.18 `double observation::efac`

Error multiplication factor

11.23.2.19 `double observation::einsteinRate`

Derivative of correctionTT_TB

11.23.2.20 `double observation::equad`

Value to add in quadrature

11.23.2.21 `char observation::flagID[MAX_FLAGS][MAX_FLAG_LEN]`

Flags in .tim file

11.23.2.22 `char observation::flagVal[MAX_FLAGS][MAX_FLAG_LEN]`

11.23.2.23 `char observation::fname[MAX_FILELEN]`

Name of data file giving TOA

11.23.2.24 `double observation::freq`

Frequency of observation (in MHz)

11.23.2.25 `double observation::freqSSB`

Frequency of observation in barycentric frame (in Hz)

11.23.2.26 `int observation::jump[MAX_FLAGS]`

Jump region

11.23.2.27 `double observation::jupiter_earth[6]`

Ephemeris values for Jupiter w.r.t. Earth centre (sec)

11.23.2.28 `int observation::nclock_correction`

11.23.2.29 `double observation::neptune_earth[6]`

Ephemeris values for Neptune w.r.t. Earth centre (sec)

11.23.2.30 `int observation::nFlags`

11.23.2.31 `longdouble observation::nphase`

allows the pulse number to be determined

11.23.2.32 `double observation::nutations[6]`

11.23.2.33 `double observation::observatory_earth[6]`

Observatory site with respect to Earth centre (sec) (REA)

11.23.2.34 `int observation::obsNjump`

Number of jumps for this observation

11.23.2.35 `double observation::origErr`

Original error on TOA after reading tim file (in us)

11.23.2.36 `longdouble observation::origsat`

11.23.2.37 `longdouble observation::pet`

Pulsar emission time

11.23.2.38 `longdouble observation::phase`

11.23.2.39 `double observation::phaseOffset`

Phase offset

11.23.2.40 `double observation::planet_ssb[9][6]`

Ephemeris values for all planets w.r.t. SSB (sec)

11.23.2.41 `longdouble observation::prefitResidual`

Pre-fit residual

11.23.2.42 `double observation::psrPos[3]`

Unit vector giving position of the pulsar at observation time from Earth

11.23.2.43 `long long observation::pulseN`

Pulse number

11.23.2.44 `longdouble observation::residual`

residual

11.23.2.45 longdouble observation::roemer

Roemer delay

11.23.2.46 longdouble observation::sat

Site arrival time

11.23.2.47 longdouble observation::sat_day**11.23.2.48 longdouble observation::sat_sec****11.23.2.49 double observation::saturn_earth[6]**

Ephemeris values for Saturn w.r.t. Earth centre (sec)

11.23.2.50 double observation::shapiroDelayJupiter

Shapiro Delay due to Jupiter

11.23.2.51 double observation::shapiroDelayNeptune

Shapiro Delay due to Neptune

11.23.2.52 double observation::shapiroDelaySaturn

Shapiro Delay due to Saturn

11.23.2.53 double observation::shapiroDelaySun

Shapiro Delay due to the Sun

11.23.2.54 double observation::shapiroDelayUranus

Shapiro Delay due to Uranus

11.23.2.55 double observation::shapiroDelayVenus

Shapiro Delay due to Venus

11.23.2.56 longdouble observation::shklovskii

Shklovskii delay term

11.23.2.57 double observation::siteVel[3]

Observatory velocity w.r.t. geocentre

11.23.2.58 double observation::sun_earth[6]

Ephemeris values for Sun w.r.t Earth (sec)

11.23.2.59 double observation::sun_ssb[6]

Ephemeris values for Sun w.r.t SSB (sec) (RCS)

11.23.2.60 double observation::tdis1

Interstellar dispersion measure delay

11.23.2.61 double observation::tdis2

Dispersion measure delay due to solar system

11.23.2.62 char observation::telID[100]

Telescope ID

11.23.2.63 double observation::TNDMErr

Error on Model DM signal from temponest fit

11.23.2.64 double observation::TNDMSignal

Model DM signal from temponest fit

11.23.2.65 double observation::TNGroupErr

Error on Model Group Noise signal from temponest fit

11.23.2.66 double observation::TNGroupSignal

Model Group Noise signal from temponest fit

11.23.2.67 double observation::TNRedErr

Error on Model red noise signal from temponest fit

11.23.2.68 double observation::TNRedSignal

Model red noise signal from temponest fit

11.23.2.69 double observation::toaDMErr

Error on TOA due to DM (in us)

11.23.2.70 double observation::toaErr

Error on TOA (in us)

11.23.2.71 longdouble observation::torb

Combined binary delays

11.23.2.72 double observation::troposphericDelay

Delay due to neutral refraction in atmosphere

11.23.2.73 double observation::uranus_earth[6]

Ephemeris values for Uranus w.r.t. Earth centre (sec)

11.23.2.74 double observation::venus_earth[6]

Ephemeris values for Venus w.r.t. Earth centre (sec)

11.23.2.75 double observation::zenith[3]

Zenith vector, in BC frame. Length=geodetic height

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

- [tempo2.h](#)

11.24 observatory Struct Reference

```
#include <tempo2.h>
```

Public Attributes

- double [x](#)
- double [y](#)
- double [z](#)
- double [longitude_grs80](#)
- double [latitude_grs80](#)
- double [height_grs80](#)
- char [name](#) [32]
- char [code](#) [16]
- char [clock_name](#) [16]

11.24.1 Member Data Documentation

11.24.1.1 char observatory::clock_name[16]

11.24.1.2 char observatory::code[16]

11.24.1.3 double observatory::height_grs80

11.24.1.4 double observatory::latitude_grs80

11.24.1.5 double observatory::longitude_grs80

11.24.1.6 char observatory::name[32]

11.24.1.7 double observatory::x

11.24.1.8 double observatory::y

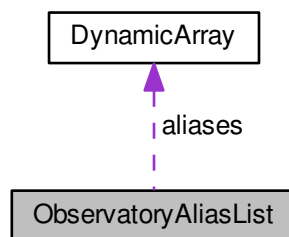
11.24.1.9 double observatory::z

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

- [tempo2.h](#)

11.25 ObservatoryAliasList Struct Reference

Collaboration diagram for ObservatoryAliasList:



Public Attributes

- char [code](#) [100]
- [DynamicArray](#) [aliases](#)

11.25.1 Member Data Documentation

11.25.1.1 [DynamicArray](#) [ObservatoryAliasList::aliases](#)

11.25.1.2 char [ObservatoryAliasList::code](#)[100]

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

- [observatory.C](#)

11.26 parameter Struct Reference

Holds the values for a parameter.

```
#include <tempo2.h>
```

Public Attributes

- char ** [label](#)
- char ** [shortlabel](#)
- [longdouble](#) * [val](#)
- [longdouble](#) * [err](#)
- int * [fitFlag](#)
- int * [paramSet](#)
- [longdouble](#) * [prefit](#)
- [longdouble](#) * [prefitErr](#)
- int [aSize](#)
- int [linkFrom](#) [5]
- int [linkTo](#) [5]
- int [nLinkTo](#)
- int [nLinkFrom](#)

11.26.1 Detailed Description

Holds the values for a parameter.

May include multiple values, for e.g. F0, F1, F2,...

Note

If this structure is modified - must update copyParam in [tempo2Util.C](#)

11.26.2 Member Data Documentation

11.26.2.1 int parameter::aSize

Number of elements in the array for this parameter

11.26.2.2 longdouble* parameter::err

Uncertainty on parameter value

11.26.2.3 int* parameter::fitFlag

= 1 if fitting required, = 2 for global fit

11.26.2.4 char** parameter::label

Label about this parameter

11.26.2.5 int parameter::linkFrom[5]

11.26.2.6 int parameter::linkTo[5]

11.26.2.7 int parameter::nLinkFrom

11.26.2.8 int parameter::nLinkTo

11.26.2.9 int* parameter::paramSet

= 1 if parameter has been set

11.26.2.10 longdouble* parameter::prefit

Pre-fit value of the parameter

11.26.2.11 longdouble* parameter::prefitErr

Pre-fit value of the uncertainty

11.26.2.12 char** parameter::shortlabel

Label about this parameter without units

11.26.2.13 longdouble* parameter::val

Value of parameter

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

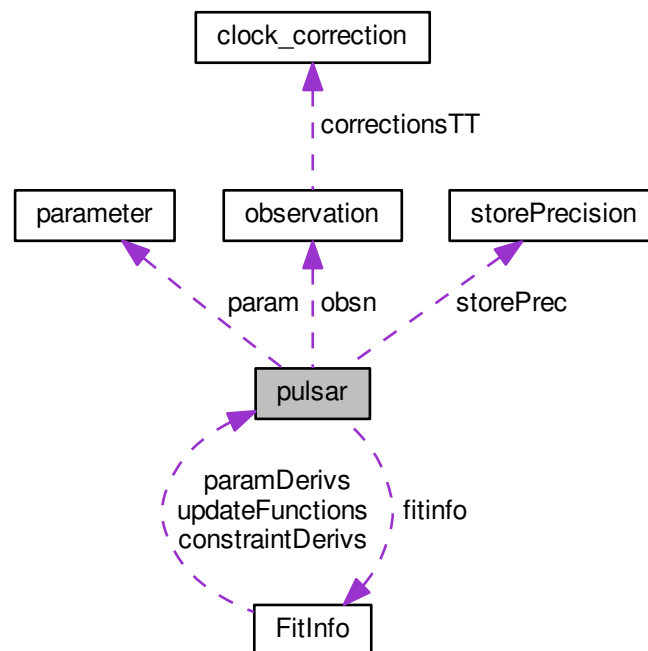
- [tempo2.h](#)

11.27 pulsar Struct Reference

contains the details for a single pulsar.

```
#include <tempo2.h>
```

Collaboration diagram for pulsar:



Public Attributes

- char [name](#) [100]
- char [eopc04_file](#) [MAX_FILELEN]
- int [fixedFormat](#)
- [parameter](#) [param](#) [MAX_PARAMS]
- char [rajStrPre](#) [100]
- char [decjStrPre](#) [100]
- char [rajStrPost](#) [100]
- char [decjStrPost](#) [100]
- char [binaryModel](#) [100]
- double ** [ToAextraCovar](#)
- int [dmoffsDMnum](#)
- int [dmoffsCMnum](#)
- double [dmoffsDM_mjd](#) [MAX_IFUNC]
- double [dmoffsDM](#) [MAX_IFUNC]
- double [dmoffsDM_error](#) [MAX_IFUNC]
- double [dmoffsDM_weight](#) [MAX_IFUNC]
- double [dmoffsCM_mjd](#) [MAX_IFUNC]
- double [dmoffsCM](#) [MAX_IFUNC]
- double [dmoffsCM_error](#) [MAX_IFUNC]
- double [dmoffsCM_weight](#) [MAX_IFUNC]
- double [gwsrsrc_ra](#)
- double [gwsrsrc_dec](#)
- double [gwsrsrc_aplus_r](#)

- double [gwsrc_aplus_i](#)
- double [gwsrc_across_r](#)
- double [gwsrc_across_i](#)
- double [gwsrc_aplus_r_e](#)
- double [gwsrc_aplus_i_e](#)
- double [gwsrc_across_r_e](#)
- double [gwsrc_across_i_e](#)
- double [gwsrc_epoch](#)
- double [gwsrc_psrdist](#)
- double [cgw_h0](#)
- double [cgw_cosinc](#)
- double [cgw_angpol](#)
- double [cgw_mc](#)
- double [gwm_raj](#)
- double [gwm_decj](#)
- double [gwm_epoch](#)
- double [gwm_phi](#)
- double [gwm_dphase](#)
- double [gwb_epoch](#)
- double [gwb_width](#)
- double [gwb_raj](#)
- double [gwb_decj](#)
- double [gwb_geom_c](#)
- double [gwb_geom_p](#)
- double [gwecc_ra](#)
- double [gwecc_dec](#)
- double [gwecc_m1](#)
- double [gwecc_m2](#)
- double [gwecc_e](#)
- double [gwecc_inc](#)
- double [gwecc_theta_nodes](#)
- double [gwecc_nodes_orientation](#)
- double [gwecc_theta_0](#)
- double [gwecc_orbital_period](#)
- double [gwecc_distance](#)
- double [gwecc_redshift](#)
- double [gwecc_epoch](#)
- double [gwecc_psrdist](#)
- int [gwecc_pulsarTermOn](#)
- double [posPulsar](#) [3]
- double [velPulsar](#) [3]
- longdouble [phaseJump](#) [MAX_JUMPS]
- int [phaseJumpDir](#) [MAX_JUMPS]
- int [phaseJumpID](#) [MAX_JUMPS]
- int [nPhaseJump](#)
- double [dmOffset](#)
- double [ne_sw](#)
- int [nCompanion](#)
- int [eclCoord](#)
- int [nJumps](#)
- char [fjumpID](#) [16]
- double [jumpVal](#) [MAX_JUMPS]
- int [fitJump](#) [MAX_JUMPS]
- double [jumpValErr](#) [MAX_JUMPS]
- char [jumpStr](#) [MAX_JUMPS][MAX_STRLEN]

- char `filterStr` [MAX_STRLEN]
- char `passStr` [MAX_STRLEN]
- double `tOffset` [MAX_TOFFSET]
- double `tOffset_f1` [MAX_TOFFSET]
- double `tOffset_f2` [MAX_TOFFSET]
- double `tOffset_t1` [MAX_TOFFSET]
- double `tOffset_t2` [MAX_TOFFSET]
- char `tOffsetSite` [MAX_TOFFSET][100]
- char `tOffsetFlags` [MAX_TOFFSET][1000]
- int `nToffset`
- int `ndmx`
- double `fitChisq`
- int `fitNfree`
- int `globalNfit`
- int `globalNoConstrain`
- int `nFit`
- int `nParam`
- int `nGlobal`
- int `fitParamGlobalI` [MAX_FIT]
- int `fitParamGlobalK` [MAX_FIT]
- int `fitParamI` [MAX_FIT]
- int `fitParamK` [MAX_FIT]
- int `fitMode`
- char `robust`
- int `rescaleErrChisq`
- double `offset`
- double `offset_e`
- double ** `covar`
- int `calcShapiro`
- int `planetShapiro`
- int `jboFormat`
- `observation * obsn`
- int `nobs`
- int `units`
- int `setUnits`
- int `tempo1`
- int `dilateFreq`
- int `timeEphemeris`
- int `t2cMethod`
- int `correctTroposphere`
- int `noWarnings`
- char `sorted`
- char `clock` [16]
- char `clockFromOverride` [64]
- char `JPL_EPHEMERIS` [MAX_FILELEN]
- char `ephemeris` [MAX_FILELEN]
- int `useCalceph`
- `storePrecision storePrec` [MAX_STOREPRECISION]
- int `nStorePrecision`
- int `bootStrap`
- char `tzrsite` [100]
- double `rmsPre`
- double `rmsPost`
- char `deleteFileName` [100]
- int `nits`

- int [ipm](#)
- int [swm](#)
- double [wave_sine](#) [MAX_WHITE]
- double [wave_sine_err](#) [MAX_WHITE]
- double [wave_cos](#) [MAX_WHITE]
- double [wave_cos_err](#) [MAX_WHITE]
- double [wave_sine_dm](#) [MAX_WHITE]
- double [wave_sine_dm_err](#) [MAX_WHITE]
- double [wave_cos_dm](#) [MAX_WHITE]
- double [wave_cos_dm_err](#) [MAX_WHITE]
- int [nWhite](#)
- int [nWhite_dm](#)
- double [waveScale](#)
- double [quad_aplus_r](#) [MAX_QUAD]
- double [quad_aplus_r_e](#) [MAX_QUAD]
- double [quad_aplus_i](#) [MAX_QUAD]
- double [quad_aplus_i_e](#) [MAX_QUAD]
- double [quad_across_r](#) [MAX_QUAD]
- double [quad_across_r_e](#) [MAX_QUAD]
- double [quad_across_i](#) [MAX_QUAD]
- double [quad_across_i_e](#) [MAX_QUAD]
- double [quadEpoch](#)
- double [quadRA](#)
- double [quadDEC](#)
- int [nQuad](#)
- double [ifuncT](#) [MAX_IFUNC]
- double [ifuncV](#) [MAX_IFUNC]
- double [ifuncE](#) [MAX_IFUNC]
- double [ifunc_weights](#) [MAX_IFUNC]
- int [ifuncN](#)
- double [clk_offsT](#) [MAX_TEL_CLK_OFFS]
- double [clk_offsV](#) [MAX_TEL_CLK_OFFS]
- double [clk_offsE](#) [MAX_TEL_CLK_OFFS]
- int [clkOffsN](#)
- double [quad_ifuncT_p](#) [MAX_IFUNC]
- double [quad_ifuncV_p](#) [MAX_IFUNC]
- double [quad_ifuncE_p](#) [MAX_IFUNC]
- int [quad_ifuncN_p](#)
- double [quad_ifuncT_c](#) [MAX_IFUNC]
- double [quad_ifuncV_c](#) [MAX_IFUNC]
- double [quad_ifuncE_c](#) [MAX_IFUNC]
- int [quad_ifuncN_c](#)
- double [quad_ifunc_p_RA](#)
- double [quad_ifunc_p_DEC](#)
- double [quad_ifunc_c_RA](#)
- double [quad_ifunc_c_DEC](#)
- double [quad_ifunc_geom_p](#)
- double [quad_ifunc_geom_c](#)
- int [nTelDX](#)
- int [setTelVelX](#)
- double [telDX_t](#) [MAX_TEL_DX]
- double [telDX_v](#) [MAX_TEL_DX]
- double [telDX_e](#) [MAX_TEL_DX]
- double [telDX_vel](#) [MAX_TEL_DX]
- double [telDX_vel_e](#) [MAX_TEL_DX]

- int [nTelDY](#)
- int [setTelVelY](#)
- double [telDY_t](#) [MAX_TEL_DY]
- double [telDY_v](#) [MAX_TEL_DY]
- double [telDY_e](#) [MAX_TEL_DY]
- double [telDY_vel](#) [MAX_TEL_DY]
- double [telDY_vel_e](#) [MAX_TEL_DY]
- int [nTelDZ](#)
- int [setTelVelZ](#)
- double [telDZ_v](#) [MAX_TEL_DZ]
- double [telDZ_t](#) [MAX_TEL_DZ]
- double [telDZ_e](#) [MAX_TEL_DZ]
- double [telDZ_vel](#) [MAX_TEL_DZ]
- double [telDZ_vel_e](#) [MAX_TEL_DZ]
- int [nT2efac](#)
- int [nT2equad](#)
- char [T2efacFlagID](#) [MAX_T2EFAC][MAX_FLAG_LEN]
- char [T2efacFlagVal](#) [MAX_T2EFAC][MAX_FLAG_LEN]
- double [T2efacVal](#) [MAX_T2EFAC]
- char [T2equadFlagID](#) [MAX_T2EQUAD][MAX_FLAG_LEN]
- char [T2equadFlagVal](#) [MAX_T2EQUAD][MAX_FLAG_LEN]
- double [T2equadVal](#) [MAX_T2EQUAD]
- double [T2globalEfac](#)
- int [nTNEF](#)
- int [nTNEQ](#)
- int [nTNSQ](#)
- int [nTNECORR](#)
- char [TNEFFlagID](#) [MAX_TNEF][MAX_FLAG_LEN]
- char [TNEFFlagVal](#) [MAX_TNEF][MAX_FLAG_LEN]
- double [TNEFVal](#) [MAX_TNEF]
- double [TNGlobalEF](#)
- char [TNEQFlagID](#) [MAX_TNEQ][MAX_FLAG_LEN]
- char [TNEQFlagVal](#) [MAX_TNEQ][MAX_FLAG_LEN]
- double [TNEQVal](#) [MAX_TNEQ]
- double [TNGlobalEQ](#)
- double [addTNGlobalEQ](#)
- char [TNSQFlagID](#) [MAX_TNSQ][MAX_FLAG_LEN]
- char [TNSQFlagVal](#) [MAX_TNSQ][MAX_FLAG_LEN]
- double [TNSQVal](#) [MAX_TNSQ]
- char [TNECORRFlagID](#) [MAX_TNECORR][MAX_FLAG_LEN]
- char [TNECORRFlagVal](#) [MAX_TNECORR][MAX_FLAG_LEN]
- double [TNECORRVal](#) [MAX_TNECORR]
- double [TNRedAmp](#)
- double [TNRedGam](#)
- int [TNRedC](#)
- double [TNRedCoeffs](#) [200]
- double [TNRedFlow](#)
- double [TNRedCorner](#)
- double [TNDMAmp](#)
- double [TNDMGam](#)
- int [TNDMC](#)
- double [TNDMCoeffs](#) [200]
- int [TNsubtractDM](#)
- int [TNsubtractRed](#)
- int [AverageResiduals](#)

- char [AverageFlag](#) [MAX_FLAG_LEN]
- float [AverageEpochWidth](#)
- int [outputTMatrix](#)
- int [useTNOrth](#)
- double [TNBandDMAmp](#)
- double [TNBandDMGam](#)
- int [TNBandDMC](#)
- int [nTNBandNoise](#)
- double [TNBandNoiseLF](#) [MAX_TNBN]
- double [TNBandNoiseHF](#) [MAX_TNBN]
- double [TNBandNoiseAmp](#) [MAX_TNBN]
- double [TNBandNoiseGam](#) [MAX_TNBN]
- int [TNBandNoiseC](#) [MAX_TNBN]
- int [nTNGroupNoise](#)
- char [TNGroupNoiseFlagID](#) [MAX_TNGN][MAX_FLAG_LEN]
- char [TNGroupNoiseFlagVal](#) [MAX_TNGN][MAX_FLAG_LEN]
- double [TNGroupNoiseAmp](#) [MAX_TNGN]
- double [TNGroupNoiseGam](#) [MAX_TNGN]
- int [TNGroupNoiseC](#) [MAX_TNGN]
- int [nDMEvents](#)
- double [TNDMEvStart](#) [MAX_TNDMEv]
- double [TNDMEvLength](#) [MAX_TNDMEv]
- double [TNDMEvAmp](#) [MAX_TNDMEv]
- double [TNDMEvGam](#) [MAX_TNDMEv]
- int [TNDMEvOff](#) [MAX_TNDMEv]
- int [TNDMEvLin](#) [MAX_TNDMEv]
- int [TNDMEvQuad](#) [MAX_TNDMEv]
- int [nTNShapeletEvents](#)
- int [TNShapeletEvN](#) [MAX_TNDMEv]
- double [TNShapeletEvPos](#) [MAX_TNDMEv]
- double [TNShapeletEvWidth](#) [MAX_TNDMEv]
- double [TNShapeletEvFScale](#) [MAX_TNDMEv]
- char [whiteNoiseModelFile](#) [MAX_STRLEN]
- double [rasim](#)
- double [decsim](#)
- int [simflag](#)
- char [fitFunc](#) [MAX_FILELEN]
- int [nconstraints](#)
- enum [constraint constraints](#) [MAX_PARAMS]
- char [auto_constraints](#)
- [FitInfo](#) [fitinfo](#)

11.27.1 Detailed Description

contains the details for a single pulsar.

Includes an array of [observations](#) and [parameters](#)

11.27.2 Member Data Documentation

11.27.2.1 `double pulsar::addTNGlobalEQ`

11.27.2.2 `char pulsar::auto_constraints`

11.27.2.3 `float pulsar::AverageEpochWidth`

11.27.2.4 `char pulsar::AverageFlag[MAX_FLAG_LEN]`

11.27.2.5 `int pulsar::AverageResiduals`

11.27.2.6 `char pulsar::binaryModel[100]`

Binary model e.g. BT/ELL1/BT2P etc.

11.27.2.7 `int pulsar::bootStrap`

0 if calculating errors using bootstrap Monte-Carlo method

11.27.2.8 `int pulsar::calcShapiro`

= 1 Calculate Solar system Shapiro delay (otherwise -1)

11.27.2.9 `double pulsar::cgw_angpol`

11.27.2.10 `double pulsar::cgw_cosinc`

11.27.2.11 `double pulsar::cgw_h0`

11.27.2.12 `double pulsar::cgw_mc`

11.27.2.13 `double pulsar::clk_offsE[MAX_TEL_CLK_OFFS]`

11.27.2.14 `double pulsar::clk_offsT[MAX_TEL_CLK_OFFS]`

11.27.2.15 `double pulsar::clk_offsV[MAX_TEL_CLK_OFFS]`

11.27.2.16 `int pulsar::clkOffsN`

11.27.2.17 `char pulsar::clock[16]`

Clock standard to use as "UTC"

11.27.2.18 `char pulsar::clockFromOverride[64]`

Clock code to assume TOAs are measured against (e.g. UTC to turn off clock corrections, or TDB/TCG to turn off those + Einstein delay)

11.27.2.19 `enum constraint pulsar::constraints[MAX_PARAMS]`

Which constraints are specified

11.27.2.20 `int pulsar::correctTroposphere`

whether or not do correct for tropospheric delay

11.27.2.21 `double** pulsar::covar`

11.27.2.22 `char pulsar::decjStrPost[100]`

String containing RAJ and DECJ (postfit)

11.27.2.23 `char pulsar::decjStrPre[100]`

String containing RAJ and DECJ (prefit)

11.27.2.24 `double pulsar::decsim`

11.27.2.25 `char pulsar::deleteFileName[100]`

File name containing deleted points

11.27.2.26 `int pulsar::dilateFreq`

whether or not to apply SS time dilation to RFs

11.27.2.27 `double pulsar::dmoffsCM[MAX_IFUNC]`

11.27.2.28 `double pulsar::dmoffsCM_error[MAX_IFUNC]`

11.27.2.29 `double pulsar::dmoffsCM_mjd[MAX_IFUNC]`

11.27.2.30 `double pulsar::dmoffsCM_weight[MAX_IFUNC]`

11.27.2.31 `int pulsar::dmoffsCMnum`

11.27.2.32 `double pulsar::dmoffsDM[MAX_IFUNC]`

11.27.2.33 `double pulsar::dmoffsDM_error[MAX_IFUNC]`

11.27.2.34 `double pulsar::dmoffsDM_mjd[MAX_IFUNC]`

11.27.2.35 `double pulsar::dmoffsDM_weight[MAX_IFUNC]`

11.27.2.36 `int pulsar::dmoffsDMnum`

11.27.2.37 `double pulsar::dmOffset`

Value to add to DM flags

11.27.2.38 `int pulsar::eciCoord`

= 1 for ecliptic coords otherwise celestial coords

11.27.2.39 `char pulsar::eopc04_file[MAX_FILELEN]`

11.27.2.40 `char pulsar::ephemeris[MAX_FILELEN]`

11.27.2.41 `char pulsar::filterStr[MAX_STRLEN]`

String describing filters

11.27.2.42 `double pulsar::fitChisq`

Chisq value from the fit

11.27.2.43 `char pulsar::fitFunc[MAX_FILELEN]`

11.27.2.44 `FitInfo pulsar::fitinfo`

11.27.2.45 `int pulsar::fitJump[MAX_JUMPS]`

= 1 if fit for jump

11.27.2.46 `int pulsar::fitMode`

= 0 not fitting with errors, = 1 fitting with errors (MODE 1)

11.27.2.47 `int pulsar::fitNfree`

Number of degrees of freedom in fit

11.27.2.48 `int pulsar::fitParamGlobal[MAX_FIT]`

11.27.2.49 `int pulsar::fitParamGlobalK[MAX_FIT]`

11.27.2.50 `int pulsar::fitParamI[MAX_FIT]`

11.27.2.51 `int pulsar::fitParamK[MAX_FIT]`

11.27.2.52 `int pulsar::fixedFormat`

= 0 for separate .par and .tim files, > 0 indicates number of lines to skip

11.27.2.53 `char pulsar::fjumpID[16]`

11.27.2.54 `int pulsar::globalNfit`

Total number of parameters in the fit

11.27.2.55 `int pulsar::globalNoConstrain`

Total number of points without constraints

11.27.2.56 double pulsar::gwb_decj

11.27.2.57 double pulsar::gwb_epoch

11.27.2.58 double pulsar::gwb_geom_c

11.27.2.59 double pulsar::gwb_geom_p

11.27.2.60 double pulsar::gwb_raj

11.27.2.61 double pulsar::gwb_width

11.27.2.62 double pulsar::gwecc_dec

11.27.2.63 double pulsar::gwecc_distance

11.27.2.64 double pulsar::gwecc_e

11.27.2.65 double pulsar::gwecc_epoch

11.27.2.66 double pulsar::gwecc_inc

11.27.2.67 double pulsar::gwecc_m1

11.27.2.68 double pulsar::gwecc_m2

11.27.2.69 double pulsar::gwecc_nodes_orientation

11.27.2.70 double pulsar::gwecc_orbital_period

11.27.2.71 double pulsar::gwecc_psrdist

11.27.2.72 int pulsar::gwecc_pulsarTermOn

11.27.2.73 double pulsar::gwecc_ra

11.27.2.74 double pulsar::gwecc_redshift

11.27.2.75 double pulsar::gwecc_theta_0

11.27.2.76 double pulsar::gwecc_theta_nodes

11.27.2.77 double pulsar::gwm_decj

11.27.2.78 double pulsar::gwm_dphase

11.27.2.79 double pulsar::gwm_epoch

11.27.2.80 double pulsar::gwm_phi

11.27.2.81 double pulsar::gwm_raj

11.27.2.82 double pulsar::gwsrc_across_i

11.27.2.83 double pulsar::gwsrc_across_i_e

11.27.2.84 double pulsar::gwsrc_across_r

11.27.2.85 double pulsar::gwsrc_across_r_e

11.27.2.86 double pulsar::gwsrc_aplus_i

11.27.2.87 double pulsar::gwsrc_aplus_i_e

11.27.2.88 double pulsar::gwsrc_aplus_r

11.27.2.89 double pulsar::gwsrc_aplus_r_e

11.27.2.90 double pulsar::gwsrc_dec

11.27.2.91 double pulsar::gwsrc_epoch

11.27.2.92 double pulsar::gwsrc_psrdist

11.27.2.93 double pulsar::gwsrc_ra

11.27.2.94 double pulsar::ifunc_weights[MAX_IFUNC]

11.27.2.95 double pulsar::ifuncE[MAX_IFUNC]

11.27.2.96 int pulsar::ifuncN

11.27.2.97 double pulsar::ifuncT[MAX_IFUNC]

11.27.2.98 double pulsar::ifuncV[MAX_IFUNC]

11.27.2.99 int pulsar::ipm

= 1 if use interplanetary medium DM correction, = 0 otherwise

11.27.2.100 int pulsar::jboFormat

= 1 => JBO arrival time format and file structure (not byte swapping) = 2 => JBO format with byte swapping

11.27.2.101 char pulsar::JPL_EPHEMERIS[MAX_FILELEN]

11.27.2.102 char pulsar::jumpStr[MAX_JUMPS][MAX_STRLEN]

String describing jump

11.27.2.103 double pulsar::jumpVal[MAX_JUMPS]

Value of jump

11.27.2.104 double pulsar::jumpValErr[MAX_JUMPS]

Error on jump

11.27.2.105 `char pulsar::name[100]`

11.27.2.106 `int pulsar::nCompanion`

Number of binary companions

11.27.2.107 `int pulsar::nconstraints`

Number of fit constraints specified

11.27.2.108 `int pulsar::nDMEvents`

11.27.2.109 `int pulsar::ndmx`

Number of DM steps

11.27.2.110 `double pulsar::ne_sw`

Electron density at 1AU due to the solar wind

11.27.2.111 `int pulsar::nFit`

Number of points in the fit

11.27.2.112 `int pulsar::nGlobal`

Number of global parameters in the fit

11.27.2.113 `int pulsar::nits`

Number of iterations for the fit

11.27.2.114 `int pulsar::nJumps`

Number of jumps

11.27.2.115 `int pulsar::nobs`

Number of observations in .tim file

11.27.2.116 `int pulsar::noWarnings`

= 1, do not display warning messages

11.27.2.117 `int pulsar::nParam`

Number of parameters in the fit

11.27.2.118 int pulsar::nPhaseJump

Number of phase jumps

11.27.2.119 int pulsar::nQuad

11.27.2.120 int pulsar::nStorePrecision

11.27.2.121 int pulsar::nT2efac

11.27.2.122 int pulsar::nT2equad

11.27.2.123 int pulsar::nTelDX

11.27.2.124 int pulsar::nTelDY

11.27.2.125 int pulsar::nTelDZ

11.27.2.126 int pulsar::nTNBandNoise

11.27.2.127 int pulsar::nTNECORR

11.27.2.128 int pulsar::nTNEF

11.27.2.129 int pulsar::nTNEQ

11.27.2.130 int pulsar::nTNGroupNoise

11.27.2.131 int pulsar::nTNShapeletEvents

11.27.2.132 int pulsar::nTNSQ

11.27.2.133 int pulsar::nToffset

11.27.2.134 int pulsar::nWhite

11.27.2.135 int pulsar::nWhite_dm

11.27.2.136 **observation*** pulsar::obsn

[MAX_OBSN_VAL];

11.27.2.137 double pulsar::offset

Offset, always fitted for

11.27.2.138 double pulsar::offset_e

Error in the offset

11.27.2.139 int pulsar::outputTMatrix

11.27.2.140 **parameter** pulsar::param[MAX_PARAMS]

11.27.2.141 `char pulsar::passStr[MAX_STRLEN]`

String describing filters

11.27.2.142 `longdouble pulsar::phaseJump[MAX_JUMPS]`

Time of phase jump

11.27.2.143 `int pulsar::phaseJumpDir[MAX_JUMPS]`

Size and direction of phase jump

11.27.2.144 `int pulsar::phaseJumpID[MAX_JUMPS]`

ID of closest point to the phase jump

11.27.2.145 `int pulsar::planetShapiro`

= 1 if included otherwise 0

11.27.2.146 `double pulsar::posPulsar[3]`

3-vector pointing at pulsar

11.27.2.147 `double pulsar::quad_across_i[MAX_QUAD]`

11.27.2.148 `double pulsar::quad_across_i_e[MAX_QUAD]`

11.27.2.149 `double pulsar::quad_across_r[MAX_QUAD]`

11.27.2.150 `double pulsar::quad_across_r_e[MAX_QUAD]`

11.27.2.151 `double pulsar::quad_aplus_i[MAX_QUAD]`

11.27.2.152 `double pulsar::quad_aplus_i_e[MAX_QUAD]`

11.27.2.153 `double pulsar::quad_aplus_r[MAX_QUAD]`

11.27.2.154 `double pulsar::quad_aplus_r_e[MAX_QUAD]`

11.27.2.155 `double pulsar::quad_ifunc_c_DEC`

11.27.2.156 `double pulsar::quad_ifunc_c_RA`

11.27.2.157 `double pulsar::quad_ifunc_geom_c`

11.27.2.158 `double pulsar::quad_ifunc_geom_p`

11.27.2.159 `double pulsar::quad_ifunc_p_DEC`

11.27.2.160 `double pulsar::quad_ifunc_p_RA`

```

11.27.2.161 double pulsar::quad_ifuncE_c[MAX_IFUNC]
11.27.2.162 double pulsar::quad_ifuncE_p[MAX_IFUNC]
11.27.2.163 int pulsar::quad_ifuncN_c
11.27.2.164 int pulsar::quad_ifuncN_p
11.27.2.165 double pulsar::quad_ifuncT_c[MAX_IFUNC]
11.27.2.166 double pulsar::quad_ifuncT_p[MAX_IFUNC]
11.27.2.167 double pulsar::quad_ifuncV_c[MAX_IFUNC]
11.27.2.168 double pulsar::quad_ifuncV_p[MAX_IFUNC]
11.27.2.169 double pulsar::quadDEC
11.27.2.170 double pulsar::quadEpoch
11.27.2.171 double pulsar::quadRA
11.27.2.172 char pulsar::rajStrPost[100]
11.27.2.173 char pulsar::rajStrPre[100]
11.27.2.174 double pulsar::rasim
11.27.2.175 int pulsar::rescaleErrChisq

```

= 1 to rescale errors based on the reduced chisq, = 0 not to do this

```

11.27.2.176 double pulsar::rmsPost
11.27.2.177 double pulsar::rmsPre
11.27.2.178 char pulsar::robust
11.27.2.179 int pulsar::setTelVelX
11.27.2.180 int pulsar::setTelVelY
11.27.2.181 int pulsar::setTelVelZ
11.27.2.182 int pulsar::setUnits
11.27.2.183 int pulsar::simflag

```

Which fit function are we using

```

11.27.2.184 char pulsar::sorted

```

ToAs sorted Path for the file containing the corrections between observatory clocks and UTC(NIST) - set in [read↵](#)
[Parfile.C](#) char OBSERVATORY_CLOCK_2_UTC_NIST[MAX_FILELEN];

11.27.2.185 `storePrecision` pulsar::storePrec[MAX_STOREPRECISION]

11.27.2.186 `int` pulsar::swm

= 0 for basic tempo2 solar wind model, = 1 for XPY Solar wind model For whitening

11.27.2.187 `int` pulsar::t2cMethod

How to transform from terrestrial to celestial coords

11.27.2.188 `char` pulsar::T2efacFlagID[MAX_T2EFAC][MAX_FLAG_LEN]

11.27.2.189 `char` pulsar::T2efacFlagVal[MAX_T2EFAC][MAX_FLAG_LEN]

11.27.2.190 `double` pulsar::T2efacVal[MAX_T2EFAC]

11.27.2.191 `char` pulsar::T2equadFlagID[MAX_T2EQUAD][MAX_FLAG_LEN]

11.27.2.192 `char` pulsar::T2equadFlagVal[MAX_T2EQUAD][MAX_FLAG_LEN]

11.27.2.193 `double` pulsar::T2equadVal[MAX_T2EQUAD]

11.27.2.194 `double` pulsar::T2globalEfac

11.27.2.195 `double` pulsar::telDX_e[MAX_TEL_DX]

11.27.2.196 `double` pulsar::telDX_t[MAX_TEL_DX]

11.27.2.197 `double` pulsar::telDX_v[MAX_TEL_DX]

11.27.2.198 `double` pulsar::telDX_vel[MAX_TEL_DX]

11.27.2.199 `double` pulsar::telDX_vel_e[MAX_TEL_DX]

11.27.2.200 `double` pulsar::telDY_e[MAX_TEL_DY]

11.27.2.201 `double` pulsar::telDY_t[MAX_TEL_DY]

11.27.2.202 `double` pulsar::telDY_v[MAX_TEL_DY]

11.27.2.203 `double` pulsar::telDY_vel[MAX_TEL_DY]

11.27.2.204 `double` pulsar::telDY_vel_e[MAX_TEL_DY]

11.27.2.205 `double` pulsar::telDZ_e[MAX_TEL_DZ]

11.27.2.206 `double` pulsar::telDZ_t[MAX_TEL_DZ]

11.27.2.207 `double` pulsar::telDZ_v[MAX_TEL_DZ]

11.27.2.208 `double` pulsar::telDZ_vel[MAX_TEL_DZ]

11.27.2.209 `double` pulsar::telDZ_vel_e[MAX_TEL_DZ]

11.27.2.210 int pulsar::tempo1

= 1 if tempo1 is emulated

11.27.2.211 int pulsar::timeEphemeris

Which code to use for Einstein delay

11.27.2.212 double pulsar::TNBAndDMAmp

11.27.2.213 int pulsar::TNBAndDMC

11.27.2.214 double pulsar::TNBAndDMGam

11.27.2.215 double pulsar::TNBAndNoiseAmp[MAX_TNBN]

11.27.2.216 int pulsar::TNBAndNoiseC[MAX_TNBN]

11.27.2.217 double pulsar::TNBAndNoiseGam[MAX_TNBN]

11.27.2.218 double pulsar::TNBAndNoiseHF[MAX_TNBN]

11.27.2.219 double pulsar::TNBAndNoiseLF[MAX_TNBN]

11.27.2.220 double pulsar::TNDMAmp

11.27.2.221 int pulsar::TNDMC

11.27.2.222 double pulsar::TNDMCoeffs[200]

11.27.2.223 double pulsar::TNDMEvAmp[MAX_TNDMEv]

11.27.2.224 double pulsar::TNDMEvGam[MAX_TNDMEv]

11.27.2.225 double pulsar::TNDMEvLength[MAX_TNDMEv]

11.27.2.226 int pulsar::TNDMEvLin[MAX_TNDMEv]

11.27.2.227 int pulsar::TNDMEvOff[MAX_TNDMEv]

11.27.2.228 int pulsar::TNDMEvQuad[MAX_TNDMEv]

11.27.2.229 double pulsar::TNDMEvStart[MAX_TNDMEv]

11.27.2.230 double pulsar::TNDMGam

11.27.2.231 char pulsar::TNECORRFlagID[MAX_TNECORR][MAX_FLAG_LEN]

11.27.2.232 char pulsar::TNECORRFlagVal[MAX_TNECORR][MAX_FLAG_LEN]

11.27.2.233 double pulsar::TNECORRVal[MAX_TNECORR]

11.27.2.234 char pulsar::TNEFFlagID[MAX_TNEF][MAX_FLAG_LEN]

11.27.2.235 char pulsar::TNEFFlagVal[MAX_TNEF][MAX_FLAG_LEN]

11.27.2.236 double pulsar::TNEFVal[MAX_TNEF]

11.27.2.237 char pulsar::TNEQFlagID[MAX_TNEQ][MAX_FLAG_LEN]

11.27.2.238 char pulsar::TNEQFlagVal[MAX_TNEQ][MAX_FLAG_LEN]

11.27.2.239 double pulsar::TNEQVal[MAX_TNEQ]

11.27.2.240 double pulsar::TNGlobalEF

11.27.2.241 double pulsar::TNGlobalEQ

11.27.2.242 double pulsar::TNGroupNoiseAmp[MAX_TNGN]

11.27.2.243 int pulsar::TNGroupNoiseC[MAX_TNGN]

11.27.2.244 char pulsar::TNGroupNoiseFlagID[MAX_TNGN][MAX_FLAG_LEN]

11.27.2.245 char pulsar::TNGroupNoiseFlagVal[MAX_TNGN][MAX_FLAG_LEN]

11.27.2.246 double pulsar::TNGroupNoiseGam[MAX_TNGN]

11.27.2.247 double pulsar::TNRedAmp

11.27.2.248 int pulsar::TNRedC

11.27.2.249 double pulsar::TNRedCoeffs[200]

11.27.2.250 double pulsar::TNRedCorner

11.27.2.251 double pulsar::TNRedFlow

11.27.2.252 double pulsar::TNRedGam

11.27.2.253 double pulsar::TNShapeletEvFScale[MAX_TNDMEv]

11.27.2.254 int pulsar::TNShapeletEvN[MAX_TNDMEv]

11.27.2.255 double pulsar::TNShapeletEvPos[MAX_TNDMEv]

11.27.2.256 double pulsar::TNShapeletEvWidth[MAX_TNDMEv]

11.27.2.257 char pulsar::TNSQFlagID[MAX_TNSQ][MAX_FLAG_LEN]

11.27.2.258 char pulsar::TNSQFlagVal[MAX_TNSQ][MAX_FLAG_LEN]

11.27.2.259 double pulsar::TNSQVal[MAX_TNSQ]

11.27.2.260 int pulsar::TNsubtractDM

11.27.2.261 int pulsar::TNsubtractRed

11.27.2.262 double** pulsar::ToAextraCovar

11.27.2.263 double pulsar::tOffset[MAX_TOFFSET]

Offsets in TOAs in seconds

11.27.2.264 double pulsar::tOffset_f1[MAX_TOFFSET]

11.27.2.265 double pulsar::tOffset_f2[MAX_TOFFSET]

Range for offset to be applied

11.27.2.266 double pulsar::tOffset_t1[MAX_TOFFSET]

11.27.2.267 double pulsar::tOffset_t2[MAX_TOFFSET]

11.27.2.268 char pulsar::tOffsetFlags[MAX_TOFFSET][1000]

11.27.2.269 char pulsar::tOffsetSite[MAX_TOFFSET][100]

11.27.2.270 char pulsar::tzrsite[100]

Site-code for polyco

11.27.2.271 int pulsar::units

TDB or SI units (tempo emulation mode uses TDB) see #define definition above for possible units

11.27.2.272 int pulsar::useCalceph

11.27.2.273 int pulsar::useTNorth

11.27.2.274 double pulsar::velPulsar[3]

3-vector giving pulsar's velocity

11.27.2.275 double pulsar::wave_cos[MAX_WHITE]

11.27.2.276 double pulsar::wave_cos_dm[MAX_WHITE]

11.27.2.277 double pulsar::wave_cos_dm_err[MAX_WHITE]

11.27.2.278 double pulsar::wave_cos_err[MAX_WHITE]

11.27.2.279 double pulsar::wave_sine[MAX_WHITE]

11.27.2.280 double pulsar::wave_sine_dm[MAX_WHITE]

11.27.2.281 double pulsar::wave_sine_dm_err[MAX_WHITE]

11.27.2.282 double pulsar::wave_sine_err[MAX_WHITE]

11.27.2.283 double pulsar::waveScale

11.27.2.284 char pulsar::whiteNoiseModelFile[MAX_STRLEN]

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

- [tempo2.h](#)

11.28 sample Struct Reference

Public Attributes

- double [x](#)
- double [y](#)
- double [e](#)
- double [pred](#)
- int [actual](#)

11.28.1 Member Data Documentation

11.28.1.1 int sample::actual

11.28.1.2 double sample::e

11.28.1.3 double sample::pred

11.28.1.4 double sample::x

11.28.1.5 double sample::y

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

- [plugin/interpolate_plug.C](#)

11.29 storePrecision Struct Reference

```
#include <tempo2.h>
```

Public Attributes

- [longdouble](#) [minPrec](#)
- char [routine](#) [100]
- char [comment](#) [MAX_STRLEN]

11.29.1 Member Data Documentation

11.29.1.1 char storePrecision::comment[MAX_STRLEN]

11.29.1.2 [longdouble](#) storePrecision::minPrec

11.29.1.3 char storePrecision::routine[100]

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

- [tempo2.h](#)

11.30 T1Polyco Struct Reference

```
#include <tempo2pred.h>
```

Public Attributes

- char [psrname](#) [64]
- char [date_string](#) [10]
- char [utc_string](#) [13]
- long double [mjd_mid](#)
- double [dm](#)
- double [doppler](#)
- double [log10rms](#)
- long double [reference_phase](#)
- long double [frequency_psr_0](#)
- char [sitename](#) [5]
- int [span](#)
- int [ncoeff](#)
- double [frequency_obs](#)
- double [binary_phase](#)
- double [binary_frequency](#)
- long double [coeff](#) [32]

11.30.1 Member Data Documentation

11.30.1.1 double T1Polyco::binary_frequency

11.30.1.2 double T1Polyco::binary_phase

11.30.1.3 long double T1Polyco::coeff[32]

11.30.1.4 char T1Polyco::date_string[10]

11.30.1.5 double T1Polyco::dm

11.30.1.6 double T1Polyco::doppler

11.30.1.7 double T1Polyco::frequency_obs

11.30.1.8 long double T1Polyco::frequency_psr_0

11.30.1.9 double T1Polyco::log10rms

11.30.1.10 long double T1Polyco::mjd_mid

11.30.1.11 int T1Polyco::ncoeff

11.30.1.12 char T1Polyco::psrname[64]

11.30.1.13 long double T1Polyco::reference_phase

11.30.1.14 char T1Polyco::sitename[5]

11.30.1.15 int T1Polyco::span

11.30.1.16 char T1Polyco::utc_string[13]

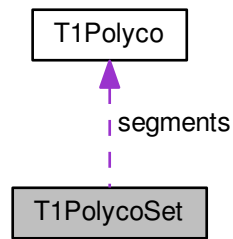
The documentation for this struct was generated from the following file:

- [tempo2pred.h](#)

11.31 T1PolycoSet Struct Reference

```
#include <tempo2pred.h>
```

Collaboration diagram for T1PolycoSet:



Public Attributes

- [T1Polyco * segments](#)
- int [nsegments](#)

11.31.1 Member Data Documentation

11.31.1.1 int T1PolycoSet::nsegments

11.31.1.2 **T1Polyco*** T1PolycoSet::segments

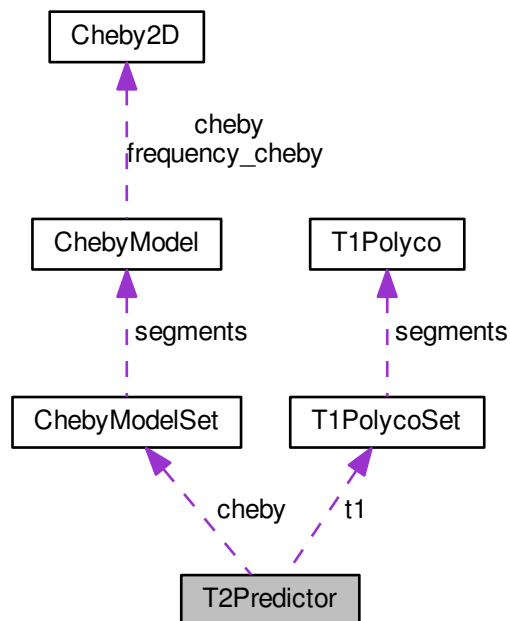
The documentation for this struct was generated from the following file:

- [tempo2pred.h](#)

11.32 T2Predictor Struct Reference

```
#include <tempo2pred.h>
```

Collaboration diagram for T2Predictor:



Public Attributes

- [T2PredictorKind](#) kind
- union {
 - [ChebyModelSet](#) cheby
 - [T1PolycoSet](#) t1
 } modelset

11.32.1 Member Data Documentation

11.32.1.1 **ChebyModelSet** T2Predictor::cheby

11.32.1.2 **T2PredictorKind** T2Predictor::kind

11.32.1.3 union { ... } T2Predictor::modelset

11.32.1.4 **T1PolycoSet** T2Predictor::t1

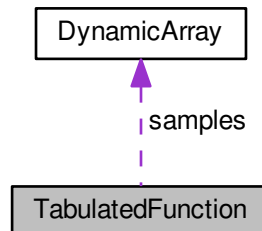
The documentation for this struct was generated from the following file:

- [tempo2pred.h](#)

11.33 TabulatedFunction Struct Reference

```
#include <tabulatedfunction.h>
```

Collaboration diagram for TabulatedFunction:



Public Attributes

- char [fileName](#) [256]
- char [header_line](#) [256]
- [DynamicArray](#) [samples](#)

11.33.1 Member Data Documentation

11.33.1.1 char `TabulatedFunction::fileName`[256]

11.33.1.2 char `TabulatedFunction::header_line`[256]

11.33.1.3 [DynamicArray](#) `TabulatedFunction::samples`

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

- [tabulatedfunction.h](#)

11.34 TabulatedFunctionSample Struct Reference

```
#include <tabulatedfunction.h>
```

Public Attributes

- double [x](#)
- double [y](#)

11.34.1 Member Data Documentation

11.34.1.1 double `TabulatedFunctionSample::x`

11.34.1.2 double TabulatedFunctionSample::y

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

- [tabulatedfunction.h](#)

11.35 XY Struct Reference

Public Attributes

- double [x](#)
- double [y](#)

11.35.1 Member Data Documentation

11.35.1.1 double XY::x

11.35.1.2 double XY::y

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

- [plugin/sigmaz_plug.C](#)

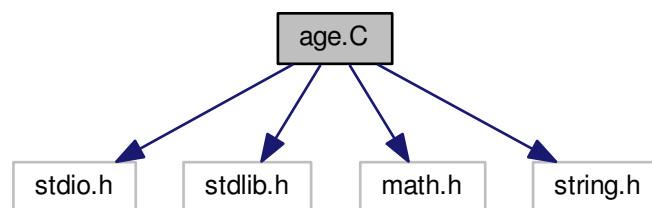
Chapter 12

File Documentation

12.1 age.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
```

Include dependency graph for age.C:



Functions

- void [rk4rms](#) (double *y*[], double *dydx*[], int *n*, double *x*, double *h*, double *yout*[], void(*[derivs](#))(double, double[], double[]))
- void [derivs](#) (double *x*, double *y*[], double *dydx*[])
- void [calc_age](#) (double *f0*, double *f1*, double *f2*, double **age*)

12.1.1 Function Documentation

12.1.1.1 void [calc_age](#) (double *f0*, double *f1*, double *f2*, double * *age*)

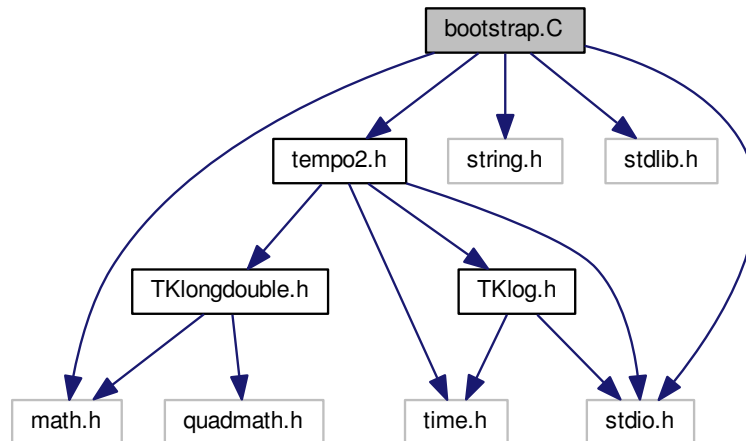
12.1.1.2 void [derivs](#) (double *x*, double *y*[], double *dydx*[])

12.1.1.3 void [rk4rms](#) (double *y*[], double *dydx*[], int *n*, double *x*, double *h*, double *yout*[], void(*) (double, double[], double[]) *derivs*)

12.2 bootstrap.C File Reference

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include "tempo2.h"
```

Include dependency graph for bootstrap.C:



Macros

- #define [MAX_ITER](#) 4096

Functions

- double [random](#) (long *idum)
- int [bootstrap](#) (pulsar *psr, int p, int npsr)

12.2.1 Macro Definition Documentation

12.2.1.1 #define MAX_ITER 4096

12.2.2 Function Documentation

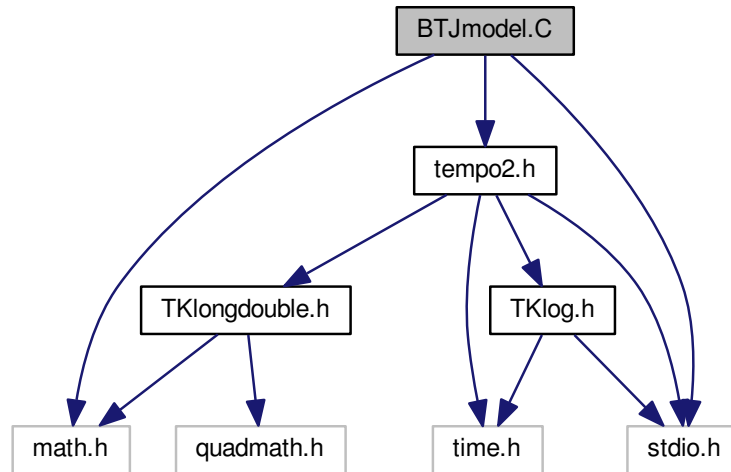
12.2.2.1 int bootstrap (pulsar * psr, int p, int npsr)

12.2.2.2 double random (long * idum)

12.3 BTJmodel.C File Reference

```
#include <stdio.h>
#include <math.h>
#include "tempo2.h"
```


Include dependency graph for BTJmodel.C:



Functions

- double [BTJmodel](#) ([pulsar](#) *psr, int p, int ipos, int param, int arr)
- void [updateBTJ](#) ([pulsar](#) *psr, double val, double err, int pos, int arr)

12.3.1 Function Documentation

12.3.1.1 double [BTJmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*, int *arr*)

12.3.1.2 void [updateBTJ](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*, int *arr*)

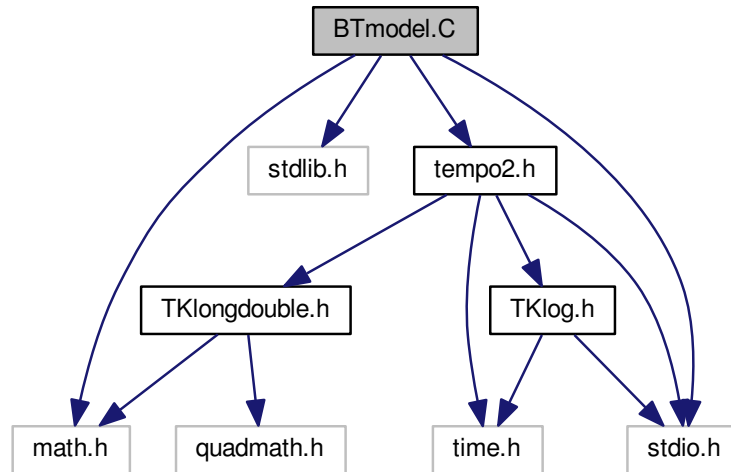
12.4 BTmodel.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"

```

Include dependency graph for BTmodel.C:



Functions

- double [BTmodel](#) ([pulsar](#) *psr, int p, int ipos, int param)
- void [updateBT](#) ([pulsar](#) *psr, double val, double err, int pos)

12.4.1 Function Documentation

12.4.1.1 double [BTmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*)

12.4.1.2 void [updateBT](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*)

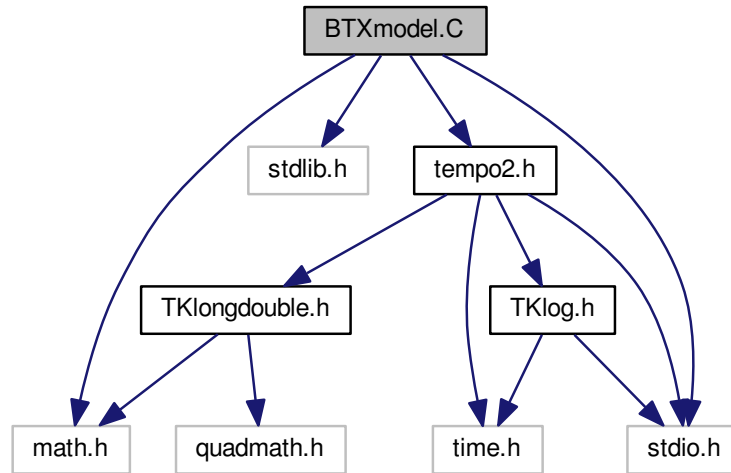
12.5 BTXmodel.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"

```

Include dependency graph for BTXmodel.C:



Functions

- double [BTXmodel](#) ([pulsar](#) *psr, int p, int ipos, int param, int k)
- void [updateBTX](#) ([pulsar](#) *psr, double val, double err, int pos, int k)

12.5.1 Function Documentation

12.5.1.1 double [BTXmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*, int *k*)

12.5.1.2 void [updateBTX](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*, int *k*)

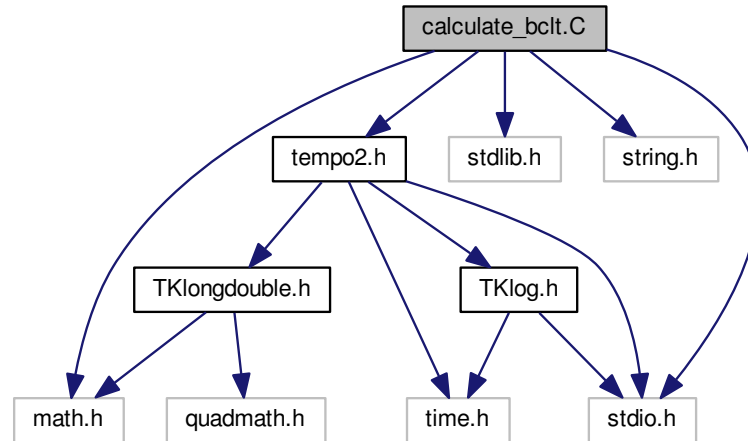
12.6 calculate_bclt.C File Reference

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"

```

Include dependency graph for calculate_bclt.C:



Functions

- void `calculate_bclt` (`pulsar` *psr, int npsr)

12.6.1 Function Documentation

12.6.1.1 void `calculate_bclt` (`pulsar` *psr, int npsr)

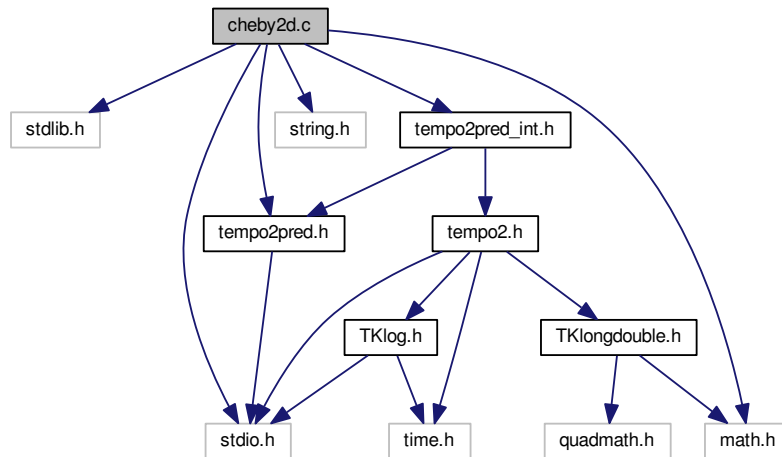
12.7 cheby2d.c File Reference

```

#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include "tempo2pred.h"
#include "tempo2pred_int.h"

```

Include dependency graph for cheby2d.c:



Macros

- `#define M_PII 3.14159265358979323846264338327950288L`

Functions

- void [Cheby2D_Init](#) ([Cheby2D](#) *cheby, int nx, int ny)
- void [Cheby2D_Destroy](#) ([Cheby2D](#) *cheby)
- void [Cheby2D_Copy](#) ([Cheby2D](#) *cheby, const [Cheby2D](#) *from)
- void [Cheby2D_Construct](#) ([Cheby2D](#) *cheby, void(*func)(long double *x, long double *y, int nx, int ny, long double *z, void *info), void *info)
- void [Cheby2D_Construct_x_Derivative](#) ([Cheby2D](#) *dcheby, const [Cheby2D](#) *cheby)
- long double [Cheby2D_Evaluate](#) (const [Cheby2D](#) *cheby, long double x, long double y)
- void [Cheby2D_Test](#) ([Cheby2D](#) *cheby, int nx_test, int ny_test, void(*func)(long double *x, long double *y, int nx, int ny, long double *z, void *info), void *info, long double *residualRMS, long double *residualMAV)
- void [testFunc](#) (long double *x, long double *y, int nx, int ny, long double *z, void *info)
- void [testCheby2D](#) ()
- void [ChebyModel_Init](#) ([ChebyModel](#) *cm, int nmjdccoeff, int nfreqcoeff)
- void [ChebyModel_Copy](#) ([ChebyModel](#) *cm, [ChebyModel](#) *from)
- void [ChebyModel_Destroy](#) ([ChebyModel](#) *cm)
- long double [ChebyModel_GetPhase](#) (const [ChebyModel](#) *cm, long double mjd, long double freq)
- long double [ChebyModel_GetFrequency](#) (const [ChebyModel](#) *cm, long double mjd, long double freq)
- void [ChebyModel_Write](#) (const [ChebyModel](#) *cm, FILE *f)
- int [ChebyModel_Read](#) ([ChebyModel](#) *cm, FILE *f)
- int [ChebyModelSet_GetNearestIndex](#) (const [ChebyModelSet](#) *cms, long double mjd)
- [ChebyModel](#) * [ChebyModelSet_GetNearest](#) (const [ChebyModelSet](#) *cms, long double mjd)
- long double [ChebyModelSet_GetPhase](#) (const [ChebyModelSet](#) *cms, long double mjd, long double freq)
- long double [ChebyModelSet_GetFrequency](#) (const [ChebyModelSet](#) *cms, long double mjd, long double freq)
- void [ChebyModelSet_Write](#) (const [ChebyModelSet](#) *cms, FILE *f)
- int [ChebyModelSet_Read](#) ([ChebyModelSet](#) *cms, FILE *f)
- void [ChebyModelSet_Init](#) ([ChebyModelSet](#) *cms)
- int [ChebyModelSet_Insert](#) ([ChebyModelSet](#) *cms, const [ChebyModelSet](#) *from)
- void [ChebyModelSet_Keep](#) ([ChebyModelSet](#) *cms, unsigned nmjd, const long double *mjd)
- void [ChebyModelSet_Destroy](#) ([ChebyModelSet](#) *cms)

Variables

- int [ChebyModelSet_OutOfRange](#) = 0

12.7.1 Macro Definition Documentation

12.7.1.1 `#define M_PII 3.14159265358979323846264338327950288L`

12.7.2 Function Documentation

12.7.2.1 `void Cheby2D_Construct (Cheby2D * cheby, void(*)(long double *x, long double *y, int nx, int ny, long double *z, void *info) func, void * info)`

12.7.2.2 `void Cheby2D_Construct_x_Derivative (Cheby2D * dcheby, const Cheby2D * cheby)`

12.7.2.3 `void Cheby2D_Copy (Cheby2D * cheby, const Cheby2D * from)`

12.7.2.4 `void Cheby2D_Destroy (Cheby2D * cheby)`

12.7.2.5 `long double Cheby2D_Evaluate (const Cheby2D * cheby, long double x, long double y)`

12.7.2.6 `void Cheby2D_Init (Cheby2D * cheby, int nx, int ny)`

12.7.2.7 `void Cheby2D_Test (Cheby2D * cheby, int nx_test, int ny_test, void(*)(long double *x, long double *y, int nx, int ny, long double *z, void *info) func, void * info, long double * residualRMS, long double * residualMAV)`

12.7.2.8 `void ChebyModel_Copy (ChebyModel * cm, ChebyModel * from)`

12.7.2.9 `void ChebyModel_Destroy (ChebyModel * cm)`

12.7.2.10 `long double ChebyModel_GetFrequency (const ChebyModel * cm, long double mjd, long double freq)`

12.7.2.11 `long double ChebyModel_GetPhase (const ChebyModel * cm, long double mjd, long double freq)`

12.7.2.12 `void ChebyModel_Init (ChebyModel * cm, int nmjdcoeff, int nfreqcoeff)`

12.7.2.13 `int ChebyModel_Read (ChebyModel * cm, FILE * f)`

12.7.2.14 `void ChebyModel_Write (const ChebyModel * cm, FILE * f)`

12.7.2.15 `void ChebyModelSet_Destroy (ChebyModelSet * cms)`

12.7.2.16 `long double ChebyModelSet_GetFrequency (const ChebyModelSet * cms, long double mjd, long double freq)`

12.7.2.17 `ChebyModel* ChebyModelSet_GetNearest (const ChebyModelSet * cms, long double mjd)`

12.7.2.18 `int ChebyModelSet_GetNearestIndex (const ChebyModelSet * cms, long double mjd)`

12.7.2.19 `long double ChebyModelSet_GetPhase (const ChebyModelSet * cms, long double mjd, long double freq)`

12.7.2.20 `void ChebyModelSet_Init (ChebyModelSet * cms)`

12.7.2.21 `int ChebyModelSet_Insert (ChebyModelSet * cms, const ChebyModelSet * from)`

12.7.2.22 `void ChebyModelSet_Keep (ChebyModelSet * cms, unsigned nmjd, const long double * mjd)`

12.7.2.23 int ChebyModelSet_Read (ChebyModelSet * cms, FILE * f)

12.7.2.24 void ChebyModelSet_Write (const ChebyModelSet * cms, FILE * f)

12.7.2.25 void testCheby2D ()

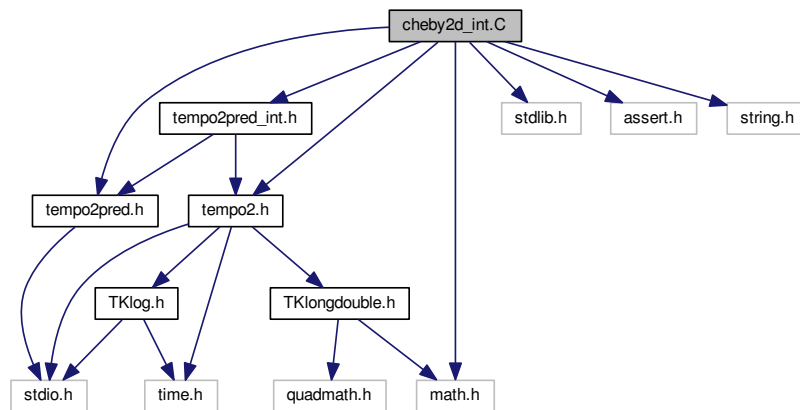
12.7.2.26 void testFunc (long double * x, long double * y, int nx, int ny, long double * z, void * info)

12.7.3 Variable Documentation

12.7.3.1 int ChebyModelSet_OutOfRange = 0

12.8 cheby2d_int.C File Reference

```
#include "tempo2pred.h"
#include "tempo2pred_int.h"
#include "tempo2.h"
#include <stdlib.h>
#include <math.h>
#include <assert.h>
#include <string.h>
Include dependency graph for cheby2d_int.C:
```



Classes

- struct [ChebyModelInfo](#)

Functions

- void [chebyModelFunc](#) (long double *x, long double *y, int nx, int ny, long double *z, void *info_in)
- void [ChebyModel_Construct](#) (ChebyModel *cm, const [pulsar](#) *psr)
- void [ChebyModel_Test](#) (ChebyModel *cm, const [pulsar](#) *psr, int nmjd, int nfreq, long double *residualRMS, long double *residualMAV)
- void [ChebyModelSet_Construct](#) (ChebyModelSet *cms, const [pulsar](#) *psr, const char *sitename, long double mjd_start, long double mjd_end, long double segment_length, long double overlap, long double freq_start, long double freq_end, int ntimecoeff, int nfreqcoeff)

- void [ChebyModelSet_Test](#) ([ChebyModelSet](#) *cms, const [pulsar](#) *psr, int nmjd, int nfreq, long double *residualRMS, long double *residualMAV)

12.8.1 Function Documentation

12.8.1.1 void [ChebyModel_Construct](#) ([ChebyModel](#) * cm, const [pulsar](#) * psr)

12.8.1.2 void [ChebyModel_Test](#) ([ChebyModel](#) * cm, const [pulsar](#) * psr, int nmjd, int nfreq, long double * residualRMS, long double * residualMAV)

12.8.1.3 void [chebyModelFunc](#) (long double * x, long double * y, int nx, int ny, long double * z, void * info_in)

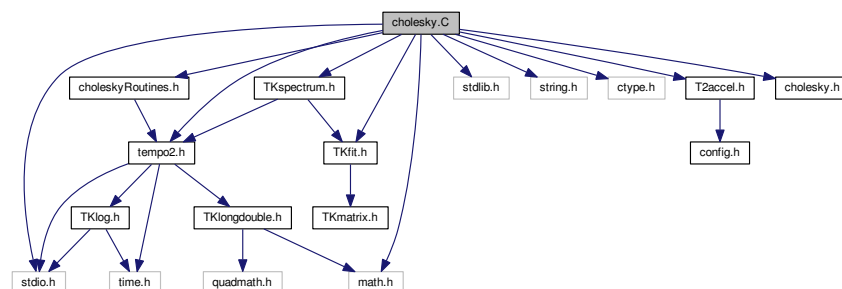
12.8.1.4 void [ChebyModelSet_Construct](#) ([ChebyModelSet](#) * cms, const [pulsar](#) * psr, const char * sitename, long double mjd_start, long double mjd_end, long double segment_length, long double overlap, long double freq_start, long double freq_end, int ntimecoeff, int nfreqcoeff)

12.8.1.5 void [ChebyModelSet_Test](#) ([ChebyModelSet](#) * cms, const [pulsar](#) * psr, int nmjd, int nfreq, long double * residualRMS, long double * residualMAV)

12.9 cholesky.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#include "tempo2.h"
#include "TKfit.h"
#include "TKspectrum.h"
#include "choleskyRoutines.h"
#include "T2accel.h"
#include "cholesky.h"
```

Include dependency graph for cholesky.C:



Macros

- #define [LINE_LENGTH](#) 2048

Functions

- void [getCholeskyMatrix](#) (double **uinv, const char *fname, [pulsar](#) *psr, double *resx, double *resy, double *rese, int np, int nc, int *ip)
- void [cholesky_readT2CholModel_R](#) (double **m, double **mm, const char *fname, double *resx, double *resy, double *rese, int np, int nc, int *ip, [pulsar](#) *psr, char *_psrJ, double _mjd_start, double _mjd_end, int recursion)
- void [cholesky_readFromCovarianceFunction](#) (double **m, const char *fname, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_covarFunc2matrix](#) (double **m, double *covarFunc, int ndays, double *resx, double *resy, double *rese, int np, int nc)
- void [getCholeskyDiagonals](#) (double **uinv, [pulsar](#) *psr, double *resx, double *resy, double *rese, int np, int nc, int *ip)
- int [cholesky_formUinv](#) (double **uinv, double **m, int np)
- void [cholesky_readT2Model1](#) (double **m, FILE *file, double *resx, double *resy, double *rese, int np, int nc, int *ip, [pulsar](#) *psr)
- void [cholesky_readT2Model2](#) (double **m, FILE *file, double *resx, double *resy, double *rese, int np, int nc, int *ip, [pulsar](#) *psr)
- void [cholesky_ecm](#) (double **m, char *fileName, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_readT2CholModel](#) (double **m, const char *fname, double *resx, double *resy, double *rese, int np, int nc, int *ip, [pulsar](#) *psr)
- void [addCovar](#) (double **m, double **mm, double *resx, double *resy, double *rese, int np, int nc, int *ip, [pulsar](#) *psr, double mjd_start, double mjd_end)
- void [cholesky_dmModelCovarParam](#) (double **m, double [alpha](#), double a, double b, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_dmModel](#) (double **m, double D_d, double d, double ref_freq, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_powerlawModel](#) (double **m, double modelAlpha, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_powerlawModel_withBeta](#) (double **m, double modelAlpha, double modelBeta, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)

12.9.1 Macro Definition Documentation

12.9.1.1 `#define LINE_LENGTH 2048`

12.9.2 Function Documentation

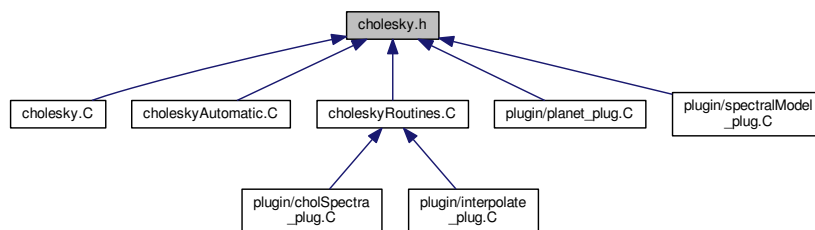
- 12.9.2.1 void `addCovar (double ** m, double ** mm, double * resx, double * resy, double * rese, int np, int nc, int * ip, pulsar * psr, double mjd_start, double mjd_end)`
- 12.9.2.2 void `cholesky_covarFunc2matrix (double ** m, double * covarFunc, int ndays, double * resx, double * resy, double * rese, int np, int nc)`
- 12.9.2.3 void `cholesky_dmModel (double ** m, double D_d, double d, double ref_freq, double * resx, double * resy, double * rese, int np, int nc)`
- 12.9.2.4 void `cholesky_dmModelCovarParam (double ** m, double alpha, double a, double b, double * resx, double * resy, double * rese, int np, int nc)`
- 12.9.2.5 void `cholesky_ecm (double ** m, char * fileName, double * resx, double * resy, double * rese, int np, int nc)`
- 12.9.2.6 int `cholesky_formUinv (double ** uinv, double ** m, int np)`

UINV is a lower triangular matrix. Matrices are row-major order, i.e. `uinv[r][c]`. returns 0 if ok.

- 12.9.2.7 void `cholesky_powerlawModel` (double ** *m*, double *modelAlpha*, double *modelFc*, double *modelA*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.9.2.8 void `cholesky_powerlawModel_withBeta` (double ** *m*, double *modelAlpha*, double *modelBeta*, double *modelFc*, double *modelA*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.9.2.9 void `cholesky_readFromCovarianceFunction` (double ** *m*, const char * *fname*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.9.2.10 void `cholesky_readT2CholModel` (double ** *m*, const char * *fname*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*, pulsar * *psr*)
- 12.9.2.11 void `cholesky_readT2CholModel_R` (double ** *m*, double ** *mm*, const char * *fname*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*, pulsar * *psr*, char * *_psrJ*, double *_mjd_start*, double *_mjd_end*, int *recursion*)
- 12.9.2.12 void `cholesky_readT2Model1` (double ** *m*, FILE * *file*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*, pulsar * *psr*)
- 12.9.2.13 void `cholesky_readT2Model2` (double ** *m*, FILE * *file*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*, pulsar * *psr*)
- 12.9.2.14 void `getCholeskyDiagonals` (double ** *uinv*, pulsar * *psr*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*)
- 12.9.2.15 void `getCholeskyMatrix` (double ** *uinv*, const char * *fname*, pulsar * *psr*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*)

12.10 cholesky.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [cholesky_readFromCovarianceFunction](#) (double ***m*, const char **fname*, double **resx*, double **resy*, double **rese*, int *np*, int *nc*)
- void [cholesky_covarFunc2matrix](#) (double ***m*, double **covarFunc*, int *ndays*, double **resx*, double **resy*, double **rese*, int *np*, int *nc*)
- void [cholesky_powerlawModel](#) (double ***m*, double *modelAlpha*, double *modelFc*, double *modelA*, double **resx*, double **resy*, double **rese*, int *np*, int *nc*)
- void [cholesky_powerlawModel_withBeta](#) (double ***m*, double *modelAlpha*, double *beta*, double *modelFc*, double *modelA*, double **resx*, double **resy*, double **rese*, int *np*, int *nc*)
- int [cholesky_formUinv](#) (double ***uinv*, double ***m*, int *np*)

- void `cholesky_dmModel` (double **m, double D, double d, double ref_freq, double *resx, double *resy, double *rese, int np, int nc)
- void `cholesky_ecm` (double **m, char *fileName, double *resx, double *resy, double *rese, int np, int nc)
- void `cholesky_dmModelCovarParam` (double **m, double alpha, double a, double b, double *resx, double *resy, double *rese, int np, int nc)

12.10.1 Function Documentation

- 12.10.1.1 void `cholesky_covarFunc2matrix` (double ** m, double * covarFunc, int ndays, double * resx, double * resy, double * rese, int np, int nc)
- 12.10.1.2 void `cholesky_dmModel` (double ** m, double D, double d, double ref_freq, double * resx, double * resy, double * rese, int np, int nc)
- 12.10.1.3 void `cholesky_dmModelCovarParam` (double ** m, double alpha, double a, double b, double * resx, double * resy, double * rese, int np, int nc)
- 12.10.1.4 void `cholesky_ecm` (double ** m, char * fileName, double * resx, double * resy, double * rese, int np, int nc)
- 12.10.1.5 int `cholesky_formUinv` (double ** uinv, double ** m, int np)

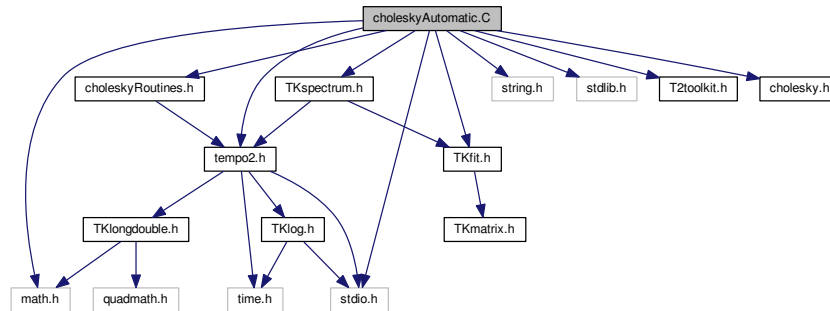
UINV is a lower triangular matrix. Matrices are row-major order, i.e. `uinv[r][c]`. returns 0 if ok.

- 12.10.1.6 void `cholesky_powerlawModel` (double ** m, double modelAlpha, double modelFc, double modelA, double * resx, double * resy, double * rese, int np, int nc)
- 12.10.1.7 void `cholesky_powerlawModel_withBeta` (double ** m, double modelAlpha, double beta, double modelFc, double modelA, double * resx, double * resy, double * rese, int np, int nc)
- 12.10.1.8 void `cholesky_readFromCovarianceFunction` (double ** m, const char * fname, double * resx, double * resy, double * rese, int np, int nc)

12.11 choleskyAutomatic.C File Reference

```
#include "choleskyRoutines.h"
#include "tempo2.h"
#include "math.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "TKfit.h"
#include "TKspectrum.h"
#include "T2toolkit.h"
#include "cholesky.h"
```

Include dependency graph for choleskyAutomatic.C:



Functions

- void **T2get_covFunc_automatic** (pulsar *psr, double expSmooth, char *outname, double *fc_w, double *fc_r, double *modelAlpha_out, double *modelVal, double *whiteNoiseLevel, int reallflag, int dcmflag)

12.11.1 Function Documentation

12.11.1.1 void T2get_covFunc_automatic (pulsar * psr, double expSmooth, char * outname, double * fc_w, double * fc_r, double * modelAlpha_out, double * modelVal, double * whiteNoiseLevel, int reallflag, int dcmflag)

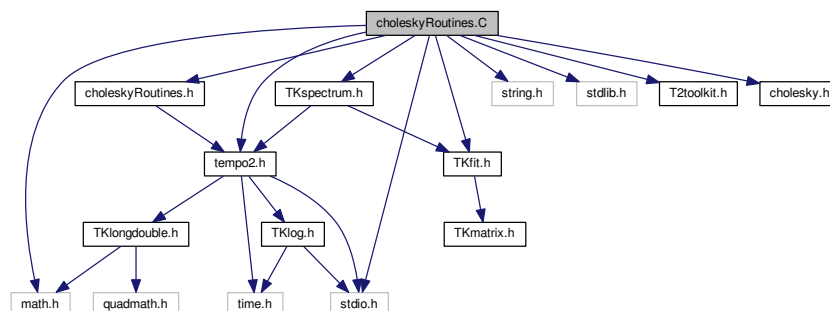
12.12 choleskyRoutines.C File Reference

```

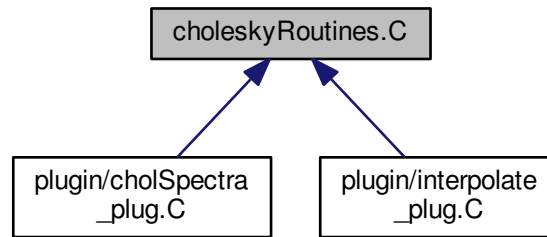
#include "choleskyRoutines.h"
#include "tempo2.h"
#include "math.h"
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include "TKfit.h"
#include "TKspectrum.h"
#include "T2toolkit.h"
#include "cholesky.h"

```

Include dependency graph for choleskyRoutines.C:



This graph shows which files directly or indirectly include this file:



Functions

- `int T2guess_vals` (double *x, double *y, int n, double *alpha, double *amp, double *fc, int *nfit, double wn, double *fc_white, int prewhite)
- `void T2getWhiteNoiseLevel` (int n, double *y, int nlast, double *av)
- `void T2getWhiteRes` (double *resx, double *resy, double *rese, int nres, double **uinv, double *cholWhiteY)
- `void T2cubicFit` (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *cubicErr)
- `void T2findSmoothCurve` (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *smoothModel, double expSmooth)
- `void T2interpolate` (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *interpX, double *interpY, int *nInterp, int interpTime, double expSmooth)
- `void T2getHighFreqRes` (double *resy, double *smoothModel, int nres, double *highFreqRes)
- `int T2calculateSpectra` (double *x, double *y, double *e, int n, int useErr, int preWhite, int specType, double *specX, double *specY)
- `int T2fitSpectra` (double *preWhiteSpecX, double *preWhiteSpecY, int nPreWhiteSpec, double *modelAlpha, double *modelFc, int *modelNfit, double *modelScale, double *fitVar, int aval, int ipw, double ifc, double iexp, int inpt, double amp, int useBeta, double *betaVal, double cutoff)
- `int T2calculateCovarFunc` (double modelAlpha, double modelFc, double modelA, int useBeta, double betaVal, double *covFunc, double *resx, double *resy, double *rese, int np)
- `void T2calculateCholesky` (double modelAlpha, double modelFc, double modelA, double fitVar, double **uinv, double *covarFunc, double *resx, double *resy, double *rese, int np, double *highFreqRes, double *errorScaleFactor, int dcmflag, int useBeta, double betaVal)
- `void T2calculateDailyCovariance` (double *x, double *y, double *e, int n, double *cv, int *in, double *zl, int usewt)
- `int T2obtainTimingResiduals` (pulsar *psr, double *resx, double *resy, double *rese)
- `void T2writeCovarFuncModel` (double alpha, double fc, double val, double white, char *fname)
- `void T2cholDecomposition` (double **a, int n, double *p)

12.12.1 Function Documentation

12.12.1.1 `void T2calculateCholesky` (double *modelAlpha*, double *modelFc*, double *modelA*, double *fitVar*, double ** *uinv*, double * *covarFunc*, double * *resx*, double * *resy*, double * *rese*, int *np*, double * *highFreqRes*, double * *errorScaleFactor*, int *dcmflag*, int *useBeta*, double *betaVal*)

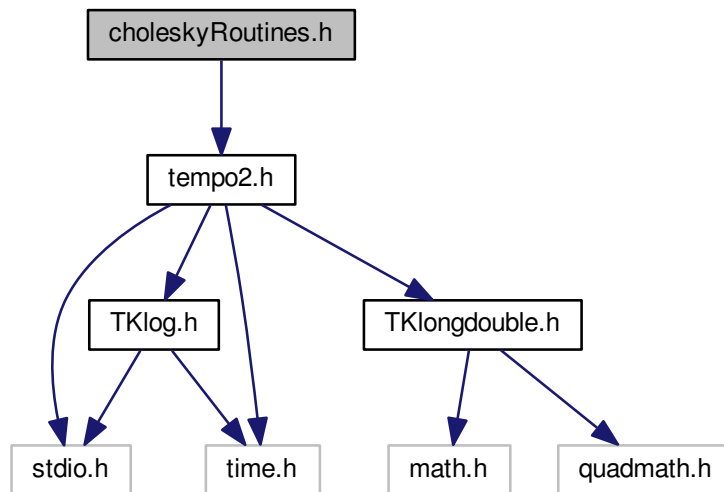
12.12.1.2 `int T2calculateCovarFunc` (double *modelAlpha*, double *modelFc*, double *modelA*, int *useBeta*, double *betaVal*, double * *covFunc*, double * *resx*, double * *resy*, double * *rese*, int *np*)

- 12.12.1.3 void T2calculateDailyCovariance (double * *x*, double * *y*, double * *e*, int *n*, double * *cv*, int * *in*, double * *zl*, int *usewt*)
- 12.12.1.4 int T2calculateSpectra (double * *x*, double * *y*, double * *e*, int *n*, int *useErr*, int *preWhite*, int *specType*, double * *specX*, double * *specY*)
- 12.12.1.5 void T2cholDecomposition (double ** *a*, int *n*, double * *p*)
- 12.12.1.6 void T2cubicFit (double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *cubicVal*, double * *cubicErr*)
- 12.12.1.7 void T2findSmoothCurve (double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *cubicVal*, double * *smoothModel*, double *expSmooth*)
- 12.12.1.8 int T2fitSpectra (double * *preWhiteSpecX*, double * *preWhiteSpecY*, int *nPreWhiteSpec*, double * *modelAlpha*, double * *modelFc*, int * *modelNfit*, double * *modelScale*, double * *fitVar*, int *aval*, int *ipw*, double *ifc*, double *iexp*, int *inpt*, double *amp*, int *useBeta*, double * *betaVal*, double *cutoff*)
- 12.12.1.9 void T2getHighFreqRes (double * *resy*, double * *smoothModel*, int *nres*, double * *highFreqRes*)
- 12.12.1.10 void T2getWhiteNoiseLevel (int *n*, double * *y*, int *nlast*, double * *av*)
- 12.12.1.11 void T2getWhiteRes (double * *resx*, double * *resy*, double * *rese*, int *nres*, double ** *uinv*, double * *cholWhiteY*)
- 12.12.1.12 int T2guess_vals (double * *x*, double * *y*, int *n*, double * *alpha*, double * *amp*, double * *fc*, int * *nfit*, double *wn*, double * *fc_white*, int *prewhite*)
- 12.12.1.13 void T2interpolate (double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *cubicVal*, double * *interpX*, double * *interpY*, int * *nInterp*, int *interpTime*, double *expSmooth*)
- 12.12.1.14 int T2obtainTimingResiduals (pulsar * *psr*, double * *resx*, double * *resy*, double * *rese*)
- 12.12.1.15 void T2writeCovarFuncModel (double *alpha*, double *fc*, double *val*, double *white*, char * *fname*)

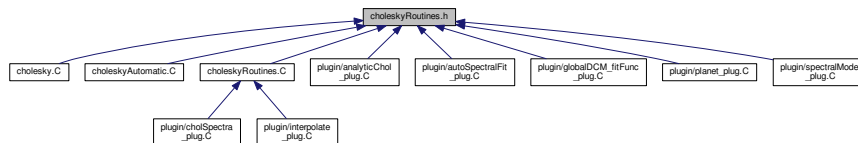
12.13 choleskyRoutines.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for choleskyRoutines.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [T2writeCovarFuncModel](#) (double [alpha](#), double fc, double val, double white, char *fname)
- void [T2get_covFunc_automatic](#) (pulsar *psr, double expSmooth, char *outname, double *fc_w, double *fc←_r, double *modelAlpha_out, double *modelVal, double *whiteNoiseLevel, int realflag, int dcmflag)
- void [T2cubicFit](#) (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *cubicErr)
- void [T2findSmoothCurve](#) (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *smoothModel, double expSmooth)
- void [T2interpolate](#) (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *interpX, double *interpY, int *nInterp, int interpTime, double expSmooth)
- void [T2getHighFreqRes](#) (double *resy, double *smoothModel, int nres, double *highFreqRes)
- int [T2calculateSpectra](#) (double *x, double *y, double *e, int n, int useErr, int preWhite, int specType, double *specX, double *specY)
- int [T2fitSpectra](#) (double *preWhiteSpecX, double *preWhiteSpecY, int nPreWhiteSpec, double *modelAlpha, double *modelFc, int *modelNfit, double *modelScale, double *fitVar, int aval, int ipw, double ifc, double iexp, int inpt, double amp, int useBeta, double *betaVal, double cutoff=0.0)
- void [T2calculateCholesky](#) (double modelAlpha, double modelFc, double modelScale, double fitVar, double **uin, double *covFunc, double *resx, double *resy, double *rese, int np, double *highFreqRes, double *errorScaleFactor, int dcmflag, int useBeta, double betaVal)

- int [T2calculateCovarFunc](#) (double modelAlpha, double modelFc, double modelA, int useBeta, double betaVal, double *covFunc, double *resx, double *resy, double *rese, int np)
- void [T2getWhiteRes](#) (double *resx, double *resy, double *rese, int nres, double **uin, double *cholWhiteY)
- void [T2calculateDailyCovariance](#) (double *x, double *y, double *e, int n, double *cv, int *in, double *zl, int usewt)
- int [T2obtainTimingResiduals](#) (pulsar *psr, double *resx, double *resy, double *rese)
- int [T2guess_vals](#) (double *x, double *y, int n, double *alpha, double *amp, double *fc, int *nfit, double wn, double *fc_white, int prewhite)
- void [T2getWhiteNoiseLevel](#) (int n, double *y, int nlast, double *av)
- void [T2cholDecomposition](#) (double **a, int n, double *p)

Variables

- double [FCALPHA](#)
- double [WNLEVEL](#)
- double [EXPSMOOTH](#)
- double [UPW](#)
- double [NFIT](#)
- double [FCFINAL](#)

12.13.1 Function Documentation

- 12.13.1.1 void [T2calculateCholesky](#) (double *modelAlpha*, double *modelFc*, double *modelScale*, double *fitVar*, double ***uin*, double * *covFunc*, double * *resx*, double * *resy*, double * *rese*, int *np*, double * *highFreqRes*, double * *errorScaleFactor*, int *dcmflag*, int *useBeta*, double *betaVal*)
- 12.13.1.2 int [T2calculateCovarFunc](#) (double *modelAlpha*, double *modelFc*, double *modelA*, int *useBeta*, double *betaVal*, double * *covFunc*, double * *resx*, double * *resy*, double * *rese*, int *np*)
- 12.13.1.3 void [T2calculateDailyCovariance](#) (double * *x*, double * *y*, double * *e*, int *n*, double * *cv*, int * *in*, double * *zl*, int *usewt*)
- 12.13.1.4 int [T2calculateSpectra](#) (double * *x*, double * *y*, double * *e*, int *n*, int *useErr*, int *preWhite*, int *specType*, double * *specX*, double * *specY*)
- 12.13.1.5 void [T2cholDecomposition](#) (double ** *a*, int *n*, double * *p*)
- 12.13.1.6 void [T2cubicFit](#) (double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *cubicVal*, double * *cubicErr*)
- 12.13.1.7 void [T2findSmoothCurve](#) (double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *cubicVal*, double * *smoothModel*, double *expSmooth*)
- 12.13.1.8 int [T2fitSpectra](#) (double * *preWhiteSpecX*, double * *preWhiteSpecY*, int *nPreWhiteSpec*, double * *modelAlpha*, double * *modelFc*, int * *modelNfit*, double * *modelScale*, double * *fitVar*, int *aval*, int *ipw*, double *ifc*, double *iexp*, int *inpt*, double *amp*, int *useBeta*, double * *betaVal*, double *cutoff* = 0.0)
- 12.13.1.9 void [T2get_covFunc_automatic](#) (pulsar * *psr*, double *expSmooth*, char * *outname*, double * *fc_w*, double * *fc_r*, double * *modelAlpha_out*, double * *modelVal*, double * *whiteNoiseLevel*, int *realflag*, int *dcmflag*)
- 12.13.1.10 void [T2getHighFreqRes](#) (double * *resy*, double * *smoothModel*, int *nres*, double * *highFreqRes*)
- 12.13.1.11 void [T2getWhiteNoiseLevel](#) (int *n*, double * *y*, int *nlast*, double * *av*)
- 12.13.1.12 void [T2getWhiteRes](#) (double * *resx*, double * *resy*, double * *rese*, int *nres*, double ** *uin*, double * *cholWhiteY*)

12.13.1.13 `int T2guess_vals (double * x, double * y, int n, double * alpha, double * amp, double * fc, int * nfit, double wn, double * fc_white, int prewhite)`

12.13.1.14 `void T2interpolate (double * resx, double * resy, double * rese, int nres, double * cubicVal, double * interpX, double * interpY, int * nlinterp, int interpTime, double expSmooth)`

12.13.1.15 `int T2obtainTimingResiduals (pulsar * psr, double * resx, double * resy, double * rese)`

12.13.1.16 `void T2writeCovarFuncModel (double alpha, double fc, double val, double white, char * fname)`

12.13.2 Variable Documentation

12.13.2.1 `double EXPSMOOTH`

12.13.2.2 `double FCALPHA`

12.13.2.3 `double FCFINAL`

12.13.2.4 `double NFIT`

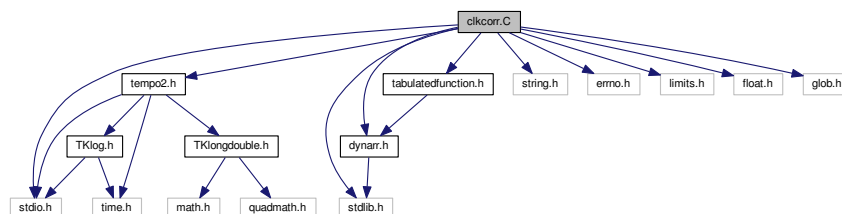
12.13.2.5 `double UPW`

12.13.2.6 `double WNLEVEL`

12.14 clkcorr.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <limits.h>
#include <float.h>
#include <glob.h>
#include "tempo2.h"
#include "dynarr.h"
#include "tabulatedfunction.h"
```

Include dependency graph for clkcorr.C:



Classes

- struct [ClockCorrectionFunction](#)

Functions

- void [ClockCorrectionFunction_load](#) ([ClockCorrectionFunction](#) *func, char *fileName)
- double [ClockCorrectionFunction_getCorrection](#) ([ClockCorrectionFunction](#) *func, double mjd)
- double [ClockCorrectionFunction_getStartMJD](#) ([ClockCorrectionFunction](#) *func)
- double [ClockCorrectionFunction_getEndMJD](#) ([ClockCorrectionFunction](#) *func)
- double [ClockCorrectionSequence_getStartMJD](#) ([DynamicArray](#) *sequence)
- double [ClockCorrectionSequence_getEndMJD](#) ([DynamicArray](#) *sequence)
- void [initialize_ClockCorrections](#) (int dispWarnings)
- void [defineClockCorrectionSequence](#) (char *fileList_in, int dispWarnings)
- [DynamicArray](#) * [makeClockCorrectionSequence](#) (const char *clockFrom, const char *clockTo, double mjd, int warnings)
- [DynamicArray](#) * [getClockCorrectionSequence](#) (const char *clockFrom, const char *clockTo, double mjd, int warnings)
- void [getClockCorrections](#) ([observation](#) *obs, const char *clockFrom_const, const char *clockTo, int warnings)
- double [getCorrectionTT](#) ([observation](#) *obs)
- double [getCorrection](#) ([observation](#) *obs, const char *clockFrom_c, const char *clockTo, int warnings)

Variables

- [DynamicArray](#) clockCorrectionFunctions
- [DynamicArray](#) clockCorrectionSequences

12.14.1 Function Documentation

- 12.14.1.1 double [ClockCorrectionFunction_getCorrection](#) ([ClockCorrectionFunction](#) * *func*, double *mjd*)
- 12.14.1.2 double [ClockCorrectionFunction_getEndMJD](#) ([ClockCorrectionFunction](#) * *func*)
- 12.14.1.3 double [ClockCorrectionFunction_getStartMJD](#) ([ClockCorrectionFunction](#) * *func*)
- 12.14.1.4 void [ClockCorrectionFunction_load](#) ([ClockCorrectionFunction](#) * *func*, char * *fileName*)
- 12.14.1.5 double [ClockCorrectionSequence_getEndMJD](#) ([DynamicArray](#) * *sequence*)
- 12.14.1.6 double [ClockCorrectionSequence_getStartMJD](#) ([DynamicArray](#) * *sequence*)
- 12.14.1.7 void [defineClockCorrectionSequence](#) (char * *fileList_in*, int *dispWarnings*)
- 12.14.1.8 void [getClockCorrections](#) ([observation](#) * *obs*, const char * *clockFrom_const*, const char * *clockTo*, int *warnings*)
- 12.14.1.9 [DynamicArray](#)* [getClockCorrectionSequence](#) (const char * *clockFrom*, const char * *clockTo*, double *mjd*, int *warnings*)
- 12.14.1.10 double [getCorrection](#) ([observation](#) * *obs*, const char * *clockFrom_c*, const char * *clockTo*, int *warnings*)
- 12.14.1.11 double [getCorrectionTT](#) ([observation](#) * *obs*)
- 12.14.1.12 void [initialize_ClockCorrections](#) (int *dispWarnings*)
- 12.14.1.13 [DynamicArray](#)* [makeClockCorrectionSequence](#) (const char * *clockFrom*, const char * *clockTo*, double *mjd*, int *warnings*)

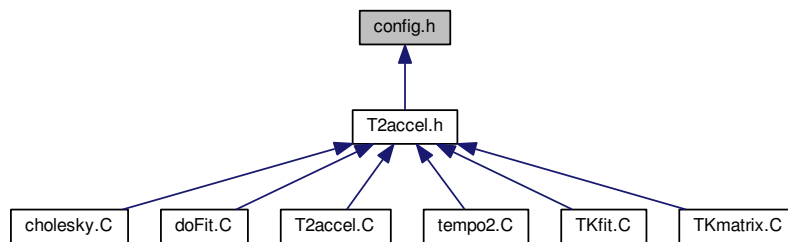
12.14.2 Variable Documentation

12.14.2.1 DynamicArray clockCorrectionFunctions

12.14.2.2 DynamicArray clockCorrectionSequences

12.15 config.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [F77_FUNC](#)(name, NAME) name ## _
- #define [F77_FUNC_](#)(name, NAME) name ## _
- #define [HAVE_BLAS](#) 1
- #define [HAVE_DLERROR](#) 1
- #define [HAVE_DLFCN_H](#) 1
- #define [HAVE_FFTW3](#) 1
- #define [HAVE_INTTYPES_H](#) 1
- #define [HAVE_LAPACK](#) 1
- #define [HAVE_LIBDL](#) 1
- #define [HAVE_LIBDLLOADER](#) 1
- #define [HAVE_LIBM](#) 1
- #define [HAVE_MEMORY_H](#) 1
- #define [HAVE_PGPLOT](#) 1
- #define [HAVE_PTHREAD](#) 1
- #define [HAVE_STDINT_H](#) 1
- #define [HAVE_STDLIB_H](#) 1
- #define [HAVE_STRINGS_H](#) 1
- #define [HAVE_STRING_H](#) 1
- #define [HAVE_SYS_STAT_H](#) 1
- #define [HAVE_SYS_TYPES_H](#) 1
- #define [HAVE_UNISTD_H](#) 1
- #define [LT_OBJDIR](#) ".libs/"
- #define [PACKAGE](#) "tempo2"
- #define [PACKAGE_BUGREPORT](#) "george.hobbs@csiro.au"
- #define [PACKAGE_NAME](#) "Tempo2"
- #define [PACKAGE_STRING](#) "Tempo2 2015.09.0"
- #define [PACKAGE_TARNAME](#) "tempo2"
- #define [PACKAGE_URL](#) "http://www.bitbucket.org/mkeith/tempo2"
- #define [PACKAGE_VERSION](#) "2015.09.0"

- `#define STDC_HEADERS 1`
- `#define TEMPO2_ARCH "linux-gnu"`
- `#define VERSION "2015.09.0"`
- `#define X_DISPLAY_MISSING 1`
- `#define _DARWIN_USE_64_BIT_INODE 1`

12.15.1 Macro Definition Documentation

12.15.1.1 `#define _DARWIN_USE_64_BIT_INODE 1`

12.15.1.2 `#define F77_FUNC(name, NAME) name ## _`

12.15.1.3 `#define F77_FUNC_(name, NAME) name ## _`

12.15.1.4 `#define HAVE_BLAS 1`

12.15.1.5 `#define HAVE_DLERROR 1`

12.15.1.6 `#define HAVE_DLFCN_H 1`

12.15.1.7 `#define HAVE_FFTW3 1`

12.15.1.8 `#define HAVE_INTPYPES_H 1`

12.15.1.9 `#define HAVE_LAPACK 1`

12.15.1.10 `#define HAVE_LIBDL 1`

12.15.1.11 `#define HAVE_LIBDLLOADER 1`

12.15.1.12 `#define HAVE_LIBM 1`

12.15.1.13 `#define HAVE_MEMORY_H 1`

12.15.1.14 `#define HAVE_PGPLOT 1`

12.15.1.15 `#define HAVE_PTHREAD 1`

12.15.1.16 `#define HAVE_STDINT_H 1`

12.15.1.17 `#define HAVE_STDLIB_H 1`

12.15.1.18 `#define HAVE_STRING_H 1`

12.15.1.19 `#define HAVE_STRINGS_H 1`

12.15.1.20 `#define HAVE_SYS_STAT_H 1`

12.15.1.21 `#define HAVE_SYS_TYPES_H 1`

12.15.1.22 `#define HAVE_UNISTD_H 1`

12.15.1.23 `#define LT_OBJDIR ".libs/"`

12.15.1.24 `#define PACKAGE "tempo2"`

```

12.15.1.25 #define PACKAGE_BUGREPORT "george.hobbs@csiro.au"

12.15.1.26 #define PACKAGE_NAME "Tempo2"

12.15.1.27 #define PACKAGE_STRING "Tempo2 2015.09.0"

12.15.1.28 #define PACKAGE_TARNAME "tempo2"

12.15.1.29 #define PACKAGE_URL "http://www.bitbucket.org/mkeith/tempo2"

12.15.1.30 #define PACKAGE_VERSION "2015.09.0"

12.15.1.31 #define STDC_HEADERS 1

12.15.1.32 #define TEMPO2_ARCH "linux-gnu"

12.15.1.33 #define VERSION "2015.09.0"

12.15.1.34 #define X_DISPLAY_MISSING 1

```

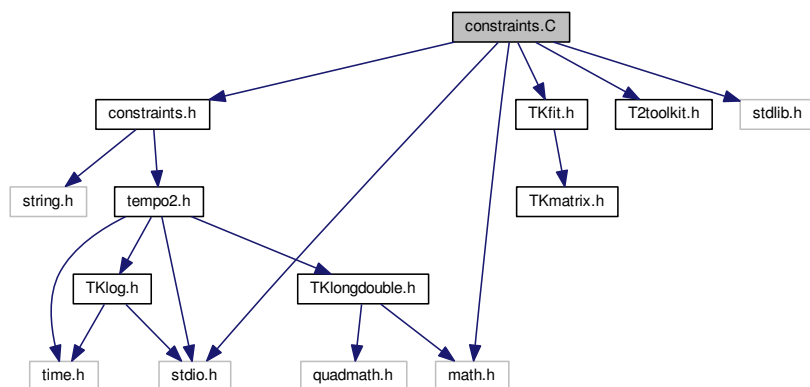
12.16 constraints.C File Reference

```

#include "constraints.h"
#include "TKfit.h"
#include "T2toolkit.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

```

Include dependency graph for constraints.C:



Functions

- `std::string get_constraint_name` (unsigned c)
- `std::string get_constraint_name` (enum `constraint` c)
- `void matrixDMConstraintWeights` (`pulsar *psr`)
- `void computeConstraintWeights` (`pulsar *psr`)
- `double consFunc_dmmodel_mean` (`pulsar *psr_array`, int ipsr, int i, int k, int order)
- `double consFunc_dmmodel_dm1` (`pulsar *psr_array`, int ipsr, int i, int k, int order)

- double `consFunc_dmmodel_cw` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_dmmodel_cw_year` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_tel_dx` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_tel_dy` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_tel_dz` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_ifunc` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_qifunc_p_year` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_qifunc_c_year` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_ifunc_year` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_quad_ifunc_p` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_quad_ifunc_c` (`pulsar *psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- void `autoConstraints` (`pulsar *psr_array`, `int ipsr`, `int npsr`)
- void `autosetDMCM` (`pulsar *psr`, `double dmstep`, `double cmstep`, `double start`, `double end`, `bool fixCMgrid`)
- double `getConstraintDeriv` (`pulsar *psr`, `int iconstraint`, `int i`, `int k`)
- double `standardConstraintFunctions` (`pulsar *psr`, `int ipsr`, `int iconstraint`, `int iparam`, `int constraintk`, `int k`)
- void `CONSTRAINTfuncs` (`pulsar *psr`, `int ipsr`, `int nparams`, `int iconstraint`, `double *OUT`)

12.16.1 Function Documentation

- 12.16.1.1 void `autoConstraints` (`pulsar * psr_array`, `int ipsr`, `int npsr`)
- 12.16.1.2 void `autosetDMCM` (`pulsar * psr`, `double dmstep`, `double cmstep`, `double start`, `double end`, `bool fixCMgrid`)
- 12.16.1.3 void `computeConstraintWeights` (`pulsar * psr`)
- 12.16.1.4 double `consFunc_dmmodel_cw` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.5 double `consFunc_dmmodel_cw_year` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.6 double `consFunc_dmmodel_dm1` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.7 double `consFunc_dmmodel_mean` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.8 double `consFunc_ifunc` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.9 double `consFunc_ifunc_year` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.10 double `consFunc_qifunc_c_year` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.11 double `consFunc_qifunc_p_year` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.12 double `consFunc_quad_ifunc_c` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.13 double `consFunc_quad_ifunc_p` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.14 double `consFunc_tel_dx` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.15 double `consFunc_tel_dy` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.16 double `consFunc_tel_dz` (`pulsar * psr_array`, `int ipsr`, `int i`, `int k`, `int order`)
- 12.16.1.17 void `CONSTRAINTfuncs` (`pulsar * psr`, `int ipsr`, `int nparams`, `int iconstraint`, `double * OUT`)
- 12.16.1.18 std::string `get_constraint_name` (`unsigned c`)

12.16.1.19 `std::string get_constraint_name (enum constraint c)`

12.16.1.20 `double getConstraintDeriv (pulsar * psr, int iconstraint, int i, int k)`

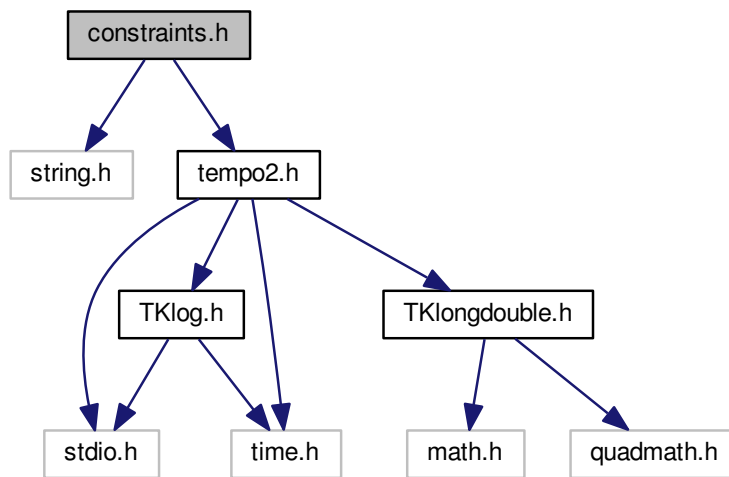
12.16.1.21 `void matrixDMConstraintWeights (pulsar * psr)`

12.16.1.22 `double standardConstraintFunctions (pulsar * psr, int ipsr, int iconstraint, int iparam, int constraintk, int k)`

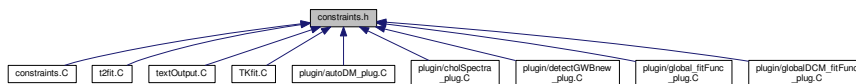
12.17 constraints.h File Reference

```
#include <string.h>
#include "tempo2.h"
```

Include dependency graph for constraints.h:



This graph shows which files directly or indirectly include this file:



Functions

- `std::string get_constraint_name (enum constraint c)`
- `void computeConstraintWeights (pulsar * psr)`
- `double consFunc_dmmodel_mean (pulsar * psr, int ipsr, int i, int k, int order)`
- `double consFunc_dmmodel_dm1 (pulsar * psr, int ipsr, int i, int k, int order)`
- `double consFunc_dmmodel_cw (pulsar * psr, int ipsr, int i, int k, int order)`
- `double consFunc_dmmodel_cw_year (pulsar * psr, int ipsr, int i, int k, int order)`
- `double consFunc_ifunc (pulsar * psr, int ipsr, int i, int k, int order)`
- `double consFunc_ifunc_year (pulsar * psr, int ipsr, int i, int k, int order)`

- double `consFunc_tel_dx` (`pulsar *psr`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_tel_dy` (`pulsar *psr`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_tel_dz` (`pulsar *psr`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_quad_ifunc_p` (`pulsar *psr`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_quad_ifunc_c` (`pulsar *psr`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_qifunc_p_year` (`pulsar *psr`, `int ipsr`, `int i`, `int k`, `int order`)
- double `consFunc_qifunc_c_year` (`pulsar *psr`, `int ipsr`, `int i`, `int k`, `int order`)
- void `autosetDMCM` (`pulsar *psr`, `double dmstep`, `double cmstep`, `double start`, `double end`, `bool fixCMgrid`)
- void `CONSTRAINTfuncs` (`pulsar *psr`, `int ipsr`, `int nparams`, `int iconstraint`, `double *OUT`)
- double `standardConstraintFunctions` (`pulsar *psr`, `int ipsr`, `int iconstraint`, `int iparam`, `int constraintk`, `int k`)

12.17.1 Function Documentation

12.17.1.1 void `autosetDMCM` (`pulsar * psr`, `double dmstep`, `double cmstep`, `double start`, `double end`, `bool fixCMgrid`)

12.17.1.2 void `computeConstraintWeights` (`pulsar * psr`)

12.17.1.3 double `consFunc_dmmodel_cw` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.4 double `consFunc_dmmodel_cw_year` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.5 double `consFunc_dmmodel_dm1` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.6 double `consFunc_dmmodel_mean` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.7 double `consFunc_ifunc` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.8 double `consFunc_ifunc_year` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.9 double `consFunc_qifunc_c_year` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.10 double `consFunc_qifunc_p_year` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.11 double `consFunc_quad_ifunc_c` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.12 double `consFunc_quad_ifunc_p` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.13 double `consFunc_tel_dx` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.14 double `consFunc_tel_dy` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.15 double `consFunc_tel_dz` (`pulsar * psr`, `int ipsr`, `int i`, `int k`, `int order`)

12.17.1.16 void `CONSTRAINTfuncs` (`pulsar * psr`, `int ipsr`, `int nparams`, `int iconstraint`, `double * OUT`)

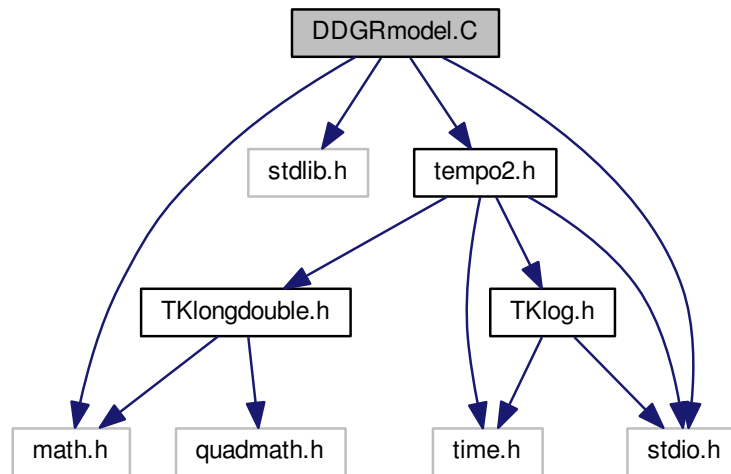
12.17.1.17 `std::string` `get_constraint_name` (`enum constraint c`)

12.17.1.18 double `standardConstraintFunctions` (`pulsar * psr`, `int ipsr`, `int iconstraint`, `int iparam`, `int constraintk`, `int k`)

12.18 DDGRmodel.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```


Include dependency graph for DDGRmodel.C:



Functions

- void [mass2dd](#) (double *am*, double *am2*, double *x*, double *ecc*, double *an*, double **arr*, double **ar*, double **xk*, double **si*, double **gamma*, double **pbdot*)
- double [DDGRmodel](#) ([pulsar](#) **psr*, int *p*, int *ipos*, int *param*)
- void [updateDDGR](#) ([pulsar](#) **psr*, double *val*, double *err*, int *pos*)

12.18.1 Function Documentation

12.18.1.1 double [DDGRmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*)

12.18.1.2 void [mass2dd](#) (double *am*, double *am2*, double *x*, double *ecc*, double *an*, double * *arr*, double * *ar*, double * *xk*, double * *si*, double * *gamma*, double * *pbdot*)

12.18.1.3 void [updateDDGR](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*)

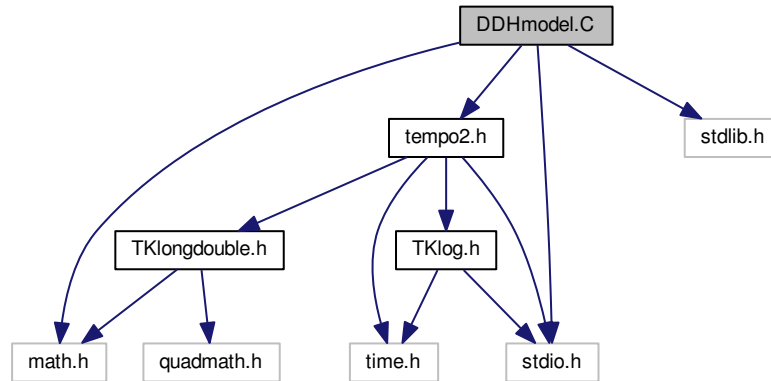
12.19 DDHmodel.C File Reference

```

#include <stdio.h>
#include <math.h>
#include "tempo2.h"
#include <stdlib.h>

```

Include dependency graph for DDHmodel.C:



Functions

- double **DDHmodel** (**pulsar** *psr, int p, int ipos, int param)
- void **updateDDH** (**pulsar** *psr, double val, double err, int pos)

12.19.1 Function Documentation

12.19.1.1 double **DDHmodel** (**pulsar** * *psr*, int *p*, int *ipos*, int *param*)

12.19.1.2 void **updateDDH** (**pulsar** * *psr*, double *val*, double *err*, int *pos*)

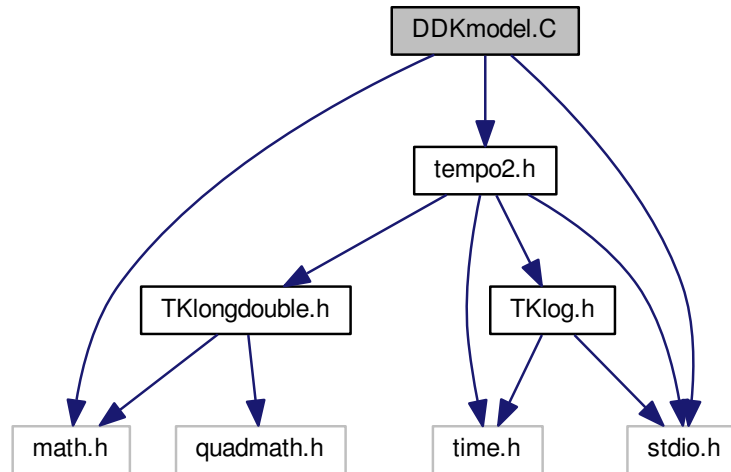
12.20 DDKmodel.C File Reference

```

#include <stdio.h>
#include <math.h>
#include "tempo2.h"

```

Include dependency graph for DDKmodel.C:



Functions

- double [DDKmodel](#) ([pulsar](#) *psr, int p, int ipos, int param)
- void [updateDDK](#) ([pulsar](#) *psr, double val, double err, int pos)

12.20.1 Function Documentation

12.20.1.1 double [DDKmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*)

12.20.1.2 void [updateDDK](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*)

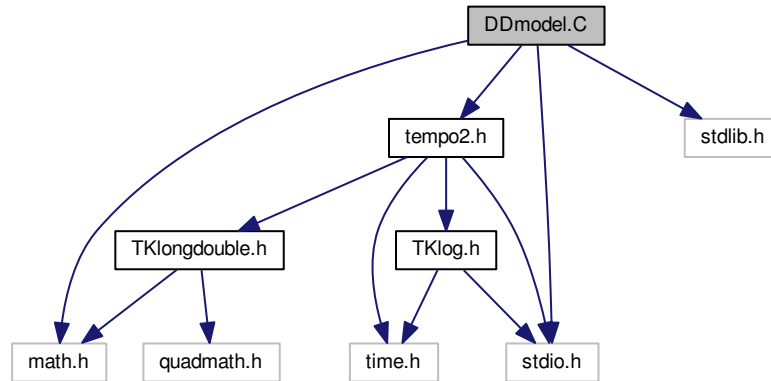
12.21 DDmodel.C File Reference

```

#include <stdio.h>
#include <math.h>
#include "tempo2.h"
#include <stdlib.h>

```

Include dependency graph for DDmodel.C:



Functions

- `longdouble DDmodel (pulsar *psr, int p, int ipos, int param)`
- `void updateDD (pulsar *psr, double val, double err, int pos)`

12.21.1 Function Documentation

12.21.1.1 `longdouble DDmodel (pulsar * psr, int p, int ipos, int param)`

12.21.1.2 `void updateDD (pulsar * psr, double val, double err, int pos)`

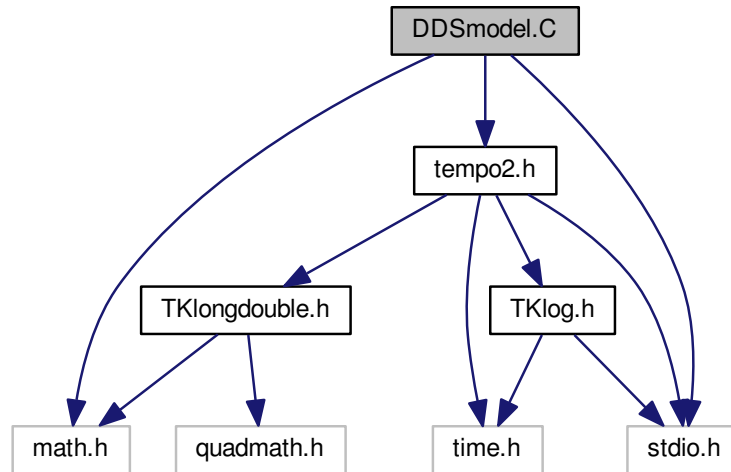
12.22 DDSmodel.C File Reference

```

#include <stdio.h>
#include <math.h>
#include "tempo2.h"

```

Include dependency graph for DDSmodel.C:



Functions

- double [DDSmodel](#) ([pulsar](#) *psr, int p, int ipos, int param)
- void [updateDDS](#) ([pulsar](#) *psr, double val, double err, int pos)

12.22.1 Function Documentation

12.22.1.1 double [DDSmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*)

12.22.1.2 void [updateDDS](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*)

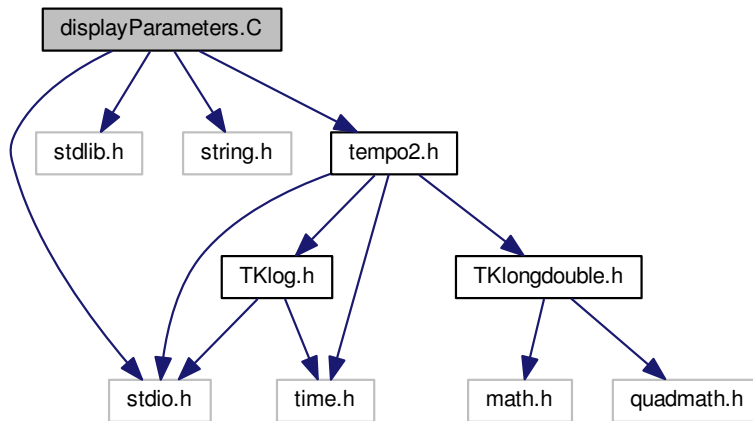
12.23 displayParameters.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"

```

Include dependency graph for displayParameters.C:



Functions

- void `displayParameters` (int `pos`, char `timFile`[][`MAX_FILELEN`], char `parFile`[][`MAX_FILELEN`], `pulsar` *`psr`, int `npsr`)

12.23.1 Function Documentation

12.23.1.1 void `displayParameters` (int `pos`, char `timFile`[][`MAX_FILELEN`], char `parFile`[][`MAX_FILELEN`], `pulsar` * `psr`, int `npsr`)

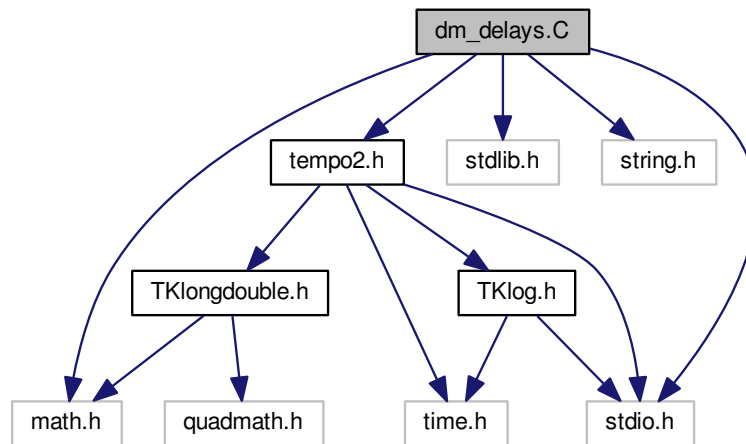
12.24 dm_delays.C File Reference

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"

```

Include dependency graph for dm_delays.C:



Functions

- `longdouble solarWindModel (pulsar psr)`
- `void dm_delays (pulsar *psr, int npsr, int p, int i, double delt, double dt_SSB)`

12.24.1 Function Documentation

12.24.1.1 `void dm_delays (pulsar * psr, int npsr, int p, int i, double delt, double dt_SSB)`

12.24.1.2 `longdouble solarWindModel (pulsar psr)`

12.25 documentation/DEVELOPER_GUIDE.md File Reference

12.26 documentation/developers.md File Reference

12.27 documentation/directories.md File Reference

12.28 documentation/USER_GUIDE.md File Reference

12.29 doFit.C File Reference

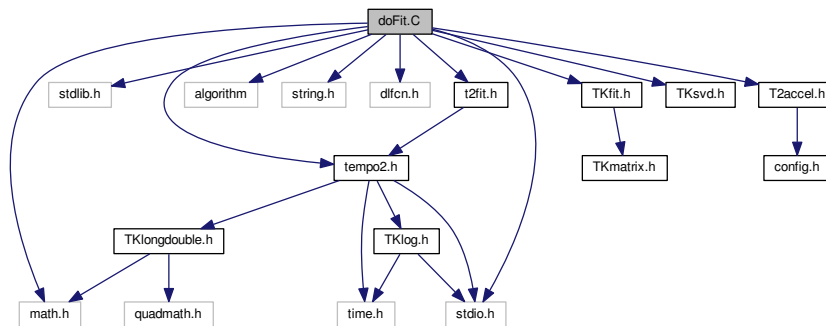
```
#include <stdio.h>
```

```

#include <stdlib.h>
#include <math.h>
#include <algorithm>
#include <string.h>
#include <dlfcn.h>
#include "tempo2.h"
#include "TKfit.h"
#include "t2fit.h"
#include "TKsvd.h"
#include "T2accel.h"

```

Include dependency graph for doFit.C:



Functions

- void [globalFITfuncs](#) (double x, double afunc[], int ma, [pulsar](#) *psr, int ipos, int ipsr)
- void [updateGlobalParameters](#) ([pulsar](#) *psr, int npsr, double *val, double *error)
- int [getNparams](#) ([pulsar](#) *psr, int offset)
- int [getNglobal](#) ([pulsar](#) *psr, int npsr)
- double [getConstraintDeriv](#) ([pulsar](#) *psr, int ipos, int i, int k)
- void [getTempoNestMaxLike](#) ([pulsar](#) *pulse, int npsr)
- void [dgesvd_ftoc](#) (double *in, double **out, int rows, int cols)
- double * [dgesvd_ctof](#) (double **in, int rows, int cols)
- void [dgesvd](#) (double **A, int m, int n, double *S, double **U, double **VT)
- void [dgemv](#) (double **A, double *vecin, double *vecout, int rowa, int cola, char AT)
- double * [dgemv_ctof](#) (double **in, int rows, int cols)
- void [dgemv_ftoc](#) (double *in, double **out, int rows, int cols)
- void [dgemm](#) (double **A, double **B, double **C, int rowa, int cola, int rowb, int colb, char AT, char BT)
- double * [dgemm_ctof](#) (double **in, int rows, int cols)
- void [dgemm_ftoc](#) (double *in, double **out, int rows, int cols)
- void [dpotri](#) (double **A, int msize)
- double * [dpotri_ctof](#) (double **in, int rows, int cols)
- void [dpotri_ftoc](#) (double *in, double **out, int rows, int cols)
- void [dpotrf](#) (double **A, int msize, double &det)
- double * [dpotrf_ctof](#) (double **in, int rows, int cols)
- void [dpotrf_ftoc](#) (double *in, double **out, int rows, int cols)
- void [dpotrf](#) (char *UPLO, int *msize, double *a, int *lda, int *info)
- void [dpotri](#) (char *UPLO, int *msize, double *a, int *lda, int *info)
- void [dgesvd](#) (char *jobu, char *jobvt, int *m, int *n, double *a, int *lda, double *s, double *u, int *ldu, double *vt, int *ldvt, double *work, int *lwork, int *info)
- void [dgemv](#) (char *jobu, int *m, int *n, double *alpha, double *a, int *lda, double *x, int *incx, double *beta, double *y, int *incy)

- void `dgemm_` (char *jobu, char *jobvt, int *m, int *n, int *k, double *alpha, double *a, int *lda, double *b, int *ldb, double *beta, double *c, int *ldc)
- void `doFit` (pulsar *psr, int npsr, int writeModel)
- void `doFitDCM` (pulsar *psr, const char *dcmFile, const char *covarFuncFile, int npsr, int writeModel)
- void `doFitOLD` (pulsar *psr, int npsr, const char *covarFuncFile)
- void `doFitAll` (pulsar *psr, int npsr, const char *covarFuncFile)
- void `FITfuncs` (double x, double afunc[], int ma, pulsar *psr, int ipos, int ipsr)
- double `getParamDeriv` (pulsar *psr, int ipos, double x, int i, int k)
- void `updateParameters` (pulsar *psr, int p, double *val, double *error)
- void `othpl` (int n, double x, double *pl)

12.29.1 Function Documentation

- 12.29.1.1 void `dgemm` (double ** *A*, double ** *B*, double ** *C*, int *rowa*, int *cola*, int *rowb*, int *colb*, char *AT*, char *BT*)
- 12.29.1.2 void `dgemm_` (char * *jobu*, char * *jobvt*, int * *m*, int * *n*, int * *k*, double * *alpha*, double * *a*, int * *lda*, double * *b*, int * *ldb*, double * *beta*, double * *c*, int * *ldc*)
- 12.29.1.3 double * `dgemm_ctof` (double ** *in*, int *rows*, int *cols*)
- 12.29.1.4 void `dgemm_ftoc` (double * *in*, double ** *out*, int *rows*, int *cols*)
- 12.29.1.5 void `dgemv` (double ** *A*, double * *vecin*, double * *vecout*, int *rowa*, int *cola*, char *AT*)
- 12.29.1.6 void `dgemv_` (char * *jobu*, int * *m*, int * *n*, double * *alpha*, double * *a*, int * *lda*, double * *x*, int * *incx*, double * *beta*, double * *y*, int * *incy*)
- 12.29.1.7 double * `dgemv_ctof` (double ** *in*, int *rows*, int *cols*)
- 12.29.1.8 void `dgemv_ftoc` (double * *in*, double ** *out*, int *rows*, int *cols*)
- 12.29.1.9 void `dgesvd` (double ** *A*, int *m*, int *n*, double * *S*, double ** *U*, double ** *VT*)
- 12.29.1.10 void `dgesvd_` (char * *jobu*, char * *jobvt*, int * *m*, int * *n*, double * *a*, int * *lda*, double * *s*, double * *u*, int * *ldu*, double * *vt*, int * *ldvt*, double * *work*, int * *lwork*, int * *info*)
- 12.29.1.11 double * `dgesvd_ctof` (double ** *in*, int *rows*, int *cols*)
- 12.29.1.12 void `dgesvd_ftoc` (double * *in*, double ** *out*, int *rows*, int *cols*)
- 12.29.1.13 void `doFit` (pulsar * *psr*, int *npsr*, int *writeModel*)
- 12.29.1.14 void `doFitAll` (pulsar * *psr*, int *npsr*, const char * *covarFuncFile*)

Master fitting routine with or without cholesky, global or not.

12.29.1.15 void `doFitDCM` (pulsar * *psr*, const char * *dcmFile*, const char * *covarFuncFile*, int *npsr*, int *writeModel*)

12.29.1.16 void `doFitOLD` (pulsar * *psr*, int *npsr*, const char * *covarFuncFile*)

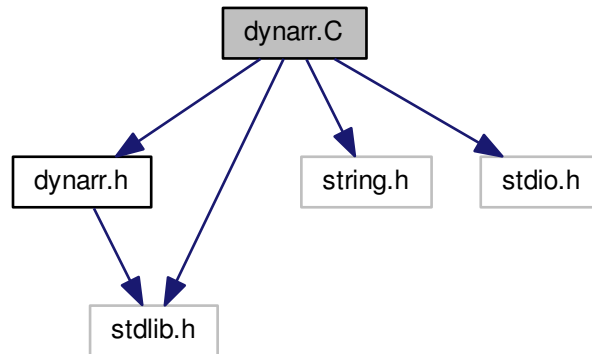
Master fitting routine with or without cholesky, global or not.

- 12.29.1.17 `void dpotrf (double ** A, int msize, double & det)`
- 12.29.1.18 `void dpotrf_ (char * UPLO, int * msize, double * a, int * lda, int * info)`
- 12.29.1.19 `double * dpotrf_ctof (double ** in, int rows, int cols)`
- 12.29.1.20 `void dpotrf_ftoc (double * in, double ** out, int rows, int cols)`
- 12.29.1.21 `void dpotri (double ** A, int msize)`
- 12.29.1.22 `void dpotri_ (char * UPLO, int * msize, double * a, int * lda, int * info)`
- 12.29.1.23 `double * dpotri_ctof (double ** in, int rows, int cols)`
- 12.29.1.24 `void dpotri_ftoc (double * in, double ** out, int rows, int cols)`
- 12.29.1.25 `void FITfuncs (double x, double afunc[], int ma, pulsar * psr, int ipos, int ipsr)`
- 12.29.1.26 `double getConstraintDeriv (pulsar * psr, int ipos, int i, int k)`
- 12.29.1.27 `int getNglobal (pulsar * psr, int npsr)`
- 12.29.1.28 `int getNparams (pulsar * psr, int offset)`
- 12.29.1.29 `double getParamDeriv (pulsar * psr, int ipos, double x, int i, int k)`
- psr->param[param_f].val[0];
-
- 12.29.1.30 `void getTempoNestMaxLike (pulsar * pulse, int npsr)`
- 12.29.1.31 `void globalFITfuncs (double x, double afunc[], int ma, pulsar * psr, int ipos, int ipsr)`
- 12.29.1.32 `void othpl (int n, double x, double * pl)`
- 12.29.1.33 `void updateGlobalParameters (pulsar * psr, int npsr, double * val, double * error)`
- 12.29.1.34 `void updateParameters (pulsar * psr, int p, double * val, double * error)`

12.30 dynarr.C File Reference

```
#include "dynarr.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

Include dependency graph for dynarr.C:



Functions

- void `DynamicArray_init` (`DynamicArray *a`, `size_t elemSize`)
- void `DynamicArray_resize` (`DynamicArray *a`, `size_t nelem`)
- void * `DynamicArray_push_back` (`DynamicArray *a`, void *`elem`)
- void `DynamicArray_free` (`DynamicArray *a`)

12.30.1 Function Documentation

12.30.1.1 void `DynamicArray_free` (`DynamicArray * a`)

12.30.1.2 void `DynamicArray_init` (`DynamicArray * a`, `size_t elemSize`)

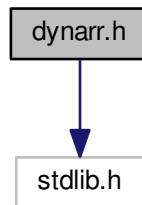
12.30.1.3 void* `DynamicArray_push_back` (`DynamicArray * a`, void * `elem`)

12.30.1.4 void `DynamicArray_resize` (`DynamicArray * a`, `size_t nelem`)

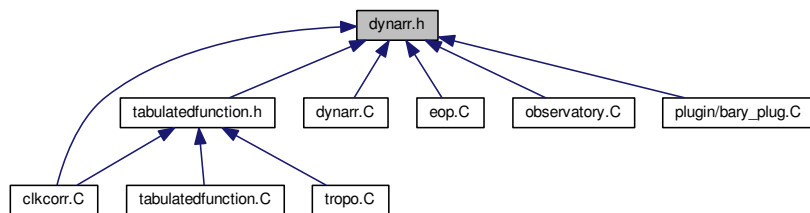
12.31 dynarr.h File Reference

```
#include <stdlib.h>
```

Include dependency graph for dynarr.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [DynamicArray](#)

Functions

- void [DynamicArray_init](#) ([DynamicArray](#) *, [size_t](#) elemSize)
- void [DynamicArray_resize](#) ([DynamicArray](#) *, [size_t](#) nelem)
- void * [DynamicArray_push_back](#) ([DynamicArray](#) *, void *elem)
- void [DynamicArray_free](#) ([DynamicArray](#) *)

12.31.1 Function Documentation

12.31.1.1 void [DynamicArray_free](#) ([DynamicArray](#) *)

12.31.1.2 void [DynamicArray_init](#) ([DynamicArray](#) *, [size_t](#) elemSize)

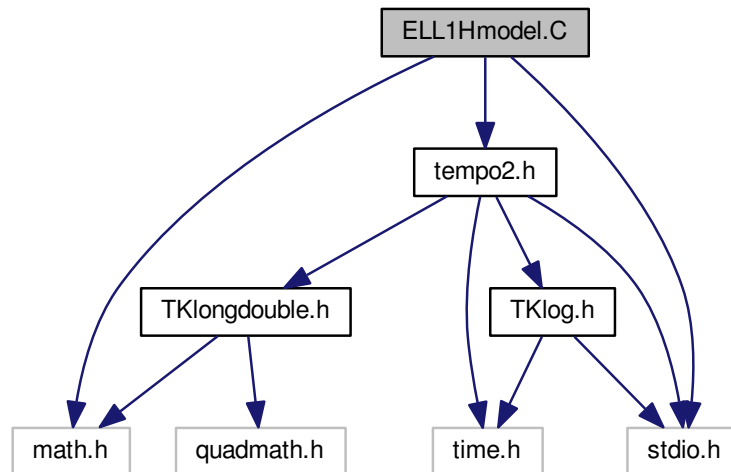
12.31.1.3 void* [DynamicArray_push_back](#) ([DynamicArray](#) *, void * elem)

12.31.1.4 void [DynamicArray_resize](#) ([DynamicArray](#) *, [size_t](#) nelem)

12.32 ELL1Hmodel.C File Reference

```
#include <stdio.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for ELL1Hmodel.C:



Functions

- double [ELL1Hmodel](#) ([pulsar](#) *psr, int p, int ipos, int param)
- void [updateELL1H](#) ([pulsar](#) *psr, double val, double err, int pos)

12.32.1 Function Documentation

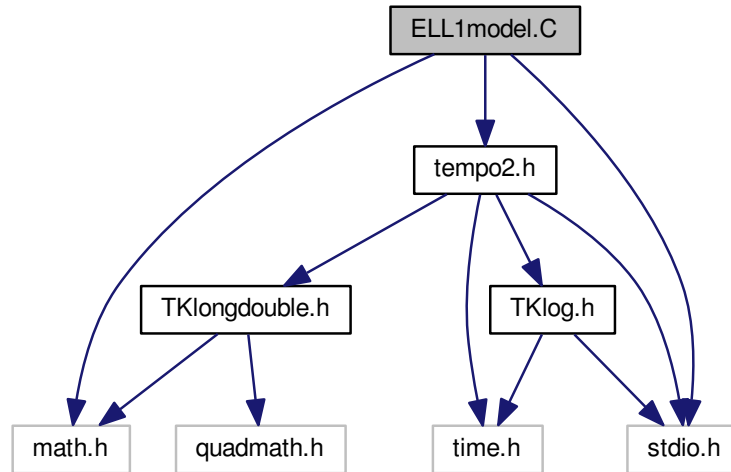
12.32.1.1 double [ELL1Hmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*)

12.32.1.2 void [updateELL1H](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*)

12.33 ELL1model.C File Reference

```
#include <stdio.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for ELL1model.C:



Functions

- double [ELL1model](#) ([pulsar](#) *psr, int p, int ipos, int param)
- void [updateELL1](#) ([pulsar](#) *psr, double val, double err, int pos)

12.33.1 Function Documentation

12.33.1.1 double [ELL1model](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*)

12.33.1.2 void [updateELL1](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*)

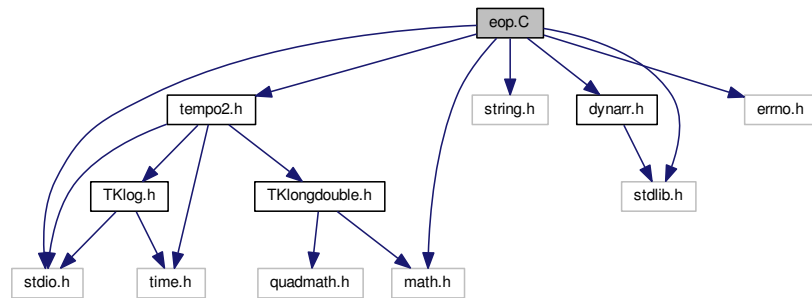
12.34 eop.C File Reference

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <errno.h>
#include "dynarr.h"
#include "tempo2.h"

```

Include dependency graph for eop.C:



Classes

- struct [EOPSample](#)

Functions

- void [load_EOP](#) ([DynamicArray](#) *EOPsamples, char *eopcFile)
- void [get_EOP](#) (double mjd, double *xp, double *yp, double *dut1, double *dut1dot, int dispWarnings, char *eopcFile)

12.34.1 Function Documentation

12.34.1.1 void [get_EOP](#) (double *mjd*, double * *xp*, double * *yp*, double * *dut1*, double * *dut1dot*, int *dispWarnings*, char * *eopcFile*)

12.34.1.2 void [load_EOP](#) ([DynamicArray](#) * *EOPsamples*, char * *eopcFile*)

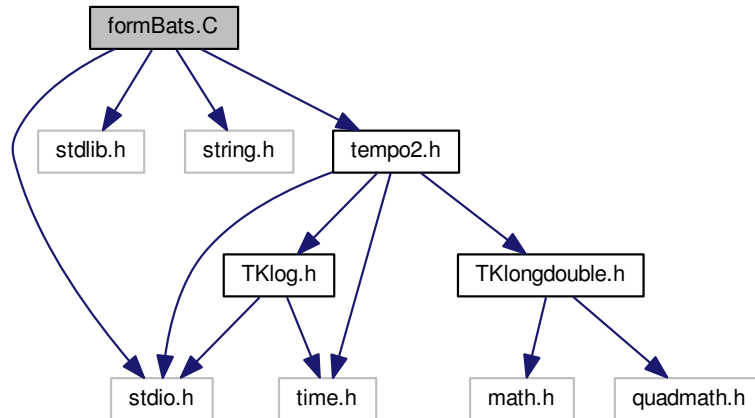
12.35 formBats.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"

```

Include dependency graph for formBats.C:



Functions

- void [formBats](#) ([pulsar](#) *psr, int npsr)

12.35.1 Function Documentation

12.35.1.1 void [formBats](#) ([pulsar](#) * *psr*, int *npsr*)

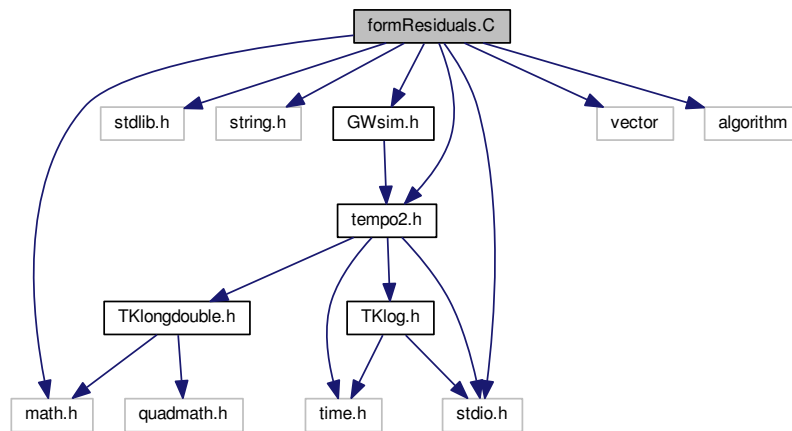
12.36 formResiduals.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
#include <vector>
#include <algorithm>

```


Include dependency graph for formResiduals.C:



Functions

- void [residualTracking](#) ([pulsar](#) *psr)
- void [averageResiduals](#) ([pulsar](#) *psr, int npsr)
- void [formResiduals](#) ([pulsar](#) *psr, int npsr, int [removeMean](#))

12.36.1 Function Documentation

12.36.1.1 void [averageResiduals](#) ([pulsar](#) * *psr*, int *npsr*)

12.36.1.2 void [formResiduals](#) ([pulsar](#) * *psr*, int *npsr*, int *removeMean*)

12.36.1.3 void [residualTracking](#) ([pulsar](#) * *psr*)

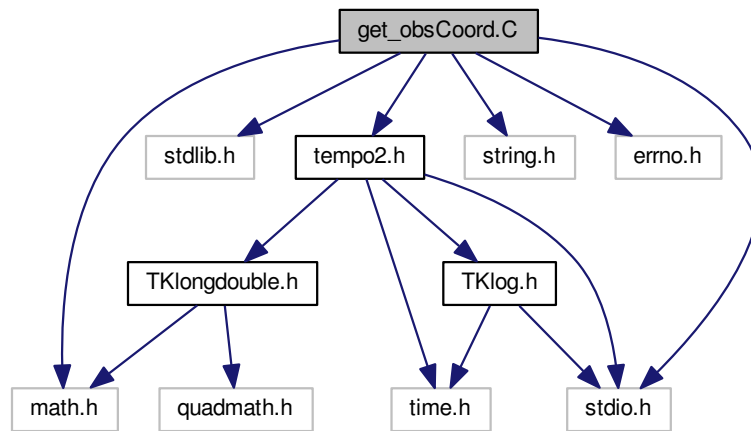
12.37 get_obsCoord.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <errno.h>
#include "tempo2.h"

```

Include dependency graph for `get_obsCoord.C`:



Functions

- double [get_precessionMatrix](#) (double prn[3][3], double mjd, double delp, double dele)
- void [remove_white](#) (char *str)
- double [lmst](#) (double mjd, double along, double *tsid, double *tsid_der)
- double [ang](#) (int i, double f)
- long [iau_cp_](#) (double *, double *)
- long [iau_pom00_](#) (double *, double *, double *, double *)
- long [iau_rxp_](#) (double *, double *, double *)
- long [iau_sxp_](#) (double *, double *, double *)
- long [iau_pxp_](#) (double *, double *, double *)
- long [iau_c2t00b_](#) (double *, double *, double *, double *, double *, double *, double *)
- long [iau_trxpv_](#) (double *, double *, double *)
- long [iau_trxp_](#) (double *, double *, double *)
- void [get_obsCoord_IAU2000B](#) (double observatory_trs[3], double zenith_trs[3], longdouble tt_mjd, longdouble utc_mjd, double observatory_crs[3], double zenith_crs[3], double observatory_velocity_crs[3], char *eopcFile)
- void [get_obsCoord](#) (pulsar *psr, int npsr)

12.37.1 Function Documentation

12.37.1.1 double [ang](#) (int *i*, double *f*)

12.37.1.2 void [get_obsCoord](#) (pulsar * *psr*, int *npsr*)

12.37.1.3 void [get_obsCoord_IAU2000B](#) (double *observatory_trs*[3], double *zenith_trs*[3], longdouble *tt_mjd*, longdouble *utc_mjd*, double *observatory_crs*[3], double *zenith_crs*[3], double *observatory_velocity_crs*[3], char * *eopcFile*)

12.37.1.4 double [get_precessionMatrix](#) (double *prn*[3][3], double *mjd*, double *delp*, double *dele*)

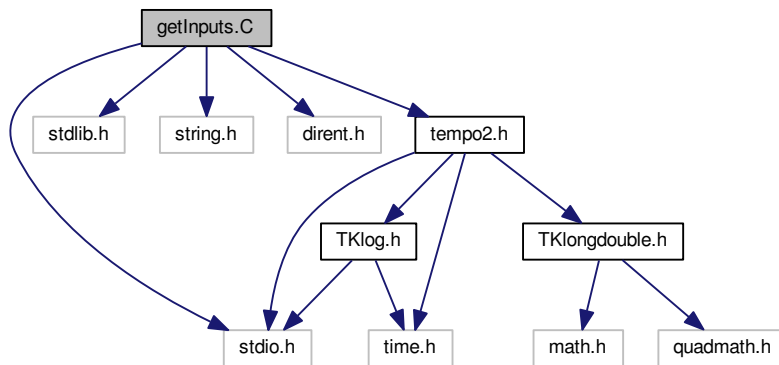
12.37.1.5 long [iau_c2t00b_](#) (double *, double *, double *, double *, double *, double *, double *)

- 12.37.1.6 `long iau_cp_ (double *, double *)`
- 12.37.1.7 `long iau_pom00_ (double *, double *, double *, double *)`
- 12.37.1.8 `long iau_pxp_ (double *, double *, double *)`
- 12.37.1.9 `long iau_rxp_ (double *, double *, double *)`
- 12.37.1.10 `long iau_sxp_ (double *, double *, double *)`
- 12.37.1.11 `long iau_trxp_ (double *, double *, double *)`
- 12.37.1.12 `long iau_trxpv_ (double *, double *, double *)`
- 12.37.1.13 `double lmst (double mjd, double olong, double * tsid, double * tsid_der)`
- 12.37.1.14 `void remove_white (char * str)`

12.38 getInputs.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <dirent.h>
#include "tempo2.h"
```

Include dependency graph for getInputs.C:



Functions

- void `printplugs` (bool full)
- void `getInputs` (pulsar *psr, int argc, char *argv[], char timFile[][MAX_FILELEN], char parFile[][MAX_FILELEN], int *list, int *npsr, int *nGlobal, int *outRes, int *writeModel, char *outputSO, int *polyco, char *polyco_args, char *polyco_file, int *newpar, int *onlypre, char *dcmFile, char *covarFuncFile, char *newparname)
- void `setPlugPath` ()

12.38.1 Function Documentation

12.38.1.1 void getInputs (pulsar * *psr*, int *argc*, char * *argv*[], char *timFile*[][MAX_FILELEN], char *parFile*[][MAX_FILELEN], int * *list*, int * *npsr*, int * *nGlobal*, int * *outRes*, int * *writeModel*, char * *outputSO*, int * *polyco*, char * *polyco_args*, char * *polyco_file*, int * *newpar*, int * *onlypre*, char * *dcmFile*, char * *covarFuncFile*, char * *newparname*)

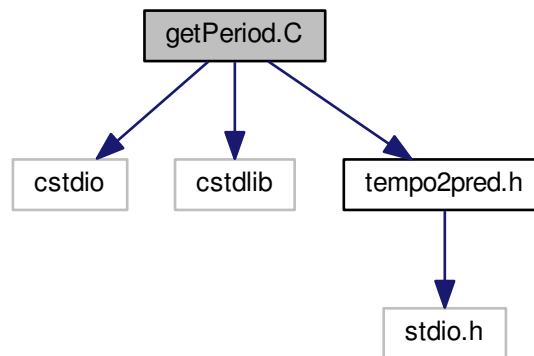
12.38.1.2 void printplugs (bool *full*)

12.38.1.3 void setPlugPath ()

12.39 getPeriod.C File Reference

```
#include <stdio>
#include <stdlib>
#include <tempo2pred.h>
```

Include dependency graph for getPeriod.C:



Functions

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

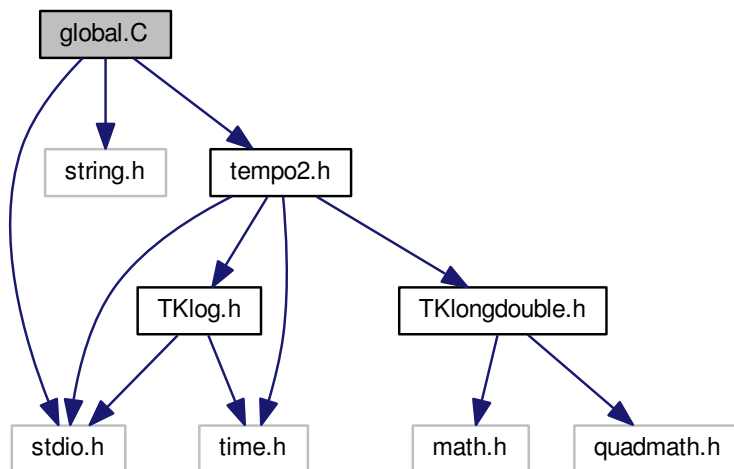
12.39.1 Function Documentation

12.39.1.1 int main (int *argc*, char ** *argv*)

12.40 global.C File Reference

```
#include <stdio.h>
#include <string.h>
#include "tempo2.h"
```

Include dependency graph for global.C:



Macros

- `#define MAX_FUNCTIONS 1024 /* Maximum functions in tempo2 */`

Functions

- void `extra_delays` (pulsar *psr, int npsr)
- void `clock_corrections` (pulsar *psr, int npsr)
- void `ephemeris_routines` (pulsar *psr, int npsr)
- void `formBatsAll` (pulsar *psr, int npsr)
- void `updateBatsAll` (pulsar *psr, int npsr)
- void `CVSdisplayVersion` (const char *file, const char *func, const char *verNum)

Variables

- char `TEMPO2_ENVIRON` [MAX_STRLen] = "TEMPO2"
- char `TEMPO2_ERROR` [MAX_STRLen] = ""
- char `NEWFIT` = 0
- int `MAX_PSR` = MAX_PSR_VAL
- int `MAX_OBSN` = MAX_OBSN_VAL
- double `ECLIPTIC_OBLIQUITY` = ECLIPTIC_OBLIQUITY_VAL
- int `forceGlobalFit` = 0
- int `veryFast` = 0
- int `displayCVSversion` = 0
- char `tempo2MachineType` [MAX_FILELEN] = ""
- char `dcmFile` [MAX_FILELEN] = "NULL"
- char `covarFuncFile` [MAX_FILELEN] = "NULL"
- char `tempo2_plug_path` [32][MAX_STRLen]
- int `tempo2_plug_path_len` = 0
- double `FCALPHA`

- double [WNLEVEL](#)
- double [EXPSMOOTH](#)
- double [UPW](#)
- double [NFIT](#)
- double [FCFINAL](#)

12.40.1 Macro Definition Documentation

12.40.1.1 `#define MAX_FUNCTIONS 1024 /* Maximum functions in tempo2 */`

12.40.2 Function Documentation

12.40.2.1 `void clock_corrections (pulsar * psr, int npsr)`

12.40.2.2 `void CVSDisplayVersion (const char * file, const char * func, const char * verNum)`

12.40.2.3 `void ephemeris_routines (pulsar * psr, int npsr)`

12.40.2.4 `void extra_delays (pulsar * psr, int npsr)`

12.40.2.5 `void formBatsAll (pulsar * psr, int npsr)`

12.40.2.6 `void updateBatsAll (pulsar * psr, int npsr)`

12.40.3 Variable Documentation

12.40.3.1 `char covarFuncFile[MAX_FILELEN] = "NULL"`

12.40.3.2 `char dcmFile[MAX_FILELEN] = "NULL"`

12.40.3.3 `int displayCVSversion = 0`

Display CVS version

12.40.3.4 `double ECLIPTIC_OBLIQUITY = ECLIPTIC_OBLIQUITY_VAL`

12.40.3.5 `double EXPSMOOTH`

12.40.3.6 `double FCALPHA`

12.40.3.7 `double FCFINAL`

12.40.3.8 `int forceGlobalFit = 0`

Global = 1 if we are forcing a global fit

12.40.3.9 `int MAX_OBSN = MAX_OBSN_VAL`

size of the arrays of [observations](#) inside each [pulsar](#)

12.40.3.10 `int MAX_PSR = MAX_PSR_VAL`

size of the array of [pulsars](#) used in tempo2

12.40.3.11 char NEWFIT =0

global boolean used to enable new fit.

Warning

this will be removed in future.

12.40.3.12 double NFIT

12.40.3.13 char TEMPO2_ENVIRON[MAX_STRLEN] ="TEMPO2"

TEMPO2 environment variable

12.40.3.14 char TEMPO2_ERROR[MAX_STRLEN] =""

TEMPO2 error messages

12.40.3.15 char tempo2_plug_path[32][MAX_STRLEN]

paths to search for plugins

12.40.3.16 int tempo2_plug_path_len =0

12.40.3.17 char tempo2MachineType[MAX_FILELEN] = ""

12.40.3.18 double UPW

12.40.3.19 int veryFast = 0

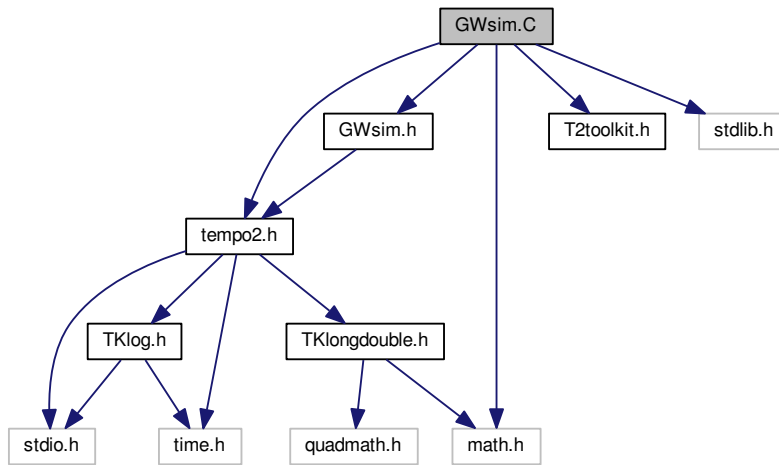
Global to run the code fast

12.40.3.20 double WNLEVEL

12.41 GWsim.C File Reference

```
#include "tempo2.h"
#include "GWsim.h"
#include "T2toolkit.h"
#include <math.h>
#include <stdlib.h>
```

Include dependency graph for GWsim.C:



Functions

- void `setupGW` (`gwSrc *gw`)
- void `matrixMult` (`longdouble m1[3][3]`, `longdouble m2[3][3]`, `longdouble out[3][3]`)
- `longdouble dotProduct` (`longdouble *m1`, `longdouble *m2`)
- void `GWbackground` (`gwSrc *gw`, `int numberGW`, `long *idum`, `longdouble flo`, `longdouble fhi`, `double gwAmp`, `double alpha`, `int loglin`)
- `longdouble calculateResidualGW` (`longdouble *kp`, `gwSrc *gw`, `longdouble time`, `longdouble dist`)
- void `setupPulsar_GWsim` (`longdouble ra_p`, `longdouble dec_p`, `longdouble *kp`)
- int `GWbackground_read` (`gwSrc *gw`, `FILE *file`, `int ireal`)
- void `GWbackground_write` (`gwSrc *gw`, `FILE *file`, `int ngw`, `int ireal`)
- `longdouble eccRes` (`pulsar *psr`, `int i`, `int *coalesceFlag`, `double *prev_p`, `double *prev_e`, `double *prev_a`, `double *prev_epoch`, `double *prev_theta`)
- `longdouble eccResWithEnergy` (`pulsar *psr`, `int i`, `int *coalesceFlag`, `double *prev_p`, `double *prev_e`, `double *prev_a`, `double *prev_epoch`, `double *prev_theta`, `float *eOut`)
- double `Rs` (`double m1`)
- double `dadt` (`double ec`, `double a`, `double m1`, `double m2`)
- double `dedt` (`double ec`, `double a`, `double m1`, `double m2`)
- double `dttd` (`double ec`, `double t`, `double p`)
- double `Fe` (`double ec`)
- double `psrangle` (`double centre_long`, `double centre_lat`, `double psr_long`, `double psr_lat`)
- double `sphharm` (`int l`, `int m`, `double x`)
- double `Findphi` (`double prob`, `double amp`, `double phase`)
- void `setupgeneralGW` (`gwgeneralSrc *gw`)
- void `GWgeneralbackground` (`gwgeneralSrc *gw`, `int *numberGW`, `long *idum`, `longdouble flo`, `longdouble fhi`, `gwgenSpec gwAmps`, `int loglin`)
- `longdouble calculateResidualgeneralGW` (`longdouble *kp`, `gwgeneralSrc *gw`, `longdouble time`, `longdouble dist`)
- void `GWgeneralanisotropicbackground` (`gwgeneralSrc *gw`, `int *numberGW`, `long *idum`, `longdouble flo`, `longdouble fhi`, `gwgenSpec gwAmps`, `int loglin`, `double ***harmlist`, `int nharms`)
- void `GWanisotropicbackground` (`gwSrc *gw`, `int numberGW`, `long *idum`, `longdouble flo`, `longdouble fhi`, `double gwAmp`, `double alpha`, `int loglin`, `double **harmlist`, `int nharms`)

- void `GWdipolebackground` (`gwSrc *gw`, `int numberGW`, `long *idum`, `longdouble flo`, `longdouble fhi`, `double gwAmp`, `double alpha`, `int loglin`, `double *dipoleamps`)
- int `GWgeneralbackground_read` (`gwgeneralSrc *gw`, `FILE *file`, `int ireal`)
- void `GWgeneralbackground_write` (`gwgeneralSrc *gw`, `FILE *file`, `int ngw`, `int ireal`)

Variables

- int `gwsim_Ngrid` =1000

12.41.1 Function Documentation

- 12.41.1.1 `longdouble calculateResidualgeneralGW (longdouble * kp, gwgeneralSrc * gw, longdouble time, longdouble dist)`
- 12.41.1.2 `longdouble calculateResidualGW (longdouble * kp, gwSrc * gw, longdouble time, longdouble dist)`
- 12.41.1.3 `double dadt (double ec, double a, double m1, double m2)`
- 12.41.1.4 `double dedt (double ec, double a, double m1, double m2)`
- 12.41.1.5 `longdouble dotProduct (longdouble * m1, longdouble * m2)`
- 12.41.1.6 `double dtdt (double ec, double t, double p)`
- 12.41.1.7 `longdouble eccRes (pulsar * psr, int i, int * coalesceFlag, double * prev_p, double * prev_e, double * prev_a, double * prev_epoch, double * prev_theta)`
- 12.41.1.8 `longdouble eccResWithEnergy (pulsar * psr, int i, int * coalesceFlag, double * prev_p, double * prev_e, double * prev_a, double * prev_epoch, double * prev_theta, float * eOut)`
- 12.41.1.9 `double Fe (double ec)`
- 12.41.1.10 `double Findphi (double prob, double amp, double phase)`
- 12.41.1.11 `void GWanisotropicbackground (gwSrc * gw, int numberGW, long * idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double ** harmlist, int nharms)`
- 12.41.1.12 `void GWbackground (gwSrc * gw, int numberGW, long * idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin)`
- 12.41.1.13 `int GWbackground_read (gwSrc * gw, FILE * file, int ireal)`
- 12.41.1.14 `void GWbackground_write (gwSrc * gw, FILE * file, int ngw, int ireal)`
- 12.41.1.15 `void GWdipolebackground (gwSrc * gw, int numberGW, long * idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double * dipoleamps)`
- 12.41.1.16 `void GWgeneralanisotropicbackground (gwgeneralSrc * gw, int * numberGW, long * idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin, double *** harmlist, int * nharms)`
- 12.41.1.17 `void GWgeneralbackground (gwgeneralSrc * gw, int * numberGW, long * idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin)`
- 12.41.1.18 `int GWgeneralbackground_read (gwgeneralSrc * gw, FILE * file, int ireal)`

- 12.41.1.19 void GWgeneralbackground_write (gwgeneralSrc * gw, FILE * file, int ngw, int ired)
- 12.41.1.20 void matrixMult (longdouble m1[3][3], longdouble m2[3][3], longdouble out[3][3])
- 12.41.1.21 double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)
- 12.41.1.22 double Rs (double m1)
- 12.41.1.23 void setupgeneralGW (gwgeneralSrc * gw)
- 12.41.1.24 void setupGW (gwSrc * gw)
- 12.41.1.25 void setupPulsar_GWsim (longdouble ra_p, longdouble dec_p, longdouble * kp)
- 12.41.1.26 double sphharm (int l, int m, double x)

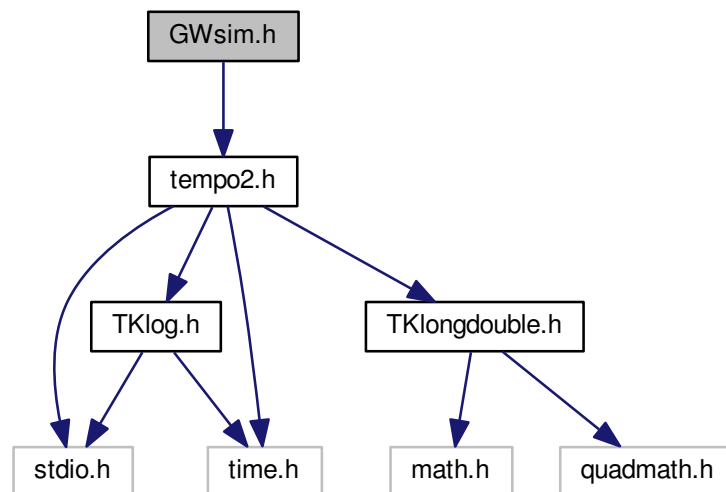
12.41.2 Variable Documentation

- 12.41.2.1 int gwsim_Ngrid =1000

12.42 GWsim.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for GWsim.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [gwSrc](#)
- struct [gwgeneralSrc](#)
- struct [gwgenSpec](#)

Typedefs

- typedef struct [gwSrc](#) [gwSrc](#)
- typedef struct [gwgeneralSrc](#) [gwgeneralSrc](#)
- typedef struct [gwgenSpec](#) [gwgenSpec](#)

Functions

- double [Fe](#) (double ec)
- double [dadt](#) (double ec, double a, double m1, double [m2](#))
- double [dedt](#) (double ec, double a, double m1, double [m2](#))
- double [dtdt](#) (double ec, double t, double p)
- double [Rs](#) (double m1)
- [longdouble](#) [eccRes](#) ([pulsar](#) *psr, int i, int *coalesceFlag, double *prev_p, double *prev_e, double *prev_a, double *prev_epoch, double *prev_theta)
- [longdouble](#) [eccResWithEnergy](#) ([pulsar](#) *psr, int i, int *coalesceFlag, double *prev_p, double *prev_e, double *prev_a, double *prev_epoch, double *prev_theta, float *eOut)
- void [setupGW](#) ([gwSrc](#) *gw)
- void [matrixMult](#) ([longdouble](#) m1[3][3], [longdouble](#) m2[3][3], [longdouble](#) out[3][3])
- [longdouble](#) [dotProduct](#) ([longdouble](#) *m1, [longdouble](#) *m2)
- void [GWbackground](#) ([gwSrc](#) *gw, int numberGW, long *idum, [longdouble](#) flo, [longdouble](#) fhi, double gwAmp, double [alpha](#), int loglin)
- [longdouble](#) [calculateResidualGW](#) ([longdouble](#) *kp, [gwSrc](#) *gw, [longdouble](#) time, [longdouble](#) dist)
- void [setupPulsar_GWsim](#) ([longdouble](#) ra_p, [longdouble](#) dec_p, [longdouble](#) *kp)
- int [GWbackground_read](#) ([gwSrc](#) *gw, FILE *file, int ireal)
- void [GWbackground_write](#) ([gwSrc](#) *gw, FILE *file, int ngw, int ireal)
- double [psrangle](#) (double centre_long, double centre_lat, double psr_long, double psr_lat)
- double [sphharm](#) (int l, int m, double x)
- double [Findphi](#) (double prob, double amp, double phase)
- void [setupgeneralGW](#) ([gwgeneralSrc](#) *gw)
- void [GWgeneralbackground](#) ([gwgeneralSrc](#) *gw, int *numberGW, long *idum, [longdouble](#) flo, [longdouble](#) fhi, [gwgenSpec](#) gwAmps, int loglin)
- void [GWgeneralanisotropicbackground](#) ([gwgeneralSrc](#) *gw, int *numberGW, long *idum, [longdouble](#) flo, [longdouble](#) fhi, [gwgenSpec](#) gwAmps, int loglin, double ***harmList, int *nharms)
- void [GWanisotropicbackground](#) ([gwSrc](#) *gw, int numberGW, long *idum, [longdouble](#) flo, [longdouble](#) fhi, double gwAmp, double [alpha](#), int loglin, double **harmList, int nharms)
- void [GWdipolebackground](#) ([gwSrc](#) *gw, int numberGW, long *idum, [longdouble](#) flo, [longdouble](#) fhi, double gwAmp, double [alpha](#), int loglin, double *dipoleamps)
- [longdouble](#) [calculateResidualgeneralGW](#) ([longdouble](#) *kp, [gwgeneralSrc](#) *gw, [longdouble](#) time, [longdouble](#) dist)
- int [GWgeneralbackground_read](#) ([gwgeneralSrc](#) *gw, FILE *file, int ireal)
- void [GWgeneralbackground_write](#) ([gwgeneralSrc](#) *gw, FILE *file, int ngw, int ireal)

12.42.1 Typedef Documentation

12.42.1.1 typedef struct gwgeneralSrc gwgeneralSrc

12.42.1.2 typedef struct gwgenSpec gwgenSpec

12.42.1.3 typedef struct gwSrc gwSrc

12.42.2 Function Documentation

12.42.2.1 longdouble calculateResidualgeneralGW (longdouble * *kp*, gwgeneralSrc * *gw*, longdouble *time*, longdouble *dist*)

12.42.2.2 longdouble calculateResidualGW (longdouble * *kp*, gwSrc * *gw*, longdouble *time*, longdouble *dist*)

12.42.2.3 double dadt (double *ec*, double *a*, double *m1*, double *m2*)

12.42.2.4 double dedt (double *ec*, double *a*, double *m1*, double *m2*)

12.42.2.5 longdouble dotProduct (longdouble * *m1*, longdouble * *m2*)

12.42.2.6 double dtdt (double *ec*, double *t*, double *p*)

12.42.2.7 longdouble eccRes (pulsar * *psr*, int *i*, int * *coalesceFlag*, double * *prev_p*, double * *prev_e*, double * *prev_a*, double * *prev_epoch*, double * *prev_theta*)

12.42.2.8 longdouble eccResWithEnergy (pulsar * *psr*, int *i*, int * *coalesceFlag*, double * *prev_p*, double * *prev_e*, double * *prev_a*, double * *prev_epoch*, double * *prev_theta*, float * *eOut*)

12.42.2.9 double Fe (double *ec*)

12.42.2.10 double Findphi (double *prob*, double *amp*, double *phase*)

12.42.2.11 void GWanisotropicbackground (gwSrc * *gw*, int *numberGW*, long * *idum*, longdouble *flo*, longdouble *fhi*, double *gwAmp*, double *alpha*, int *loglin*, double ** *harmlist*, int *nharms*)

12.42.2.12 void GWbackground (gwSrc * *gw*, int *numberGW*, long * *idum*, longdouble *flo*, longdouble *fhi*, double *gwAmp*, double *alpha*, int *loglin*)

12.42.2.13 int GWbackground_read (gwSrc * *gw*, FILE * *file*, int *ireal*)

12.42.2.14 void GWbackground_write (gwSrc * *gw*, FILE * *file*, int *ngw*, int *ireal*)

12.42.2.15 void GWdipolebackground (gwSrc * *gw*, int *numberGW*, long * *idum*, longdouble *flo*, longdouble *fhi*, double *gwAmp*, double *alpha*, int *loglin*, double * *dipoleamps*)

12.42.2.16 void GWgeneralanisotropicbackground (gwgeneralSrc * *gw*, int * *numberGW*, long * *idum*, longdouble *flo*, longdouble *fhi*, gwgenSpec *gwAmps*, int *loglin*, double *** *harmlist*, int * *nharms*)

12.42.2.17 void GWgeneralbackground (gwgeneralSrc * *gw*, int * *numberGW*, long * *idum*, longdouble *flo*, longdouble *fhi*, gwgenSpec *gwAmps*, int *loglin*)

12.42.2.18 int GWgeneralbackground_read (gwgeneralSrc * *gw*, FILE * *file*, int *ireal*)

12.42.2.19 void GWgeneralbackground_write (gwgeneralSrc * *gw*, FILE * *file*, int *ngw*, int *ireal*)

12.42.2.20 void matrixMult (longdouble m1[3][3], longdouble m2[3][3], longdouble out[3][3])

12.42.2.21 double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)

12.42.2.22 double Rs (double m1)

12.42.2.23 void setupgeneralGW (gwgeneralSrc * gw)

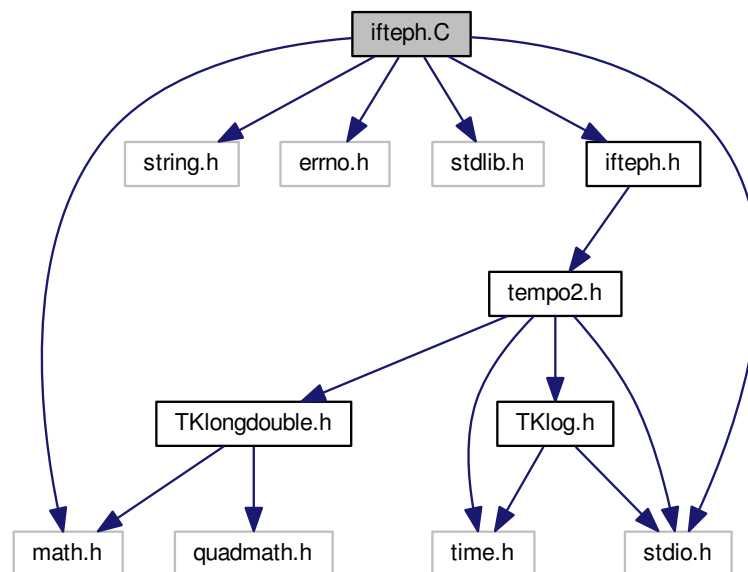
12.42.2.24 void setupGW (gwSrc * gw)

12.42.2.25 void setupPulsar_GWsim (longdouble ra_p, longdouble dec_p, longdouble * kp)

12.42.2.26 double sphharm (int l, int m, double x)

12.43 ifteph.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <errno.h>
#include <stdlib.h>
#include <math.h>
#include "ifteph.h"
Include dependency graph for ifteph.C:
```



Classes

- struct [IFTE_interpolation_info](#)
- struct [IFTEphemeris](#)

Functions

- void [IFTswap4](#) (char *word)
- void [IFTswapInt](#) (int *word)
- void [IFTswapInts](#) (int *word, int n)
- void [IFTswap8](#) (char *dword)
- void [IFTswapDouble](#) (double *dbl)
- void [IFTswap8N](#) (char *dwords, int n)
- void [IFTswapDoubles](#) (double *dbl, int N)
- void [IFTE_init](#) (const char *fname)
- void [IFTE_close_file](#) ()
- void [IFTE_get_Vals](#) (double JDeph0, double JDeph1, int kind, double *res)
- void [IFTE_get_DeltaT_DeltaTDot](#) (double Teph0, double Teph1, double *DeltaT, double *DeltaTDot)
- double [IFTE_DeltaT](#) (double Teph0, double Teph1)
- double [IFTE_DeltaTDot](#) (double Teph0, double Teph1)
- void [IFTE_get_vE_vEDot](#) (double Teph0, double Teph1, double *vE, double *vEDot)
- void [IFTE_get_vE](#) (double Teph0, double Teph1, double *vE)
- void [IFTE_get_vEDot](#) (double Teph0, double Teph1, double *vEDot)

12.43.1 Function Documentation

12.43.1.1 void [IFTE_close_file](#) ()

12.43.1.2 double [IFTE_DeltaT](#) (double *Teph0*, double *Teph1*)

12.43.1.3 double [IFTE_DeltaTDot](#) (double *Teph0*, double *Teph1*)

12.43.1.4 void [IFTE_get_DeltaT_DeltaTDot](#) (double *Teph0*, double *Teph1*, double * *DeltaT*, double * *DeltaTDot*)

12.43.1.5 void [IFTE_get_Vals](#) (double *JDeph0*, double *JDeph1*, int *kind*, double * *res*)

12.43.1.6 void [IFTE_get_vE](#) (double *Teph0*, double *Teph1*, double * *vE*)

12.43.1.7 void [IFTE_get_vE_vEDot](#) (double *Teph0*, double *Teph1*, double * *vE*, double * *vEDot*)

12.43.1.8 void [IFTE_get_vEDot](#) (double *Teph0*, double *Teph1*, double * *vEDot*)

12.43.1.9 void [IFTE_init](#) (const char * *fname*)

12.43.1.10 void [IFTswap4](#) (char * *word*)

12.43.1.11 void [IFTswap8](#) (char * *dword*)

12.43.1.12 void [IFTswap8N](#) (char * *dwords*, int *n*)

12.43.1.13 void [IFTswapDouble](#) (double * *dbl*)

12.43.1.14 void [IFTswapDoubles](#) (double * *dbl*, int *N*)

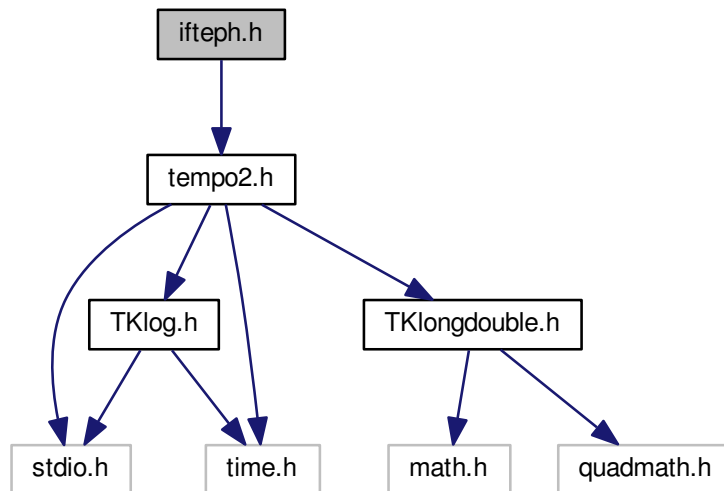
12.43.1.15 void [IFTswapInt](#) (int * *word*)

12.43.1.16 void [IFTswapInts](#) (int * *word*, int *n*)

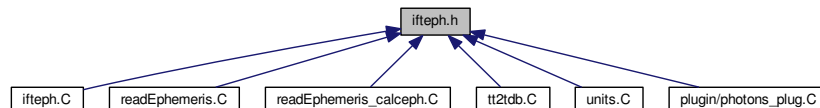
12.44 ifteph.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for ifteph.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define IFTE_JD0` 2443144.5003725 /* Epoch of TCB, TCG and TT */
- `#define IFTE_MJD0` 43144.0003725
- `#define IFTE_TEPH0` -65.564518e-6
- `#define IFTE_LC` 1.48082686742e-8
- `#define IFTE_KM1` 1.55051979176e-8
- `#define IFTE_K` (((longdouble)1.0) + ((longdouble)IFTE_KM1)) /* needs quad precision */

Functions

- void `IFTE_init` (const char *fname)
- void `IFTE_get_DeltaT_DeltaTDot` (double Teph0, double Teph1, double *DeltaT, double *DeltaTDot)
- double `IFTE_DeltaT` (double Teph0, double Teph1)
- double `IFTE_DeltaTDot` (double Teph0, double Teph1)
- void `IFTE_close_file` ()

- void `IFTE_get_vE_vEDot` (double *Teph0*, double *Teph1*, double **ve*, double **vEDot*)
- void `IFTE_get_vE` (double *Teph0*, double *Teph1*, double **vE*)
- void `IFTE_get_vEDot` (double *Teph0*, double *Teph1*, double **vEDot*)

12.44.1 Macro Definition Documentation

12.44.1.1 `#define IFTE_JD0 2443144.5003725 /* Epoch of TCB, TCG and TT */`

12.44.1.2 `#define IFTE_K (((longdouble)1.0) + ((longdouble)IFTE_KM1)) /* needs quad precision */`

12.44.1.3 `#define IFTE_KM1 1.55051979176e-8`

12.44.1.4 `#define IFTE_LC 1.48082686742e-8`

12.44.1.5 `#define IFTE_MJD0 43144.0003725`

12.44.1.6 `#define IFTE_TEPH0 -65.564518e-6`

12.44.2 Function Documentation

12.44.2.1 void `IFTE_close_file` ()

12.44.2.2 double `IFTE_DeltaT` (double *Teph0*, double *Teph1*)

12.44.2.3 double `IFTE_DeltaTDot` (double *Teph0*, double *Teph1*)

12.44.2.4 void `IFTE_get_DeltaT_DeltaTDot` (double *Teph0*, double *Teph1*, double * *DeltaT*, double * *DeltaTDot*)

12.44.2.5 void `IFTE_get_vE` (double *Teph0*, double *Teph1*, double * *vE*)

12.44.2.6 void `IFTE_get_vE_vEDot` (double *Teph0*, double *Teph1*, double * *ve*, double * *vEDot*)

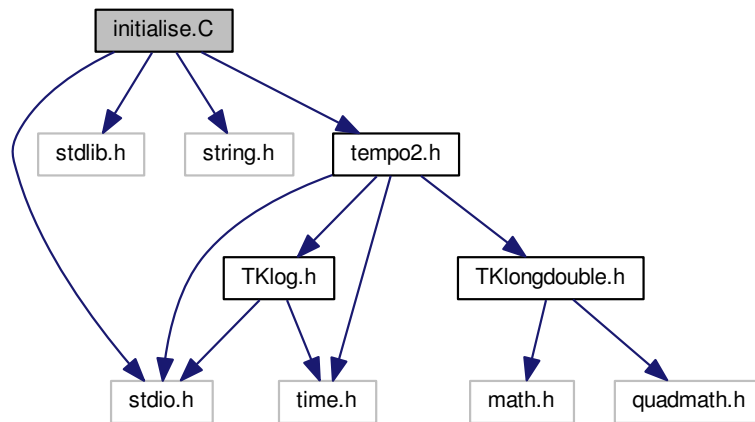
12.44.2.7 void `IFTE_get_vEDot` (double *Teph0*, double *Teph1*, double * *vEDot*)

12.44.2.8 void `IFTE_init` (const char * *fname*)

12.45 initialise.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"
```


Include dependency graph for initialise.C:



Functions

- void [initialise](#) ([pulsar](#) *psr, int noWarnings)
- void [initialiseOne](#) ([pulsar](#) *psr, int noWarnings, int fullSetup)
- void [allocateMemory](#) ([pulsar](#) *psr, int realloc)
- void [destroyOne](#) ([pulsar](#) *psr)
- void [destroyMemory](#) ([pulsar](#) *psr)

12.45.1 Function Documentation

12.45.1.1 void [allocateMemory](#) ([pulsar](#) * *psr*, int *realloc*)

12.45.1.2 void [destroyMemory](#) ([pulsar](#) * *psr*)

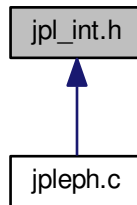
12.45.1.3 void [destroyOne](#) ([pulsar](#) * *psr*)

12.45.1.4 void [initialise](#) ([pulsar](#) * *psr*, int *noWarnings*)

12.45.1.5 void [initialiseOne](#) ([pulsar](#) * *psr*, int *noWarnings*, int *fullSetup*)

12.46 jpl_int.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- struct [jpl_eph_data](#)
- struct [interpolation_info](#)

Macros

- `#define` [MAX_KERNEL_SIZE](#) 2036
- `#define` [JPL_HEADER_SIZE](#) (5 * sizeof(double) + 41 * sizeof([JPLlong](#)))

Typedefs

- typedef unsigned int [JPLlong](#)

12.46.1 Macro Definition Documentation

12.46.1.1 `#define` [JPL_HEADER_SIZE](#) (5 * sizeof(double) + 41 * sizeof([JPLlong](#)))

12.46.1.2 `#define` [MAX_KERNEL_SIZE](#) 2036

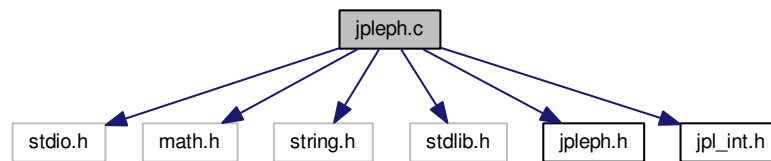
12.46.2 Typedef Documentation

12.46.2.1 typedef unsigned int [JPLlong](#)

12.47 jpleph.c File Reference

```
#include <stdio.h>
#include <math.h>
#include <string.h>
#include <stdlib.h>
#include "jpleph.h"
#include "jpl_int.h"
```

Include dependency graph for jpleph.c:



Macros

- `#define TRUE 1`
- `#define FALSE 0`
- `#define SWAP_MACRO(A, B, TEMP) { TEMP = A; A = B; B = TEMP; }`

Functions

- `double DLL_FUNC jpl_get_double (const void *ephem, const int value)`
- `double DLL_FUNC jpl_get_long (const void *ephem, const int value)`
- `int DLL_FUNC jpl_pleph (void *ephem, const double et[2], const int ntarg, const int ncent, double rrd[], const int calc_velocity)`
- `int DLL_FUNC jpl_state (void *ephem, const double et[2], const int list[12], double pv[][6], double nut[4], const int bary)`
- `void *DLL_FUNC jpl_init_ephemeris (const char *ephemeris_filename, char nam[][6], double *val)`
- `void DLL_FUNC jpl_close_ephemeris (void *ephem)`

12.47.1 Macro Definition Documentation

12.47.1.1 `#define FALSE 0`

12.47.1.2 `#define SWAP_MACRO(A, B, TEMP) { TEMP = A; A = B; B = TEMP; }`

12.47.1.3 `#define TRUE 1`

12.47.2 Function Documentation

12.47.2.1 `void DLL_FUNC jpl_close_ephemeris (void * ephem)`

12.47.2.2 `double DLL_FUNC jpl_get_double (const void * ephem, const int value)`

12.47.2.3 `double DLL_FUNC jpl_get_long (const void * ephem, const int value)`

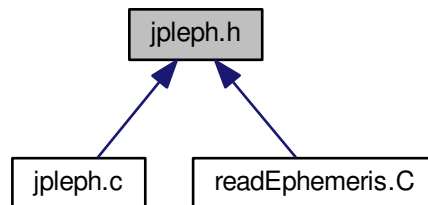
12.47.2.4 `void* DLL_FUNC jpl_init_ephemeris (const char * ephemeris_filename, char nam[][6], double * val)`

12.47.2.5 `int DLL_FUNC jpl_pleph (void * ephem, const double et[2], const int ntarg, const int ncent, double rrd[], const int calc_velocity)`

12.47.2.6 `int DLL_FUNC jpl_state (void * ephem, const double et[2], const int list[12], double pv[][6], double nut[4], const int bary)`

12.48 jpleph.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define DLL_FUNC`
- `#define JPL_EPHEM_START_JD 0`
- `#define JPL_EPHEM_END_JD 8`
- `#define JPL_EPHEM_STEP 16`
- `#define JPL_EPHEM_N_CONSTANTS 24`
- `#define JPL_EPHEM_AU_IN_KM 28`
- `#define JPL_EPHEM_EARTH_MOON_RATIO 36`
- `#define JPL_EPHEM_EPHEMERIS_VERSION 200`
- `#define JPL_EPHEM_KERNEL_SIZE 204`
- `#define JPL_EPHEM_KERNEL_RECORD_SIZE 208`
- `#define JPL_EPHEM_KERNEL_NCOEFF 212`
- `#define JPL_EPHEM_KERNEL_SWAP_BYTES 216`

Functions

- `void *DLL_FUNC jpl_init_ephemeris` (const char *ephemeris_filename, char nam[][6], double *val)
- `void DLL_FUNC jpl_close_ephemeris` (void *ephem)
- `int DLL_FUNC jpl_state` (void *ephem, const double et[2], const int list[12], double pv[][6], double nut[4], const int bary)
- `int DLL_FUNC jpl_pleph` (void *ephem, const double et[2], const int ntarg, const int ncent, double rrd[], const int calc_velocity)
- `double DLL_FUNC jpl_get_double` (const void *ephem, const int value)
- `double DLL_FUNC jpl_get_long` (const void *ephem, const int value)
- `int DLL_FUNC make_sub_ephem` (const void *ephem, const char *sub_filename, const double start_jd, const double end_jd)

12.48.1 Macro Definition Documentation

12.48.1.1 `#define DLL_FUNC`

12.48.1.2 `#define JPL_EPHEM_AU_IN_KM 28`

12.48.1.3 `#define JPL_EPHEM_EARTH_MOON_RATIO 36`

12.48.1.4 `#define JPL_EPHEM_END_JD 8`

12.48.1.5 `#define JPL_EPHEM_EPHEMERIS_VERSION 200`

12.48.1.6 `#define JPL_EPHEM_KERNEL_NCOEFF 212`

12.48.1.7 `#define JPL_EPHEM_KERNEL_RECORD_SIZE 208`

12.48.1.8 `#define JPL_EPHEM_KERNEL_SIZE 204`

12.48.1.9 `#define JPL_EPHEM_KERNEL_SWAP_BYTES 216`

12.48.1.10 `#define JPL_EPHEM_N_CONSTANTS 24`

12.48.1.11 `#define JPL_EPHEM_START_JD 0`

12.48.1.12 `#define JPL_EPHEM_STEP 16`

12.48.2 Function Documentation

12.48.2.1 `void DLL_FUNC jpl_close_ephemeris (void * ephem)`

12.48.2.2 `double DLL_FUNC jpl_get_double (const void * ephem, const int value)`

12.48.2.3 `double DLL_FUNC jpl_get_long (const void * ephem, const int value)`

12.48.2.4 `void* DLL_FUNC jpl_init_ephemeris (const char * ephemeris_filename, char nam[[6], double * val)`

12.48.2.5 `int DLL_FUNC jpl_pleph (void * ephem, const double et[2], const int ntarg, const int ncent, double rrd[], const int calc_velocity)`

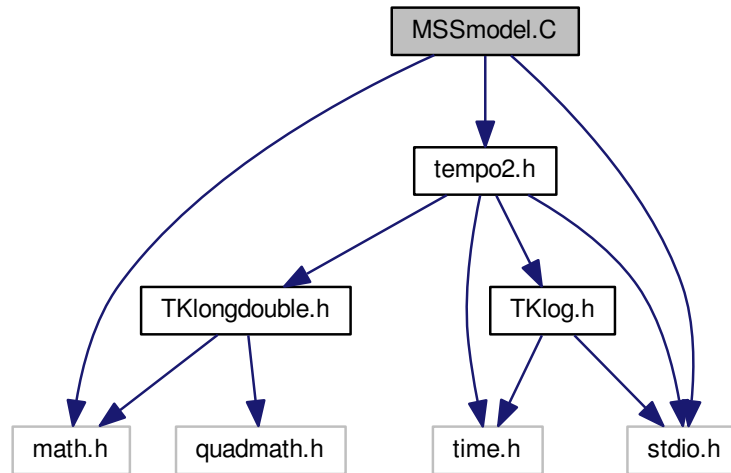
12.48.2.6 `int DLL_FUNC jpl_state (void * ephem, const double et[2], const int list[12], double pv[[6], double nut[4], const int bary)`

12.48.2.7 `int DLL_FUNC make_sub_ephem (const void * ephem, const char * sub_filename, const double start_jd, const double end_jd)`

12.49 MSSmodel.C File Reference

```
#include <stdio.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for MSSmodel.C:



Functions

- double [MSSmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateMSS](#) ([pulsar](#) *psr, double val, double err, int pos)

12.49.1 Function Documentation

12.49.1.1 double [MSSmodel](#) ([pulsar](#) * *psr*, int *p*, int *obs*, int *param*)

`RAD/pow(365.25*SECDAY,2.);`

12.49.1.2 void [updateMSS](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*)

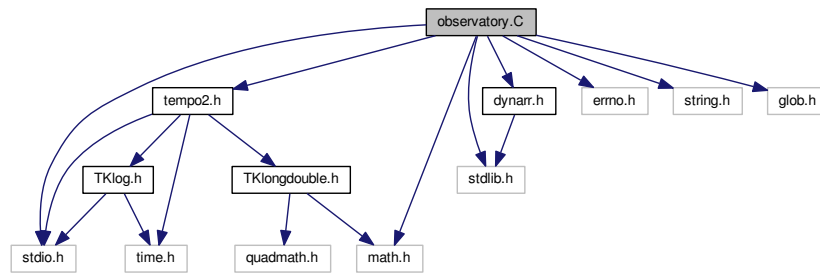
12.50 observatory.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <glob.h>
#include <math.h>
#include "tempo2.h"
#include "dynarr.h"

```

Include dependency graph for observatory.C:



Classes

- struct [ObservatoryAliasList](#)

Macros

- `#define` [GRS80_A](#) 6378137.0 /* semi-major axis (m) */
- `#define` [GRS80_F](#) 1.0/298.257222101 /* flattening */

Functions

- void [GRS80_to_ITRF](#) ([observatory](#) *obs)
- void [ITRF_to_GRS80](#) ([observatory](#) *obs)
- double [fang](#) (int i, double f)
- void [readObservatoryFile](#) (char *fname)
- void [readAliases](#) (char *fname)
- void [initObservatories](#) ()
- void [lookup_observatory_alias](#) (char *incode, char *outcode)
- [observatory](#) * [getObservatory](#) (char *code)

12.50.1 Macro Definition Documentation

12.50.1.1 `#define` [GRS80_A](#) 6378137.0 /* semi-major axis (m) */

12.50.1.2 `#define` [GRS80_F](#) 1.0/298.257222101 /* flattening */

12.50.2 Function Documentation

12.50.2.1 double [fang](#) (int *i*, double *f*)

12.50.2.2 [observatory](#)* [getObservatory](#) (char * *code*)

12.50.2.3 void [GRS80_to_ITRF](#) ([observatory](#) * *obs*)

12.50.2.4 void [initObservatories](#) ()

12.50.2.5 void [ITRF_to_GRS80](#) ([observatory](#) * *obs*)

12.50.2.6 void lookup_observatory_alias (char * *incode*, char * *outcode*)

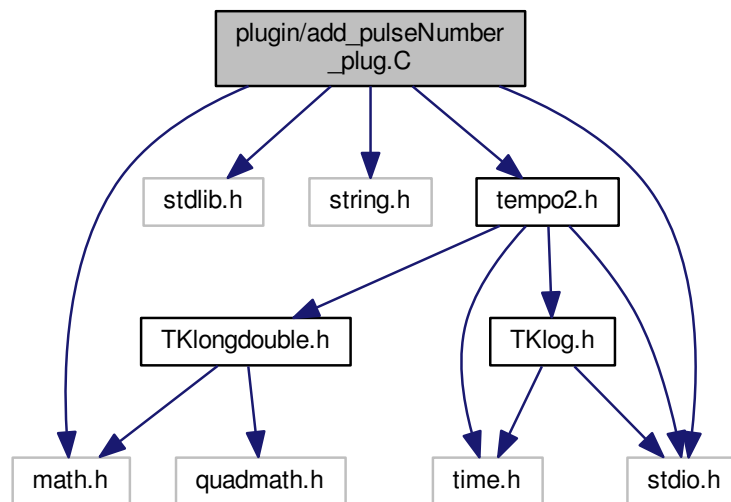
12.50.2.7 void readAliases (char * *fname*)

12.50.2.8 void readObservatoryFile (char * *fname*)

12.51 plugin/add_pulseNumber_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for add_pulseNumber_plug.C:



Functions

- int [tempoOutput](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int *npsr*)

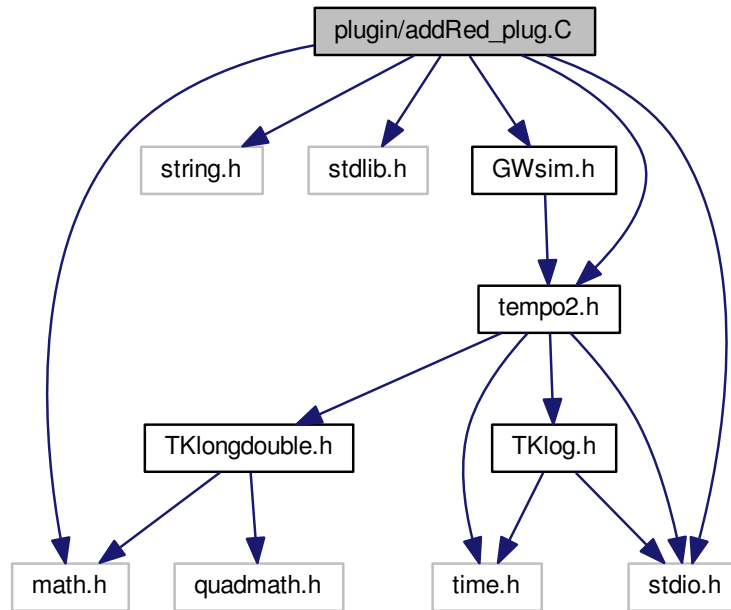
12.51.1 Function Documentation

12.51.1.1 int [tempoOutput](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int *npsr*)

12.52 plugin/addRed_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
```


Include dependency graph for addRed_plug.C:



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.52.1 Function Documentation

12.52.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

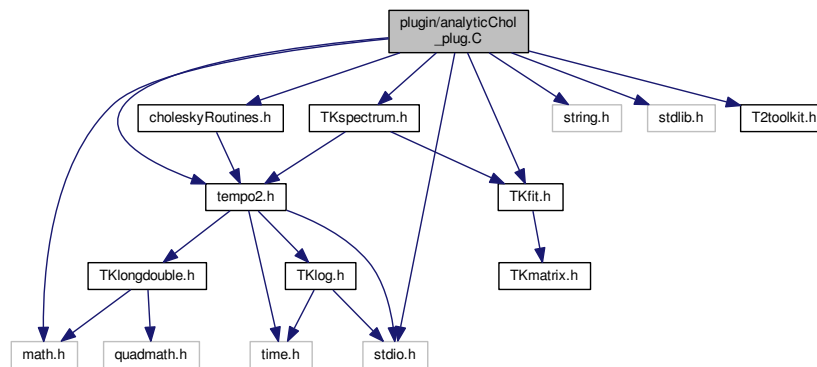
12.52.1.2 void [help](#) ()

12.52.2 Variable Documentation

12.52.2.1 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.53 plugin/analyticChol_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKspectrum.h"
#include "TKfit.h"
#include "T2toolkit.h"
#include "choleskyRoutines.h"
Include dependency graph for analyticChol_plug.C:
```



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.53.1 Function Documentation

12.53.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.53.1.2 void [help](#) ()

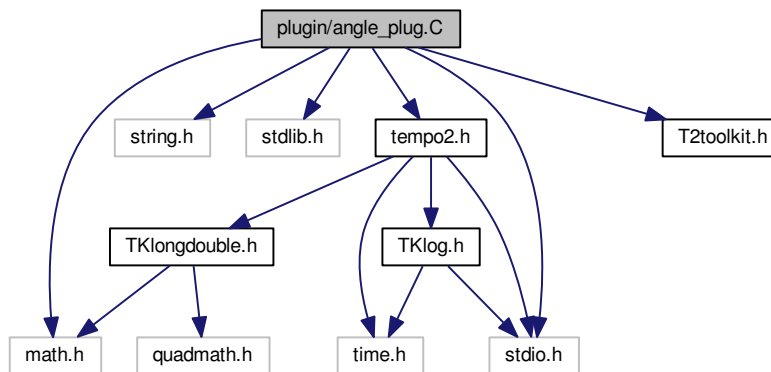
12.53.2 Variable Documentation

12.53.2.1 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.54 plugin/angle_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
```

Include dependency graph for angle_plug.C:



Functions

- double [psrangle](#) (double centre_long, double centre_lat, double psr_long, double psr_lat)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

12.54.1 Function Documentation

12.54.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

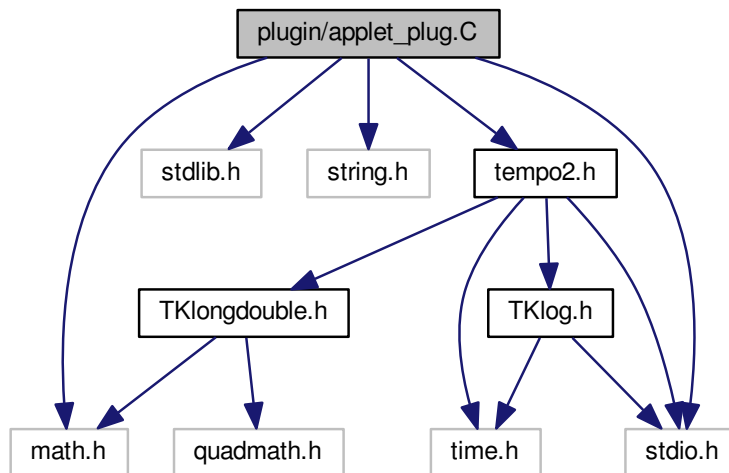
12.54.1.2 void [help](#) ()

12.54.1.3 double [psrangle](#) (double *centre_long*, double *centre_lat*, double *psr_long*, double *psr_lat*)

12.55 plugin/applet_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for applet_plug.C:



Functions

- int [nint_derived](#) (double x)
- int [rnd8](#) (double rval, double rerr, int ifac, char *cval, int *lv, char *cerr, int *le, char *msg)
- void [parseLine](#) (pulsar *psr, char *line, double *errMult, char *null, char *format, char *dformat, int *rad, FILE *fout)
- double [fortranMod](#) (double a, double p)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], pulsar *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.55.1 Function Documentation

12.55.1.1 double [fortranMod](#) (double *a*, double *p*)

12.55.1.2 int [graphicalInterface](#) (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)

12.55.1.3 void [help](#) ()

12.55.1.4 int [nint_derived](#) (double *x*)

12.55.1.5 void [parseLine](#) (pulsar * *psr*, char * *line*, double * *errMult*, char * *null*, char * *format*, char * *dformat*, int * *rad*, FILE * *fout*)

12.55.1.6 int [rnd8](#) (double *rval*, double *rerr*, int *ifac*, char * *cval*, int * *lv*, char * *cerr*, int * *le*, char * *msg*)

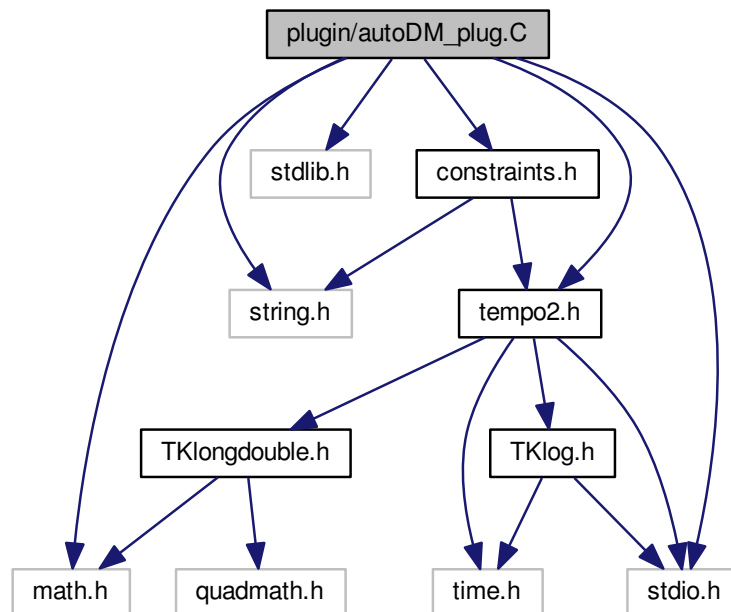
12.55.2 Variable Documentation

12.55.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.56 plugin/autoDM_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "constraints.h"
```

Include dependency graph for autoDM_plug.C:



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.56.1 Function Documentation

12.56.1.1 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.56.1.2 void help ()

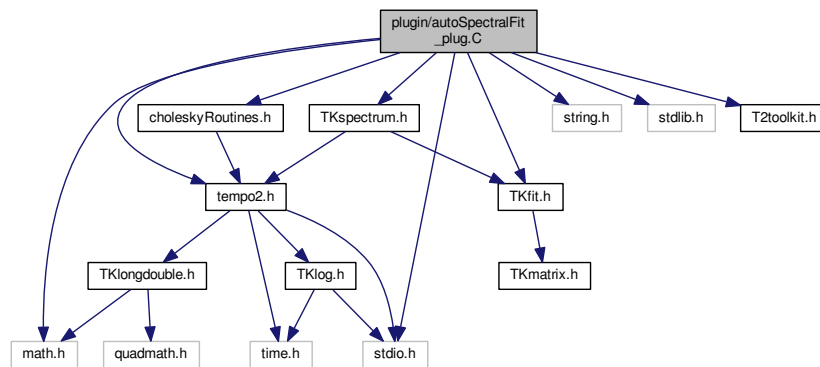
12.56.2 Variable Documentation

12.56.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.57 plugin/autoSpectralFit_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <TKspectrum.h>
#include "T2toolkit.h"
#include "TKfit.h"
#include "choleskyRoutines.h"
```

Include dependency graph for autoSpectralFit_plug.C:



Macros

- #define [MAX_FREQ](#) 10000

Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

12.57.1 Macro Definition Documentation

12.57.1.1 #define MAX_FREQ 10000

12.57.2 Function Documentation

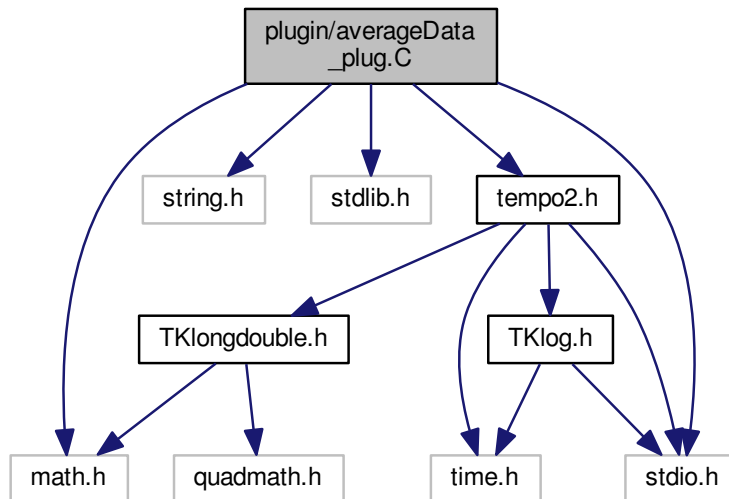
12.57.2.1 int graphicalInterface (int argc, char * argv[], [pulsar](#) * psr, int * npsr)

12.57.2.2 void help ()

12.58 plugin/averageData_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for averageData_plug.C:



Macros

- `#define` [MAX_TIMES](#) 1000

Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.58.1 Macro Definition Documentation

12.58.1.1 `#define` [MAX_TIMES](#) 1000

12.58.2 Function Documentation

12.58.2.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.58.2.2 void help ()

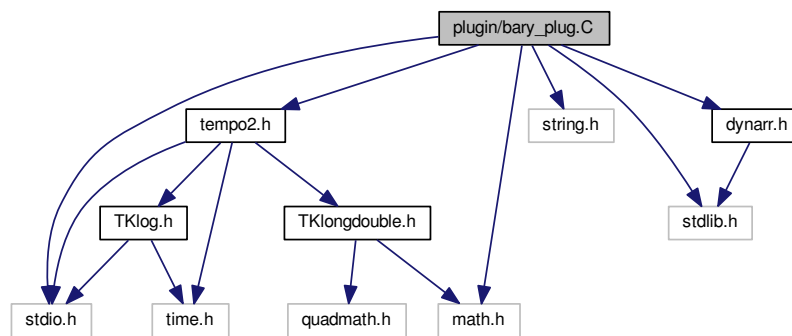
12.58.3 Variable Documentation

12.58.3.1 const char* plugVersionCheck = TEMPO2_h_VER

12.59 plugin/bary_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "dynarr.h"
```

Include dependency graph for bary_plug.C:



Macros

- #define [GRS80_A](#) 6378137.0 /* semi-major axis (m) */
- #define [GRS80_F](#) 1.0/298.257222101 /* flattening */

Functions

- void [ITRF_to_GRS80](#) (observatory *obs)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = (char *)[TEMPO2_h_VER](#)

12.59.1 Macro Definition Documentation

12.59.1.1 #define [GRS80_A](#) 6378137.0 /* semi-major axis (m) */

12.59.1.2 #define [GRS80_F](#) 1.0/298.257222101 /* flattening */

12.59.2 Function Documentation

12.59.2.1 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.59.2.2 `void help ()`

12.59.2.3 `void ITRF_to_GRS80 (observatory * obs)`

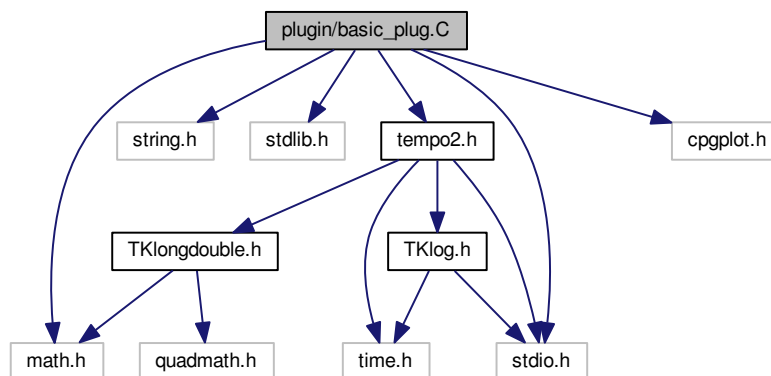
12.59.3 Variable Documentation

12.59.3.1 `const char* plugVersionCheck = (char *)TEMPO2_h_VER`

12.60 plugin/basic_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
```

Include dependency graph for basic_plug.C:



Functions

- void `callFit` (`pulsar *psr`, `int npsr`)
- void `plot_ppdot` (`pulsar *psr`)
- void `help` ()
- int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.60.1 Function Documentation

12.60.1.1 void callFit (pulsar * psr, int npsr)

12.60.1.2 int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)

12.60.1.3 void help ()

12.60.1.4 void plot_ppdot (pulsar * psr)

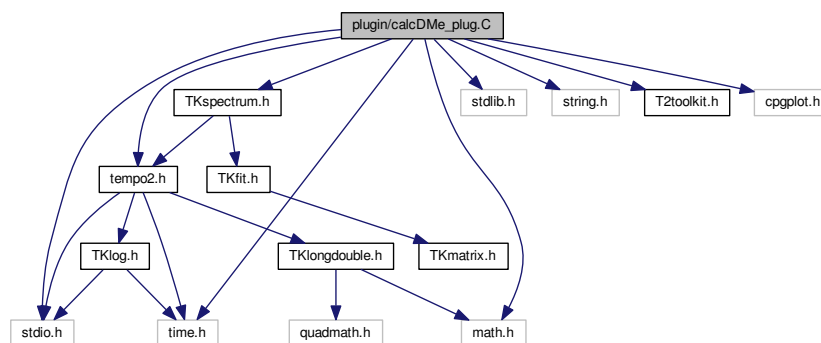
12.60.2 Variable Documentation

12.60.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.61 plugin/calcDMe_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include <time.h>
#include <cpgplot.h>
#include "TKspectrum.h"
```

Include dependency graph for calcDMe_plug.C:



Functions

- void [callFit](#) (pulsar *psr, int npsr)
- void [get_binObs](#) (pulsar *psr)
- void [setFitParams](#) (pulsar *psr)
- void [setAllDeleted](#) (pulsar *psr)
- void [resetDMandF0](#) (pulsar *psr)
- void [findFirst](#) (pulsar *psr)
- void [interpolateSplineSmooth](#) ()
- void [interpolateWeightedSmooth](#) ()
- void [init](#) (int argc, char *argv[])
- void [handleFreqPoints](#) (pulsar *psr)
- void [findSessions](#) (pulsar *psr)
- void [describe](#) ()

- double `findMean` (double *x, int count)
- void `display` (char *gr, int publish, double *xx, double *yy, longdouble *ddmMJD, longdouble *ddm, longdouble *ddmErr, int outInterpCount, int outSmoothCount, int ddmCount, char *xlab, char *ylab, char *title, double meanMJDval, double meanVal)
- void `output` (char *outFileName, int ascii, double dm0, int header, int outDM, double *outX, double *outY, int outInterpCount, int outSmoothCount, int mean, int meanMJD, double *meanMJDval, double *meanVal, int splineOut, int rawOut, longdouble *ddmMJD, longdouble *ddm, longdouble *ddmErr, int ddmCount)
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], pulsar *psr, int *npsr)

Variables

- char `parFile` [23][MAX_FILELEN]
- char `timFile` [23][MAX_FILELEN]
- char * `outFileName`
- int `fitObs` [MAX_OBSN_VAL]
- int `dmObs` [MAX_OBSN_VAL]
- int `impObs` [MAX_OBSN_VAL]
- int `fitCount` = 0
- int `dmCount` = 0
- int `impCount` = 0
- double `freqArray` [8]
- int `freqOffset` [2]
- char `freq1f` [MAX_STRLEN]
- char `freq2f` [MAX_STRLEN]
- longdouble `dm0` = -1.0
- longdouble `dm0_err` = -1.0
- longdouble `f0_0` = -1.0
- longdouble `f0_0_err` = -1.0
- longdouble `binSizeDays` = 14
- int `nf` = 0
- int `valID` [MAX_FLAGS]
- int `f0fit` = 1
- int `binObs` [MAX_OBSN_VAL][2]
- int `bin_fitCount` = 0
- int `bin_dmCount` = 0
- int `bin_fitCount_inc` = -1
- int `bin_dmCount_inc` = -1
- longdouble `binStart` = 1e10
- longdouble `ddm` [MAX_OBSN_VAL]
- longdouble `ddmMJD` [MAX_OBSN_VAL]
- longdouble `ddmErr` [MAX_OBSN_VAL]
- int `ddmCount` = 0
- double `outX` [2 * MAX_OBSN_VAL]
- double `outY` [2 * MAX_OBSN_VAL]
- int `outInterpCount` = -1
- int `outSmoothCount` = -1
- int `smoothWidth` = 100
- int `ascii` = 1
- int `gotOut` = 0
- int `header` = 0
- int `outDM` = 0
- int `mean` = 0
- int `meanMJD` = 0

- double `meanMJDval` = 0
- double `meanVal` = 0
- int `doDisplay` = 0
- char * `xlab`
- char * `ylab`
- char * `title`
- int `hardcopy` = 0
- char * `gr`
- int `splineOut` = 0
- int `rawOut` = 0
- double `sessionSeparation` = 3
- double * `start_sessions`
- double * `finish_sessions`
- int `lastUsedSession` = -1
- int `nSessions` = 0
- int `allParTim` = 0
- char `dcmFile` [MAX_FILELEN]
- const char * `plugVersionCheck` = TEMPO2_h_VER

12.61.1 Function Documentation

12.61.1.1 void `callFit` (`pulsar` * *psr*, int *npsr*)

12.61.1.2 void `describe` ()

12.61.1.3 void `display` (char * *gr*, int *publish*, double * *xx*, double * *yy*, longdouble * *ddmMJD*, longdouble * *ddm*, longdouble * *ddmErr*, int *outInterpCount*, int *outSmoothCount*, int *ddmCount*, char * *xlab*, char * *ylab*, char * *title*, double *meanMJDval*, double *meanVal*)

12.61.1.4 void `findFirst` (`pulsar` * *psr*)

12.61.1.5 double `findMean` (double * *x*, int *count*)

12.61.1.6 void `findSessions` (`pulsar` * *psr*)

12.61.1.7 void `get_binObs` (`pulsar` * *psr*)

12.61.1.8 int `graphicalInterface` (int *argc*, char * *argv*[], `pulsar` * *psr*, int * *npsr*)

12.61.1.9 void `handleFreqPoints` (`pulsar` * *psr*)

12.61.1.10 void `help` ()

12.61.1.11 void `init` (int *argc*, char * *argv*[])

12.61.1.12 void `interpolateSplineSmooth` ()

12.61.1.13 void `interpolateWeightedSmooth` ()

12.61.1.14 void `output` (char * *outFileName*, int *ascii*, double *dm0*, int *header*, int *outDM*, double * *outX*, double * *outY*, int *outInterpCount*, int *outSmoothCount*, int *mean*, int *meanMJD*, double * *meanMJDval*, double * *meanVal*, int *splineOut*, int *rawOut*, longdouble * *ddmMJD*, longdouble * *ddm*, longdouble * *ddmErr*, int *ddmCount*)

12.61.1.15 void `resetDMandF0` (`pulsar` * *psr*)

12.61.1.16 void setAllDeleted (pulsar * *psr*)

12.61.1.17 void setFitParams (pulsar * *psr*)

12.61.2 Variable Documentation

12.61.2.1 int allParTim = 0

12.61.2.2 int ascii = 1

12.61.2.3 int bin_dmCount = 0

12.61.2.4 int bin_dmCount_inc = -1

12.61.2.5 int bin_fitCount = 0

12.61.2.6 int bin_fitCount_inc = -1

12.61.2.7 int binObs[MAX_OBSN_VAL][2]

12.61.2.8 longdouble binSizeDays = 14

12.61.2.9 longdouble binStart = 1e10

12.61.2.10 char dcmFile[MAX_FILELEN]

12.61.2.11 longdouble ddm[MAX_OBSN_VAL]

12.61.2.12 int ddmCount = 0

12.61.2.13 longdouble ddmErr[MAX_OBSN_VAL]

12.61.2.14 longdouble ddmMJD[MAX_OBSN_VAL]

12.61.2.15 longdouble dm0 = -1.0

12.61.2.16 longdouble dm0_err = -1.0

12.61.2.17 int dmCount = 0

12.61.2.18 int dmObs[MAX_OBSN_VAL]

12.61.2.19 int doDisplay = 0

12.61.2.20 longdouble f0_0 = -1.0

12.61.2.21 longdouble f0_0_err = -1.0

12.61.2.22 int f0fit = 1

12.61.2.23 double* finish_sessions

12.61.2.24 int fitCount = 0

12.61.2.25 int fitObs[MAX_OBSN_VAL]

12.61.2.26 char freq1f[MAX_STRLEN]
12.61.2.27 char freq2f[MAX_STRLEN]
12.61.2.28 double freqArray[8]
12.61.2.29 int freqOffset[2]
12.61.2.30 int gotOut = 0
12.61.2.31 char* gr
12.61.2.32 int hardcopy = 0
12.61.2.33 int header = 0
12.61.2.34 int impCount = 0
12.61.2.35 int impObs[MAX_OBSN_VAL]
12.61.2.36 int lastUsedSession = -1
12.61.2.37 int mean = 0
12.61.2.38 int meanMJD = 0
12.61.2.39 double meanMJDval = 0
12.61.2.40 double meanVal = 0
12.61.2.41 int nf = 0
12.61.2.42 int nSessions = 0
12.61.2.43 int outDM = 0
12.61.2.44 char* outFileName
12.61.2.45 int outInterpCount = -1
12.61.2.46 int outSmoothCount = -1
12.61.2.47 double outX[2 * MAX_OBSN_VAL]
12.61.2.48 double outY[2 * MAX_OBSN_VAL]
12.61.2.49 char parFile[23][MAX_FILELEN]
12.61.2.50 const char* plugVersionCheck = TEMPO2_h_VER
12.61.2.51 int rawOut = 0
12.61.2.52 double sessionSeparation = 3
12.61.2.53 int smoothWidth = 100

12.61.2.54 `int splineOut = 0`

12.61.2.55 `double* start_sessions`

12.61.2.56 `char timFile[23][MAX_FILELEN]`

12.61.2.57 `char * title`

12.61.2.58 `int valid[MAX_FLAGS]`

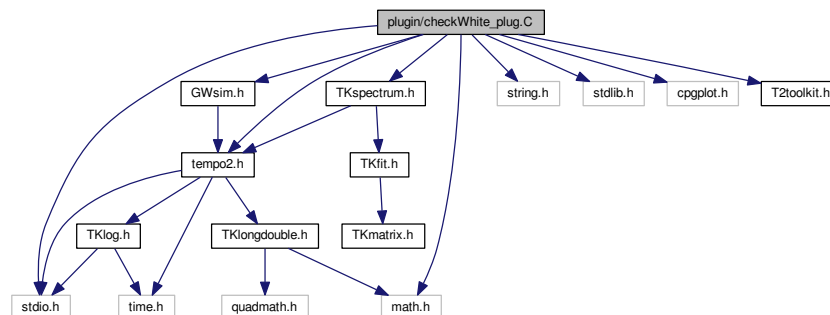
12.61.2.59 `char* xlab`

12.61.2.60 `char * ylab`

12.62 plugin/checkWhite_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <cpgplot.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "GWsim.h"
```

Include dependency graph for checkWhite_plug.C:



Macros

- `#define MAX_POLY 30`

Functions

- void `lombScargle` (`pulsar *psr`)
- void `plotResiduals` (`pulsar *psr`)
- void `shufflePoints` (`pulsar *psr`, long idum)
- void `shuffle` (double *R, double *err, double *R2, double *shuffledE, int N, long *idum)
- double `calcStat` (double *x, double *y, double *e, int n, int type)
- void `plotHistogram` (float *x, int count)
- void `average` (`pulsar *psr`)

- void `corr2pt` (`pulsar` *psr, long idum)
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], `pulsar` *psr, int *npsr)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.62.1 Macro Definition Documentation

12.62.1.1 `#define MAX_POLY 30`

12.62.2 Function Documentation

12.62.2.1 void `average` (`pulsar` * *psr*)

12.62.2.2 double `calcStat` (double * *x*, double * *y*, double * *e*, int *n*, int *type*)

12.62.2.3 void `corr2pt` (`pulsar` * *psr*, long *idum*)

12.62.2.4 int `graphicalInterface` (int *argc*, char * *argv*[], `pulsar` * *psr*, int * *npsr*)

12.62.2.5 void `help` ()

12.62.2.6 void `lombScargle` (`pulsar` * *psr*)

12.62.2.7 void `plotHistogram` (float * *x*, int *count*)

12.62.2.8 void `plotResiduals` (`pulsar` * *psr*)

12.62.2.9 void `shuffle` (double * *R*, double * *err*, double * *R2*, double * *shuffledE*, int *N*, long * *idum*)

12.62.2.10 void `shufflePoints` (`pulsar` * *psr*, long *idum*)

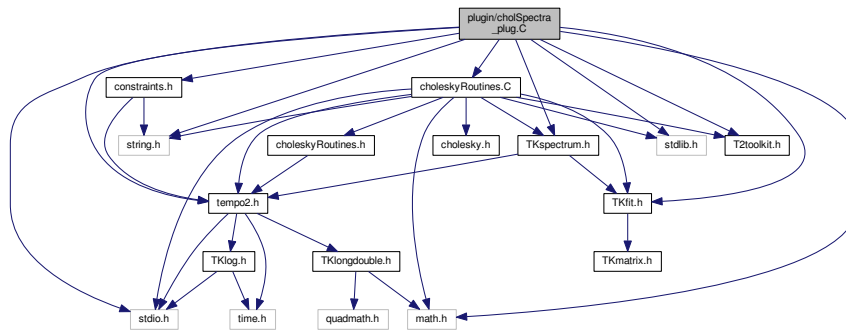
12.62.3 Variable Documentation

12.62.3.1 const char* `plugVersionCheck` = `TEMPO2_h_VER`

12.63 plugin/choISpectra_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKspectrum.h"
#include "TKfit.h"
#include "T2toolkit.h"
#include "constraints.h"
#include "choleskyRoutines.C"
```


Include dependency graph for choISpectra_plug.C:



Functions

- void [calculateSpectrum](#) ([pulsar](#) *psr, double T, int nSpec, double *px, double *py_r, double *py_i, int out↵White, int outUinv)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- double [OMEGA0](#) =0
- [longdouble](#) toffset = [longdouble](#)(52601.0)
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.63.1 Function Documentation

12.63.1.1 void [calculateSpectrum](#) ([pulsar](#) * *psr*, double *T*, int *nSpec*, double * *px*, double * *py_r*, double * *py_i*, int *outWhite*, int *outUinv*)

12.63.1.2 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.63.1.3 void [help](#) ()

12.63.2 Variable Documentation

12.63.2.1 double [OMEGA0](#) =0

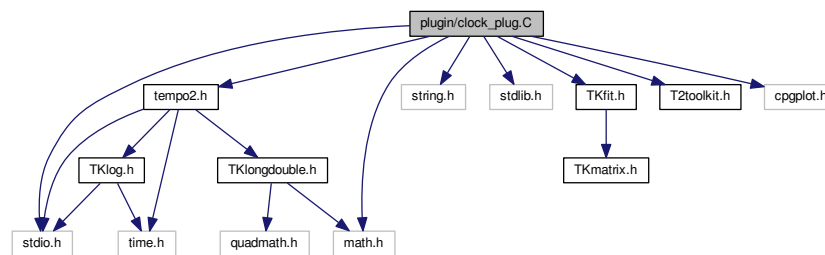
12.63.2.2 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.63.2.3 [longdouble](#) toffset = [longdouble](#)(52601.0)

12.64 plugin/clock_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKfit.h"
#include "T2toolkit.h"
#include <cpgplot.h>
```

Include dependency graph for clock_plug.C:



Functions

- void [help](#) ()
- double [mjd2year](#) (double mjd)
- void [slaClyd](#) (int iy, int im, int id, int *ny, int *nd, int *jstat)
- void [slaCalyd](#) (int iy, int im, int id, int *ny, int *nd, int *j)
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- bool [cholmode](#) =false
- char [covarFuncFile](#) [MAX_FILELEN]
- const char * [plugVersionCheck](#) = TEMPO2_h_VER

12.64.1 Function Documentation

12.64.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.64.1.2 void [help](#) ()

12.64.1.3 double [mjd2year](#) (double *mjd*)

12.64.1.4 void [slaCalyd](#) (int *iy*, int *im*, int *id*, int * *ny*, int * *nd*, int * *j*)

12.64.1.5 void [slaClyd](#) (int *iy*, int *im*, int *id*, int * *ny*, int * *nd*, int * *jstat*)

12.64.2 Variable Documentation

12.64.2.1 bool [cholmode](#) =false

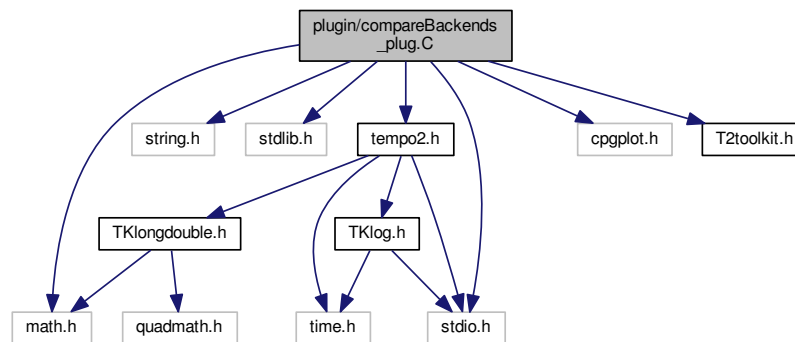
12.64.2.2 `char covarFuncFile[MAX_FILELEN]`

12.64.2.3 `const char* plugVersionCheck = TEMPO2_h_VER`

12.65 plugin/compareBackends_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
#include "T2toolkit.h"
```

Include dependency graph for compareBackends_plug.C:



Functions

- void [runPlugin](#) ([pulsar](#) *psr, int npsr, char *flagID1, char *flagID2, char *flagVal1, char *flagVal2, char *grDev, double maxTimeDiff)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.65.1 Function Documentation

12.65.1.1 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.65.1.2 `void help ()`

12.65.1.3 `void runPlugin (pulsar * psr, int npsr, char * flagID1, char * flagID2, char * flagVal1, char * flagVal2, char * grDev, double maxTimeDiff)`

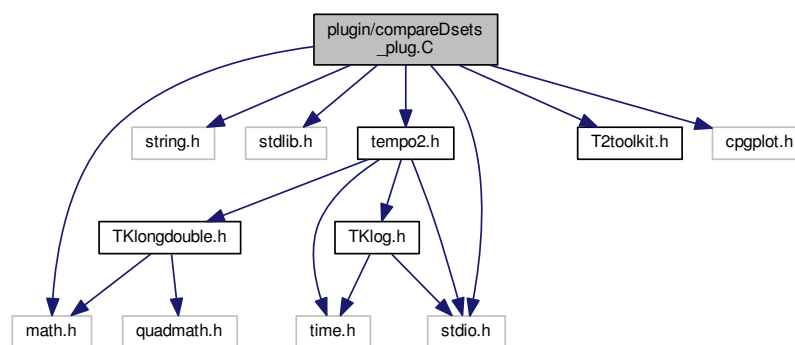
12.65.2 Variable Documentation

12.65.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.66 plugin/compareDsets_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include <cpgplot.h>
```

Include dependency graph for compareDsets_plug.C:



Functions

- void `compareDatasets` (`pulsar *psr`, `int *npsr`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`, `double maxDiff`, `char *compare`, `char *compare2`)
- int `checkSecondComparison` (`pulsar *psr`, `int i`, `int j`, `char *compare2`)
- int `findOverlap` (`pulsar *psr`, `int *npsr`, `int *overlap1`, `int *overlap2`, `double maxDiff`, `char *compare`, `char *compare2`)
- int `idPoint` (`pulsar *psr`, `int np`, `float *x_1`, `float *y_1`, `int *id_1`, `int count_1`, `float *x_2`, `float *y_2`, `int *id_2`, `int count_2`, `float mouseX`, `float mouseY`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`, `int view`)
- int `idPoint2` (`pulsar *psr`, `int np`, `float *x`, `float *y`, `int *id1`, `int count`, `float mouseX`, `float mouseY`, `int *overlap1`, `int *overlap2`, `int view`)
- void `help` ()
- int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

Variables

- `const char * plugVersionCheck = TEMPO2_h_VER`

12.66.1 Function Documentation

12.66.1.1 `int checkSecondComparison (pulsar * psr, int i, int j, char * compare2)`

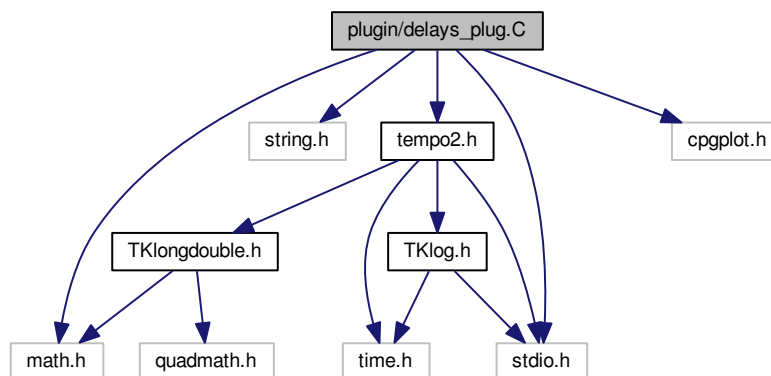
- 12.66.1.2 void compareDatasets (pulsar * psr, int * npsr, char parFile[MAX_PSR_VAL][MAX_FILELEN], char timFile[MAX_PSR_VAL][MAX_FILELEN], double maxDiff, char * compare, char * compare2)
- 12.66.1.3 int findOverlap (pulsar * psr, int * npsr, int * overlap1, int * overlap2, double maxDiff, char * compare, char * compare2)
- 12.66.1.4 int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)
- 12.66.1.5 void help ()
- 12.66.1.6 int idPoint (pulsar * psr, int np, float * x_1, float * y_1, int * id_1, int count_1, float * x_2, float * y_2, int * id_2, int count_2, float mouseX, float mouseY, char parFile[MAX_PSR_VAL][MAX_FILELEN], char timFile[MAX_PSR_VAL][MAX_FILELEN], int view)
- 12.66.1.7 int idPoint2 (pulsar * psr, int np, float * x, float * y, int * id1, int count, float mouseX, float mouseY, int * overlap1, int * overlap2, int view)

12.66.2 Variable Documentation

- 12.66.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.67 plugin/delays_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
Include dependency graph for delays_plug.C:
```



Macros

- #define MAX_HIGHLIGHT 100 /* Maximum number of points that can be highlighted */

Functions

- void doPlot (pulsar *psr, int npsr)

- float `findMin` (float *x, `pulsar` *psr, int i1, int i2)
- float `findMax` (float *x, `pulsar` *psr, int i1, int i2)
- float `findMean` (float *x, `pulsar` *psr, int i1, int i2)
- void `callFit` (`pulsar` *psr, int npsr)
- float `deletePoint` (`pulsar` *psr, int npsr, float *x, float *y, float mouseX, float mouseY)
- float `idPoint` (`pulsar` *psr, int npsr, float *x, float *y, float mouseX, float mouseY, int *ihighlight, int *nhighlight)
- double `fortranMod` (double a, double p)
- void `createNewArrivalTimes` (`pulsar` *psr, int npsr)
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], `pulsar` *psr, int *npsr)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.67.1 Macro Definition Documentation

12.67.1.1 `#define MAX_HIGHLIGHT 100` /* Maximum number of points that can be highlighted */

12.67.2 Function Documentation

12.67.2.1 void `callFit` (`pulsar` * *psr*, int *npsr*)

12.67.2.2 void `createNewArrivalTimes` (`pulsar` * *psr*, int *npsr*)

12.67.2.3 float `deletePoint` (`pulsar` * *psr*, int *npsr*, float * *x*, float * *y*, float *mouseX*, float *mouseY*)

12.67.2.4 void `doPlot` (`pulsar` * *psr*, int *npsr*)

12.67.2.5 float `findMax` (float * *x*, `pulsar` * *psr*, int *i1*, int *i2*)

12.67.2.6 float `findMean` (float * *x*, `pulsar` * *psr*, int *i1*, int *i2*)

12.67.2.7 float `findMin` (float * *x*, `pulsar` * *psr*, int *i1*, int *i2*)

12.67.2.8 double `fortranMod` (double *a*, double *p*)

12.67.2.9 int `graphicalInterface` (int *argc*, char * *argv*[], `pulsar` * *psr*, int * *npsr*)

12.67.2.10 void `help` ()

12.67.2.11 float `idPoint` (`pulsar` * *psr*, int *npsr*, float * *x*, float * *y*, float *mouseX*, float *mouseY*, int * *ihighlight*, int * *nhighlight*)

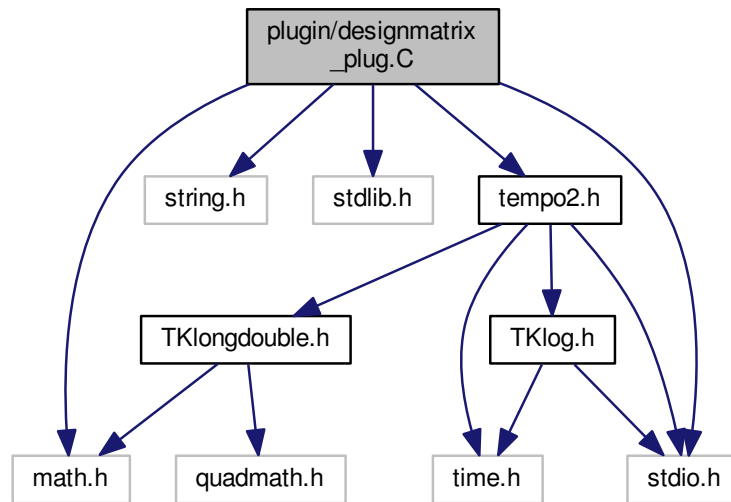
12.67.3 Variable Documentation

12.67.3.1 const char* `plugVersionCheck` = `TEMPO2_h_VER`

12.68 plugin/designmatrix_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for designmatrix_plug.C:



Functions

- int [tempo2_GetNumberOfParameters](#) (pulsar *pPsr)
- void [ProcessTempo2Objects](#) (int argc, char *argv[], pulsar *pPsr, int *pnPsr, char parFile[][MAX_FILELEN], char timFile[][MAX_FILELEN])
- void [WriteDesignMatrix](#) (const char *strFileName, pulsar *pPsr, int nTargetPulsar)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], pulsar *pPsr, int *pnPsr)

12.68.1 Function Documentation

12.68.1.1 int graphicalInterface (int argc, char * argv[], pulsar * pPsr, int * pnPsr)

12.68.1.2 void help ()

12.68.1.3 void ProcessTempo2Objects (int argc, char * argv[], pulsar * pPsr, int * pnPsr, char parFile[][MAX_FILELEN], char timFile[][MAX_FILELEN])

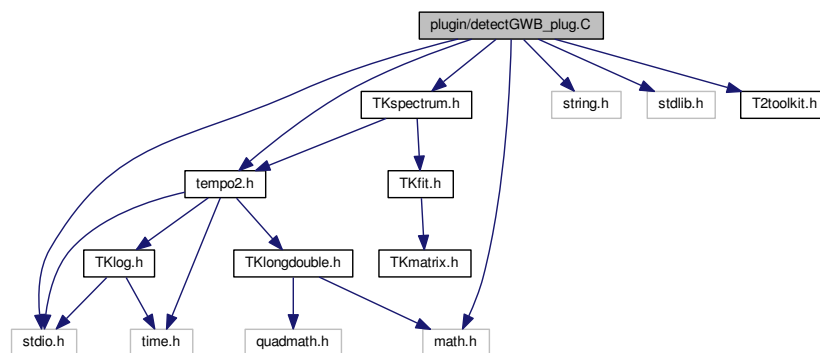
12.68.1.4 int tempo2_GetNumberOfParameters (pulsar * pPsr)

12.68.1.5 void WriteDesignMatrix (const char * strFileName, pulsar * pPsr, int nTargetPulsar)

12.69 plugin/detectGWB_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKspectrum.h"
```

Include dependency graph for detectGWB_plug.C:



Functions

- void [getSpectrum](#) ([pulsar](#) *psr, double *px, double *py_r, double *py_i, int *nSpec, double [toffset](#), double startOverlap, double endOverlap, double stepMJD, char *[covarFuncFile](#))
- void [formCholeskyMatrixPlugin](#) (double *c, double *resx, double *resy, double *rese, int np, double **uinv)
- int [calcSpectra_plugin](#) (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY_R, double *specY_I, int nfit)
- double [psrangle](#) (double centre_long, double centre_lat, double psr_long, double psr_lat)
- void [hdfunc](#) (double x, double *p, int ma)
- void [hdfunc_offs](#) (double x, double *p, int ma)
- void [hdfunc_meanSub](#) (double x, double *p, int ma)
- void [hdfunc_removeCosine](#) (double x, double *p, int ma)
- void [hdfunc_cosineSub](#) (double x, double *p, int ma)
- void [cosineFunc](#) (double x, double *p, int ma)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- bool [write_debug_files](#) =true
- double [OMEGA0](#) =0
- double [GLOBAL_MEANSUB](#) =0
- double [GLOBAL_COSVAL](#) =0

12.69.1 Function Documentation

12.69.1.1 `int calcSpectra_plugin (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY_R, double * specY_L, int nfit)`

12.69.1.2 `void cosineFunc (double x, double * p, int ma)`

12.69.1.3 `void formCholeskyMatrixPlugin (double * c, double * resx, double * resy, double * rese, int np, double ** uinv)`

12.69.1.4 `void getSpectrum (pulsar * psr, double * px, double * py_r, double * py_i, int * nSpec, double toffset, double startOverlap, double endOverlap, double stepMJD, char * covarFuncFile)`

12.69.1.5 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.69.1.6 `void hdfunc (double x, double * p, int ma)`

12.69.1.7 `void hdfunc_cosineSub (double x, double * p, int ma)`

12.69.1.8 `void hdfunc_meanSub (double x, double * p, int ma)`

12.69.1.9 `void hdfunc_offs (double x, double * p, int ma)`

12.69.1.10 `void hdfunc_removeCosine (double x, double * p, int ma)`

12.69.1.11 `void help ()`

12.69.1.12 `double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)`

12.69.2 Variable Documentation

12.69.2.1 `double GLOBAL_COSVAL =0`

12.69.2.2 `double GLOBAL_MEANSUB =0`

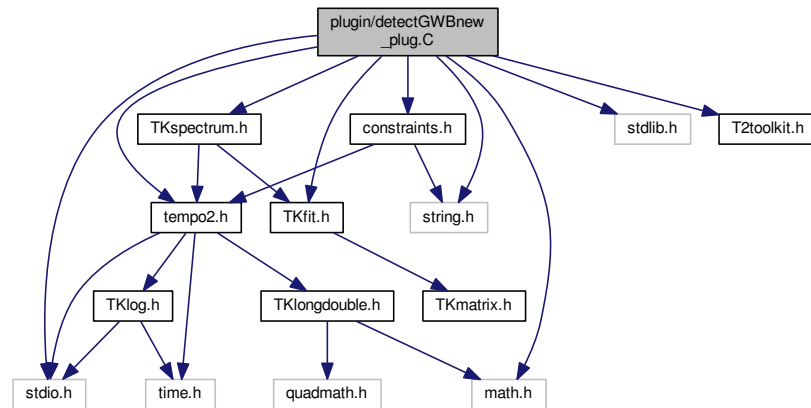
12.69.2.3 `double OMEGA0 =0`

12.69.2.4 `bool write_debug_files =true`

12.70 plugin/detectGWBnew_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "TKfit.h"
#include "constraints.h"
```

Include dependency graph for detectGWBnew_plug.C:



Functions

- double [getSpectrum](#) ([pulsar](#) *psr, double *px, double *py_r, double *py_i, int *nSpec, double [toffset](#), double startOverlap, double endOverlap, double stepMJD, char *[covarFuncFile](#), double t)
- void [formCholeskyMatrixPlugin](#) (double *c, double *resx, double *resy, double *rese, int np, double **uinv)
- int [calcSpectra_plugin](#) (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY_R, double *specY_I, int nfit)
- double [psrangle](#) (double centre_long, double centre_lat, double psr_long, double psr_lat)
- void [hdfunc](#) (double x, double *p, int ma)
- int [offsetToCM](#) ([pulsar](#) *psr)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)
- void [fitPolyFunc](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival, int ipsr)
- void [fitMeanSineFunc](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival, int ipsr)

Variables

- bool [write_debug_files](#) =true
- bool [write_python_files](#) =false
- double [OMEGA0](#) =0
- char [notim](#) =0

12.70.1 Function Documentation

12.70.1.1 int [calcSpectra_plugin](#) (double ** *uinv*, double * *resx*, double * *resy*, int *nres*, double * *specX*, double * *specY_R*, double * *specY_I*, int *nfit*)

12.70.1.2 void [fitMeanSineFunc](#) (double *x*, double * *v*, int *nfit*, [pulsar](#) * *psr*, int *ival*, int *ipsr*)

12.70.1.3 void [fitPolyFunc](#) (double *x*, double * *v*, int *nfit*, [pulsar](#) * *psr*, int *ival*, int *ipsr*)

12.70.1.4 void [formCholeskyMatrixPlugin](#) (double * *c*, double * *resx*, double * *resy*, double * *rese*, int *np*, double ** *uinv*)

12.70.1.5 `double getSpectrum (pulsar * psr, double * px, double * py_r, double * py_i, int * nSpec, double toffset, double startOverlap, double endOverlap, double stepMJD, char * covarFuncFile, double t)`

12.70.1.6 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.70.1.7 `void hdfunc (double x, double * p, int ma)`

12.70.1.8 `void help ()`

12.70.1.9 `int offsetToCM (pulsar * psr)`

12.70.1.10 `double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)`

12.70.2 Variable Documentation

12.70.2.1 `char notim =0`

12.70.2.2 `double OMEGA0 =0`

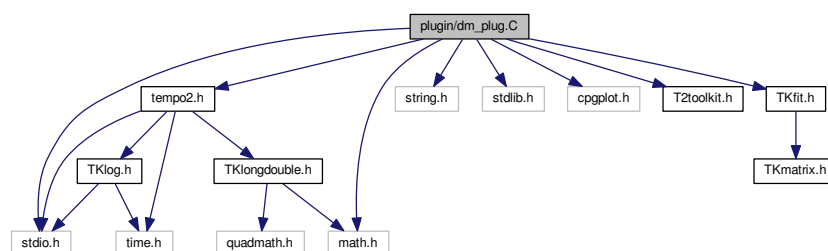
12.70.2.3 `bool write_debug_files =true`

12.70.2.4 `bool write_python_files =false`

12.71 plugin/dm_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cpplot.h>
#include "T2toolkit.h"
#include "TKfit.h"
```

Include dependency graph for dm_plug.C:



Macros

- `#define MAX_TIMES 1000`

Functions

- `void help ()`

- void `doPlot` (double *epoch, double *dmVal, double *dmE, int *id, int n, `pulsar` *psr)
- double `mjd2year` (double mjd)
- void `selectData` (`pulsar` *psr, float *rx, float *ry, double f1, double f2, float *plotX, float *plotY, float *plotE, int *nplot)
- void `slaClyd` (int iy, int im, int id, int *ny, int *nd, int *jstat)
- void `slaCalyd` (int iy, int im, int id, int *ny, int *nd, int *j)
- int `graphicalInterface` (int argc, char *argv[], `pulsar` *psr, int *npsr)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.71.1 Macro Definition Documentation

12.71.1.1 `#define MAX_TIMES 1000`

12.71.2 Function Documentation

12.71.2.1 void `doPlot` (double * *epoch*, double * *dmVal*, double * *dmE*, int * *id*, int *n*, `pulsar` * *psr*)

12.71.2.2 int `graphicalInterface` (int *argc*, char * *argv*[], `pulsar` * *psr*, int * *npsr*)

12.71.2.3 void `help` ()

12.71.2.4 double `mjd2year` (double *mjd*)

12.71.2.5 void `selectData` (`pulsar` * *psr*, float * *rx*, float * *ry*, double *f1*, double *f2*, float * *plotX*, float * *plotY*, float * *plotE*, int * *nplot*)

12.71.2.6 void `slaCalyd` (int *iy*, int *im*, int *id*, int * *ny*, int * *nd*, int * *j*)

12.71.2.7 void `slaClyd` (int *iy*, int *im*, int *id*, int * *ny*, int * *nd*, int * *jstat*)

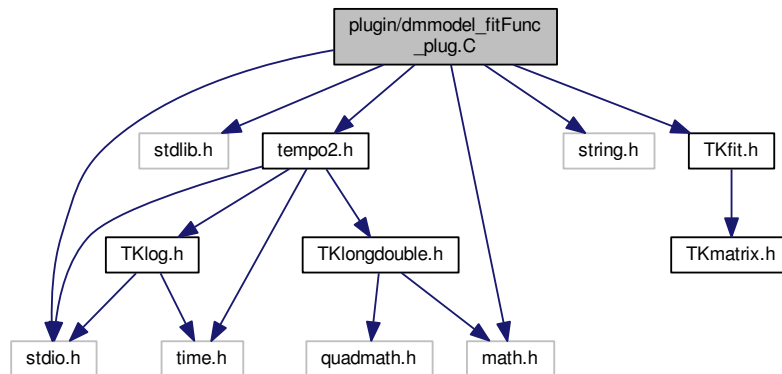
12.71.3 Variable Documentation

12.71.3.1 const char* `plugVersionCheck` = `TEMPO2_h_VER`

12.72 plugin/dmmodel_fitFunc_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <tempo2.h>
#include <string.h>
#include <math.h>
#include "TKfit.h"
```

Include dependency graph for dmmodel_fitFunc_plug.C:



Functions

- void [updateDMvals](#) ([pulsar](#) *psr, int p)
- void [getFitLabels](#) ([pulsar](#) *psr, int p, char **ret)
- int [pluginFitFunc](#) ([pulsar](#) *psr, int npsr, int writeModel)

12.72.1 Function Documentation

12.72.1.1 void [getFitLabels](#) ([pulsar](#) * *psr*, int *p*, char ** *ret*)

12.72.1.2 int [pluginFitFunc](#) ([pulsar](#) * *psr*, int *npsr*, int *writeModel*)

12.72.1.3 void [updateDMvals](#) ([pulsar](#) * *psr*, int *p*)

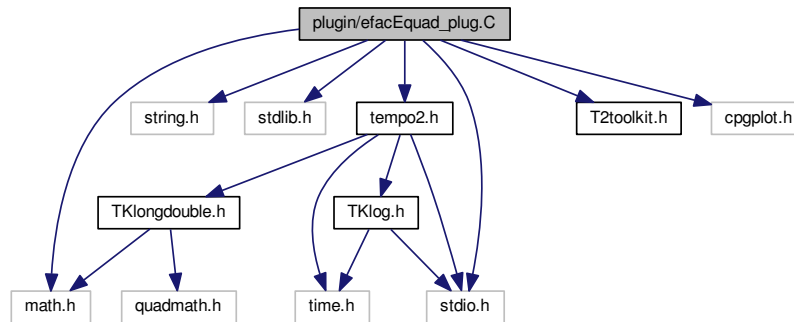
12.73 plugin/efacEquad_plug.C File Reference

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include <cpgplot.h>

```

Include dependency graph for efacEquad_plug.C:



Macros

- `#define FMAX(x, y) ((x<y)?y:x)`
- `#define EPS1 0.001`
- `#define EPS2 1.0e-8`
- `#define SWAP(a, b) temp=(a);(a)=(b);(b)=temp;`
- `#define M 7`
- `#define NSTACK 50`
- `#define FREE_ARG char*`
- `#define NR_END 1`
- `#define ITMAX 100`
- `#define EPS 3.0e-7`
- `#define FPMIN 1.0e-30`

Functions

- void `calcEfacEquad` (double *px, double *py, double *pe, int npts, double *efacRet, double *equadRet, int disp)
- void `calcEfacEquad2` (double *px, double *py, double *pe, int npts, double *efacRet, double *equadRet, int disp, double correctEfac, double correctEquad, double minEquad, double maxEquad, double stepEquad, double minEfac, double maxEfac, double stepEfac, char *grDev)
- int * `ivector` (long nl, long nh)
- void `free_ivector` (int *v, long nl, long nh)
- void `kstwo` (double data1[], unsigned long n1, double data2[], unsigned long n2, double *d, double *prob)
- double `probks` (double alam)
- double `gaussFunc` (double val)
- void `sort` (unsigned long n, double arr[])
- void `nrerror` (const char *error_text)
- double `erff` (double x)
- void `ksone` (double data[], unsigned long n, double(*func)(double), double *d, double *prob)
- double `gammp` (double a, double x)
- double `gammln` (double xx)
- void `gcf` (double *gammcf, double a, double x, double *gln)
- void `gser` (double *gamser, double a, double x, double *gln)
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], `pulsar` *psr, int *npsr)
- void `calcEfacEquad` (double *px, double *py, double *pe, int npts, double *efacRet, double *equadRet, int disp, char *grDev)

Variables

- const char * [plugVersionCheck](#) = TEMPO2_h_VER

12.73.1 Macro Definition Documentation

- 12.73.1.1 `#define EPS 3.0e-7`
- 12.73.1.2 `#define EPS1 0.001`
- 12.73.1.3 `#define EPS2 1.0e-8`
- 12.73.1.4 `#define FMAX(x, y) ((x<y)?y:x)`
- 12.73.1.5 `#define FPMIN 1.0e-30`
- 12.73.1.6 `#define FREE_ARG char*`
- 12.73.1.7 `#define ITMAX 100`
- 12.73.1.8 `#define M 7`
- 12.73.1.9 `#define NR_END 1`
- 12.73.1.10 `#define NSTACK 50`
- 12.73.1.11 `#define SWAP(a, b) temp=(a);(a)=(b);(b)=temp;`

12.73.2 Function Documentation

- 12.73.2.1 `void calcEfacEquad (double * px, double * py, double * pe, int npts, double * efacRet, double * equadRet, int disp)`
- 12.73.2.2 `void calcEfacEquad (double * px, double * py, double * pe, int npts, double * efacRet, double * equadRet, int disp, char * grDev)`
- 12.73.2.3 `void calcEfacEquad2 (double * px, double * py, double * pe, int npts, double * efacRet, double * equadRet, int disp, double correctEfac, double correctEquad, double minEquad, double maxEquad, double stepEquad, double minEfac, double maxEfac, double stepEfac, char * grDev)`
- 12.73.2.4 `double erff (double x)`
- 12.73.2.5 `void free_ivector (int * v, long nl, long nh)`
- 12.73.2.6 `double gamm1n (double xx)`
- 12.73.2.7 `double gammf (double a, double x)`
- 12.73.2.8 `double gaussFunc (double val)`
- 12.73.2.9 `void gcf (double * gammcf, double a, double x, double * gln)`
- 12.73.2.10 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`
- 12.73.2.11 `void gser (double * gamser, double a, double x, double * gln)`

12.73.2.12 void help ()

12.73.2.13 int * ivector (long *nl*, long *nh*)

12.73.2.14 void ksone (double *data*[], unsigned long *n*, double(*) (double) *func*, double * *d*, double * *prob*)

12.73.2.15 void kstwo (double *data1*[], unsigned long *n1*, double *data2*[], unsigned long *n2*, double * *d*, double * *prob*)

12.73.2.16 void nerror (const char * *error_text*)

12.73.2.17 double probks (double *alam*)

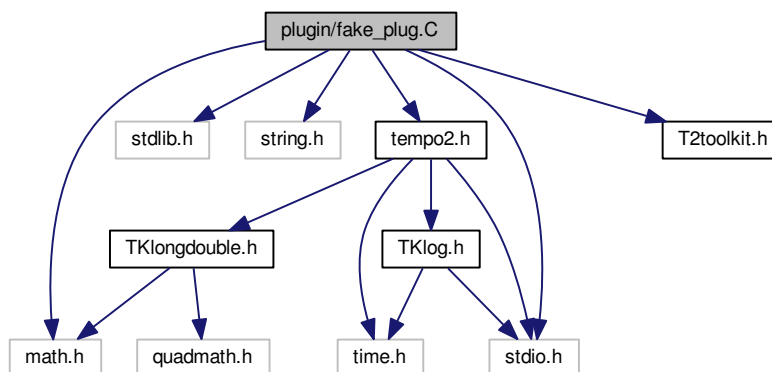
12.73.2.18 void sort (unsigned long *n*, double *arr*[])

12.73.3 Variable Documentation

12.73.3.1 const char* plugVersionCheck = TEMPO2_h_VER

12.74 plugin/fake_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
Include dependency graph for fake_plug.C:
```



Functions

- void [callFit](#) ([pulsar](#) *psr, int npsr)
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.74.1 Function Documentation

12.74.1.1 void callFit (pulsar * psr, int npsr)

12.74.1.2 int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)

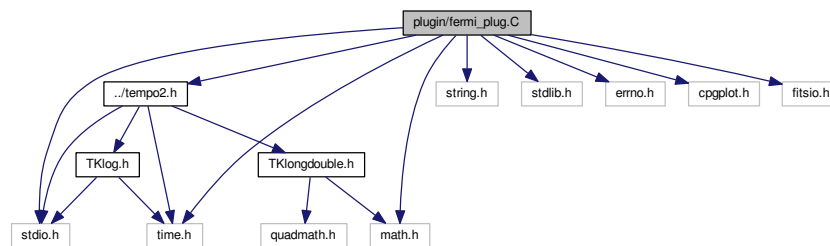
12.74.2 Variable Documentation

12.74.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.75 plugin/fermi_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <errno.h>
#include "../tempo2.h"
#include <cpgplot.h>
#include <fitsio.h>
#include <time.h>
```

Include dependency graph for fermi_plug.C:



Macros

- #define SECDAY 86400.0

Functions

- void extra_delays_fermi (pulsar *psr, int npsr)
- void clock_corrections_fermi (pulsar *psr, int npsr)
- void ephemeris_routines_fermi (pulsar *psr, int npsr)
- void formBatsAll_fermi (pulsar *psr, int npsr)
- float HTest (int Nphotons, float phases[])
- void cpgpt (int n, const float *xpts, const longdouble *ypts, int symbol)
- longdouble met2mjd (double met)
- double mjd2met (longdouble mjd)
- double inner_product (double vect_x[], double vect_y[])
- void outer_product (double vect_x[], double vect_y[], double vect_z[])
- int graphicalInterface (int argc, char *argv[], pulsar *psr, int *npsr)

Variables

- `const char * plugVersionCheck = TEMPO2_h_VER`

12.75.1 Macro Definition Documentation

12.75.1.1 `#define SECDAY 86400.0`

12.75.2 Function Documentation

12.75.2.1 `void clock_corrections_fermi (pulsar * psr, int npsr)`

12.75.2.2 `void cpgpt (int n, const float * xpts, const longdouble * ypts, int symbol)`

12.75.2.3 `void ephemeris_routines_fermi (pulsar * psr, int npsr)`

12.75.2.4 `void extra_delays_fermi (pulsar * psr, int npsr)`

12.75.2.5 `void formBatsAll_fermi (pulsar * psr, int npsr)`

12.75.2.6 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.75.2.7 `float HTest (int Nphotons, float phases[])`

12.75.2.8 `double inner_product (double vect_x[], double vect_y[])`

12.75.2.9 `longdouble met2mjd (double met)`

12.75.2.10 `double mjd2met (longdouble mjd)`

12.75.2.11 `void outer_product (double vect_x[], double vect_y[], double vect_z[])`

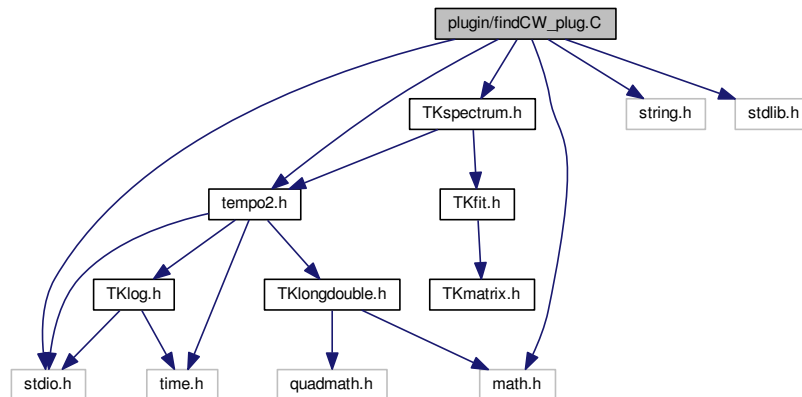
12.75.3 Variable Documentation

12.75.3.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.76 plugin/findCW_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKspectrum.h"
```

Include dependency graph for findCW_plug.C:



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.76.1 Function Documentation

12.76.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.76.1.2 void [help](#) ()

12.76.2 Variable Documentation

12.76.2.1 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

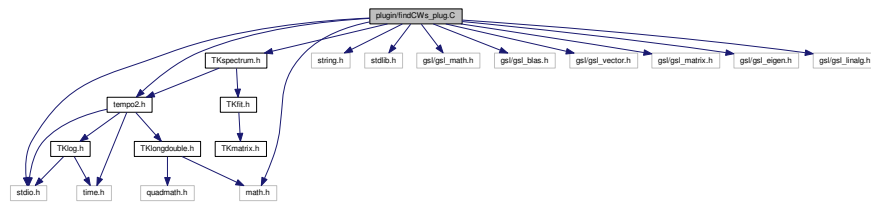
12.77 plugin/findCWs_plug.C File Reference

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKspectrum.h"
#include <gsl/gsl_math.h>
#include <gsl/gsl_blas.h>
#include <gsl/gsl_vector.h>
#include <gsl/gsl_matrix.h>
#include <gsl/gsl_eigen.h>
#include <gsl/gsl_linalg.h>

```

Include dependency graph for findCWs_plug.C:



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

12.77.1 Function Documentation

12.77.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.77.1.2 void [help](#) ()

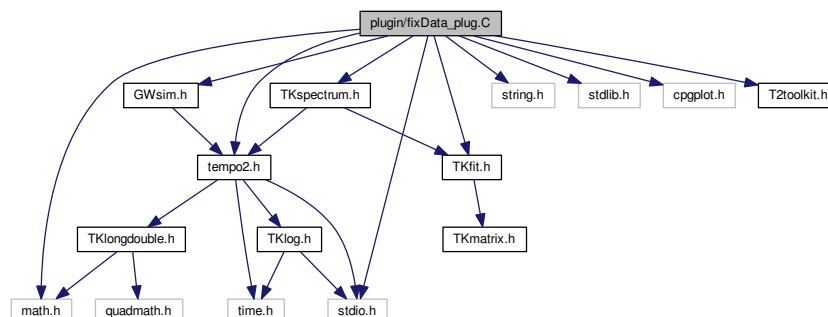
12.78 plugin/fixData_plug.C File Reference

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <cpgplot.h>
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "TKfit.h"
#include "tempo2.h"
#include "GWsim.h"

```

Include dependency graph for fixData_plug.C:



Functions

- void [doPlugin1](#) ([pulsar](#) *psr, char *flag, int removeQuad)

- void `doPlugin3` (`pulsar *psr`, `char *flag`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`, `int argc`, `char *argv[]`, `float dstep`)
- int `determine1dStructureFunction` (`float *x`, `float *y`, `float *ye`, `int nn`, `double *errfac1`, `double *vsf`, `double *mverr`)
- void `doPlugin2` (`pulsar *psr`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`, `int argc`, `char *argv[]`)
- float `plotHistogram` (`float *x`, `int count`, `int *flagCol`, `int nFlag`, `char flagV[100][16]`)
- void `doSummary` (`pulsar *psr`, `float errStep`)
- void `help` ()
- int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

Variables

- int `nit` = 1
- `longdouble` `gwamp` = 0
- `longdouble` `alpha` = 1
- `char` `plotout` [20] = "/xs"
- int `plotoutSet` = 0
- int `script` = 0
- int `dayGap` = 1
- const `char *` `plugVersionCheck` = `TEMPO2_h_VER`

12.78.1 Function Documentation

- 12.78.1.1 int `determine1dStructureFunction` (`float * x`, `float * y`, `float * ye`, `int nn`, `double * errfac1`, `double * vsf`, `double * mverr`)
- 12.78.1.2 void `doPlugin1` (`pulsar * psr`, `char * flag`, `int removeQuad`)
- 12.78.1.3 void `doPlugin2` (`pulsar * psr`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`, `int argc`, `char * argv[]`)
- 12.78.1.4 void `doPlugin3` (`pulsar * psr`, `char * flag`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`, `int argc`, `char * argv[]`, `float dstep`)
- 12.78.1.5 void `doSummary` (`pulsar * psr`, `float errStep`)
- 12.78.1.6 int `graphicalInterface` (`int argc`, `char * argv[]`, `pulsar * psr`, `int * npsr`)
- 12.78.1.7 void `help` ()
- 12.78.1.8 float `plotHistogram` (`float * x`, `int count`, `int * flagCol`, `int nFlag`, `char flagV[100][16]`)

12.78.2 Variable Documentation

- 12.78.2.1 `longdouble` `alpha` = 1
- 12.78.2.2 int `dayGap` = 1
- 12.78.2.3 `longdouble` `gwamp` = 0
- 12.78.2.4 int `nit` = 1
- 12.78.2.5 `char` `plotout`[20] = "/xs"

12.78.2.6 `int plotoutSet = 0`

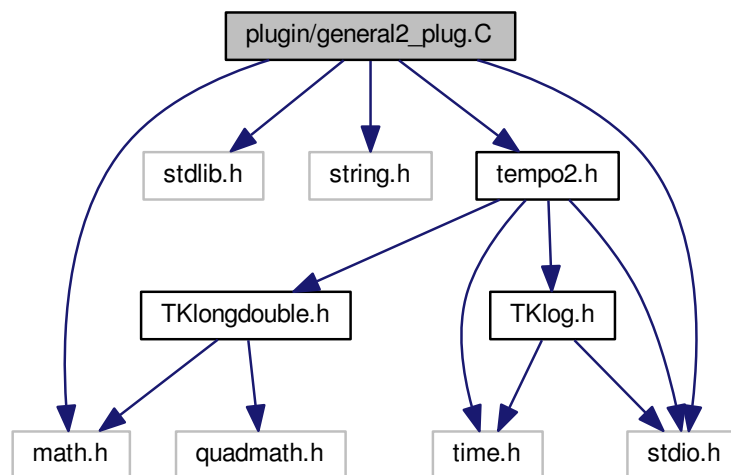
12.78.2.7 `const char* plugVersionCheck = TEMPO2_h_VER`

12.78.2.8 `int script = 0`

12.79 plugin/general2_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for general2_plug.C:



Functions

- `int nint_derived (double x)`
- `int rnd8 (double rval, double rerr, int ifac, char *cval, int *lv, char *cerr, int *le, char *msg)`
- `void parseLine (pulsar *psr, char *line, double *errMult, char *null, char *format, char *dformat, int *rad, FILE *fout)`
- `double fortranMod (double a, double p)`
- `int tempoOutput (int argc, char *argv[], pulsar *psr, int npsr)`

Variables

- `const char * plugVersionCheck = TEMPO2_h_VER`

12.79.1 Function Documentation

12.79.1.1 `double fortranMod (double a, double p)`

12.79.1.2 `int nint_derived (double x)`

12.79.1.3 `void parseLine (pulsar * psr, char * line, double * errMult, char * null, char * format, char * dformat, int * rad, FILE * fout)`

12.79.1.4 `int rnd8 (double rval, double rerr, int ifac, char * cval, int * lv, char * cerr, int * le, char * msg)`

12.79.1.5 `int tempoOutput (int argc, char * argv[], pulsar * psr, int npsr)`

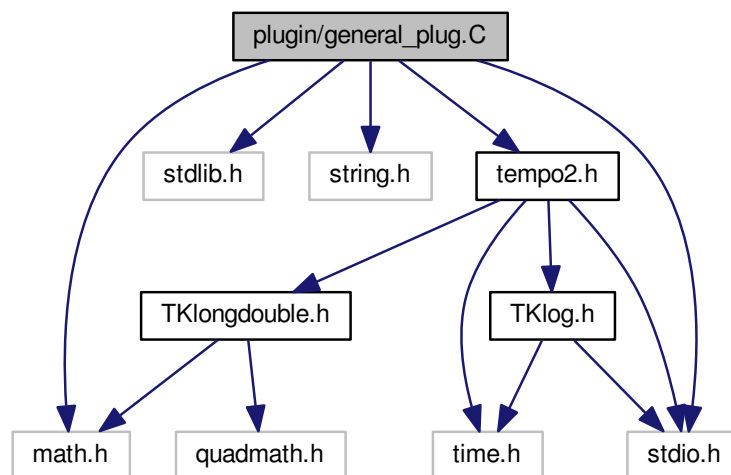
12.79.2 Variable Documentation

12.79.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.80 plugin/general_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for general_plug.C:



Functions

- `int nint_derived (double x)`
- `int rnd8 (double rval, double rerr, int ifac, char *cval, int *lv, char *cerr, int *le, char *msg)`
- `void parseLine (pulsar *psr, char *line, double *errMult, char *null, char *format, int *rad)`
- `int tempoOutput (int argc, char *argv[], pulsar *psr, int npsr)`

Variables

- `const char * plugVersionCheck = TEMPO2_h_VER`

12.80.1 Function Documentation

12.80.1.1 `int nint_derived (double x)`

12.80.1.2 `void parseLine (pulsar * psr, char * line, double * errMult, char * null, char * format, int * rad)`

12.80.1.3 `int rnd8 (double rval, double rerr, int ifac, char * cval, int * lv, char * cerr, int * le, char * msg)`

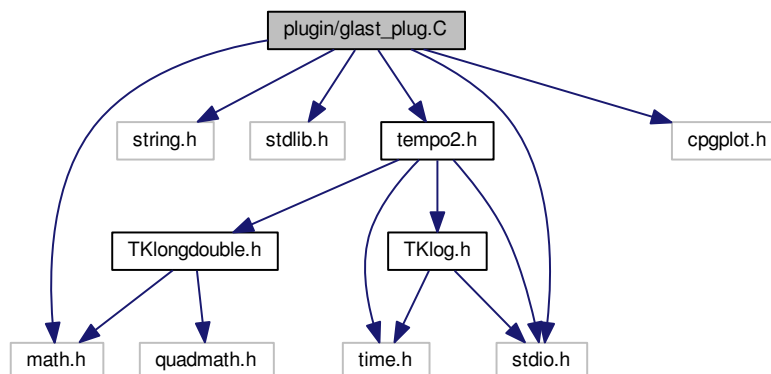
12.80.1.4 `int tempoOutput (int argc, char * argv[], pulsar * psr, int npsr)`

12.80.2 Variable Documentation

12.80.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.81 plugin/glast_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
Include dependency graph for glast_plug.C:
```



Macros

- `#define NRANSI`
- `#define SWAP(a, b) itemp=(a);(a)=(b);(b)=itemp;`
- `#define M 7`
- `#define NSTACK 50`
- `#define NR_END 1`
- `#define FREE_ARG char*`

Functions

- `void sla_CALDJ (int IY, int IM, int ID, double *DJM, int *J)`

- void [sla_CLDJ](#) (int *IY*, int *IM*, int *ID*, double **DJM*, int **J*)
- void [slaClyd](#) (int *iy*, int *im*, int *id*, int **ny*, int **nd*, int **jstat*)
- void [slaCalyd](#) (int *iy*, int *im*, int *id*, int **ny*, int **nd*, int **j*)
- int [getParameter](#) ([pulsar](#) *psr*, const char **param*, double **value*)
- float [fitwave_function](#) ([pulsar](#) **psr*, float *x*, float *fitwaves_omega*, float *fitwaves_epoch*)
- void [indexx_patrick](#) (unsigned long *n*, float *arr*[], unsigned long *indx*[])
- void [help](#) ()
- int [graphicalInterface](#) (int *argc*, char **argv*[], [pulsar](#) **psr*, int **npsr*)
- void [nrerror](#) (const char **error_text*)
- int * [ivector](#) (long *nl*, long *nh*)
- void [free_ivector](#) (int **v*, long *nl*, long *nh*)

12.81.1 Macro Definition Documentation

12.81.1.1 `#define FREE_ARG char*`

12.81.1.2 `#define M 7`

12.81.1.3 `#define NR_END 1`

12.81.1.4 `#define NRANSI`

12.81.1.5 `#define NSTACK 50`

12.81.1.6 `#define SWAP(a, b) itemp=(a);(a)=(b);(b)=itemp;`

12.81.2 Function Documentation

12.81.2.1 float [fitwave_function](#) ([pulsar](#) * *psr*, float *x*, float *fitwaves_omega*, float *fitwaves_epoch*)

12.81.2.2 void [free_ivector](#) (int * *v*, long *nl*, long *nh*)

12.81.2.3 int [getParameter](#) ([pulsar](#) *psr*, const char * *param*, double * *value*)

12.81.2.4 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.81.2.5 void [help](#) ()

12.81.2.6 void [indexx_patrick](#) (unsigned long *n*, float *arr*[], unsigned long *indx*[])

12.81.2.7 int* [ivector](#) (long *nl*, long *nh*)

12.81.2.8 void [nrerror](#) (const char * *error_text*)

12.81.2.9 void [sla_CALDJ](#) (int *IY*, int *IM*, int *ID*, double * *DJM*, int * *J*)

12.81.2.10 void [sla_CLDJ](#) (int *IY*, int *IM*, int *ID*, double * *DJM*, int * *J*)

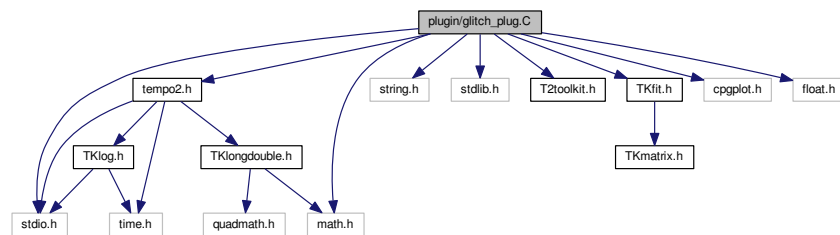
12.81.2.11 void [slaCalyd](#) (int *iy*, int *im*, int *id*, int * *ny*, int * *nd*, int * *j*)

12.81.2.12 void [slaClyd](#) (int *iy*, int *im*, int *id*, int * *ny*, int * *nd*, int * *jstat*)

12.82 plugin/glitch_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include <cpgplot.h>
#include <float.h>
```

Include dependency graph for glitch_plug.C:



Classes

- struct [glitchS](#)
- struct [lm_control_struct](#)
- struct [lm_status_struct](#)
- struct [lmcurve_data_struct](#)

Macros

- #define [LM_MACHEP](#) DBL_EPSILON /* resolution of arithmetic */
- #define [LM_DWARF](#) DBL_MIN /* smallest nonzero number */
- #define [LM_SQRT_DWARF](#) sqrt(DBL_MIN) /* square should not underflow */
- #define [LM_SQRT_GIANT](#) sqrt(DBL_MAX) /* square should not overflow */
- #define [LM_USERTOL](#) 1e-9
- #define [MIN](#)(a, b) (((a)<=(b)) ? (a) : (b))
- #define [MAX](#)(a, b) (((a)>=(b)) ? (a) : (b))
- #define [SQR](#)(x) (x)*(x)
- #define [MAX_TIMES](#) 2000

Typedefs

- typedef struct [glitchS](#) [glitchS](#)

Functions

- void [defineGlitchVal](#) ([glitchS](#) *glitch, int nglt)

- void [doPlot](#) (double *epoch, double *f0, double *f0e, double *f1, double *f1e, int fitf1, int *nFit, int *id, int n, float *gt, int nglt, int *plotType, int nplot, double plotOffset, double *plotResX, double *plotResY, double *plotResE, int nplotVal, int combine, float fontSize, char *title, float *yscale_min, float *yscale_max, int *yscale_set, int interactive)
- void [plot1](#) (double *epoch, double *f0, float *yerr1, float *yerr2, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [plot8](#) (double *epoch, double *f0, float *yerr1, float *yerr2, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [plot2](#) (double *epoch, double *f0, double *f0e, int *nFit, int *id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [interactivePlot](#) (double *epoch, double *f0, double *f0e, int *nFit, int *id, int n)
- void [fitFuncs](#) (double x, double *p, int m)
- void [changeFit](#) ([glitchS](#) *glitch, int nglt)
- void [drawMenu](#) (float minx, float maxx, float miny, float maxy, [glitchS](#) *glitch, int nglt, int fitf0, int fitf1)
- void [checkMenu](#) (float minx, float maxx, float miny, float maxy, [glitchS](#) *glitch, int nglt, float mx, float my, int *fitf0, int *fitf1, char key)
- double [nonlinearFunc](#) (double t, const double *par, int obsNum)
- void [plot3](#) (double *epoch, double *f1, double *f1e, int *nFit, int *id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [plot6](#) (double *epoch, double *f1, double *f1e, int *nFit, int *id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [plot4](#) (double *epoch, double *f0, double *f0e, int *nFit, int *id, int n, double plotOffset, float *gt, int nglt, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [plot5](#) (double *epoch, double *f0, double *f0e, int *nFit, int *id, int n, double plotOffset, float *gt, int nglt, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [plot7](#) (double *plotResX, double *plotResY, double *plotResE, int nplotVal, double plotOffset, int combine, int pos, int nplot, double start, double end, double psrF0, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [plot9](#) (double *epoch, double *f0, double *f0e, int *nFit, int *id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)
- void [lm_printout_std](#) (int n_par, const double *par, int m_dat, const void *data, const double *fvec, int printflags, int iflag, int iter, int nfev)
- double [lm_enorm](#) (int, const double *)
- void [lmmin](#) (int n_par, double *par, int m_dat, const void *data, void(*evaluate)(const double *par, int m_dat, const void *data, double *fvec, int *info), const [lm_control_struct](#) *control, [lm_status_struct](#) *status, void(*printout)(int n_par, const double *par, int m_dat, const void *data, const double *fvec, int printflags, int iflag, int iter, int nfev))
- void [lm_lmdif](#) (int m, int n, double *x, double *fvec, double ftol, double xtol, double gtol, int maxfev, double epsfcn, double *diag, int mode, double factor, int *info, int *nfev, double *fjac, int *ipvt, double *qtf, double *wa1, double *wa2, double *wa3, double *wa4, void(*evaluate)(const double *par, int m_dat, const void *data, double *fvec, int *info), void(*printout)(int n_par, const double *par, int m_dat, const void *data, const double *fvec, int printflags, int iflag, int iter, int nfev), int printflags, const void *data)
- void [lmcurve_fit](#) (int n_par, double *par, int m_dat, const double *t, const double *y, double(*f)(double t, const double *par, int obsNum), const [lm_control_struct](#) *control, [lm_status_struct](#) *status)
- void [lm_lmpar](#) (int n, double *r, int ldr, int *ipvt, double *diag, double *qtb, double delta, double *par, double *x, double *sdiag, double *aux, double *xdi)
- void [lm_qrfac](#) (int m, int n, double *a, int pivot, int *ipvt, double *rdiag, double *acnorm, double *wa)
- void [lm_qrsolv](#) (int n, double *r, int ldr, int *ipvt, double *diag, double *qtb, double *x, double *sdiag, double *wa)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)
- void [lmcurve_evaluate](#) (const double *par, int m_dat, const void *data, double *fvec, int *info)

Variables

- [glitchS](#) * [global_glitch](#)
- int [global_nglt](#)
- int [global_fitf0](#)
- int [global_fitf1](#)
- double [global_valf0](#)
- double [global_valf1](#)
- double [global_footer](#)
- double [global_header](#)
- const [lm_control_struct](#) [lm_control_double](#)
- const [lm_control_struct](#) [lm_control_float](#)
- const char * [lm_infmsg](#) []
- const char * [lm_shortmsg](#) []
- const char * [plugVersionCheck](#) = (char *)[TEMPO2_h_VER](#)

12.82.1 Macro Definition Documentation

- 12.82.1.1 `#define LM_DWARF DBL_MIN /* smallest nonzero number */`
- 12.82.1.2 `#define LM_MACHEP DBL_EPSILON /* resolution of arithmetic */`
- 12.82.1.3 `#define LM_SQRT_DWARF sqrt(DBL_MIN) /* square should not underflow */`
- 12.82.1.4 `#define LM_SQRT_GIANT sqrt(DBL_MAX) /* square should not overflow */`
- 12.82.1.5 `#define LM_USERTOL 1e-9`
- 12.82.1.6 `#define MAX(a, b) (((a)>=(b)) ? (a) : (b))`
- 12.82.1.7 `#define MAX_TIMES 2000`
- 12.82.1.8 `#define MIN(a, b) (((a)<=(b)) ? (a) : (b))`
- 12.82.1.9 `#define SQR(x) (x)*(x)`

12.82.2 Typedef Documentation

- 12.82.2.1 `typedef struct glitchS glitchS`

12.82.3 Function Documentation

- 12.82.3.1 `void changeFit (glitchS * glitch, int nglt)`
- 12.82.3.2 `void checkMenu (float minx, float maxx, float miny, float maxy, glitchS * glitch, int nglt, float mx, float my, int * fitf0, int * fitf1, char key)`
- 12.82.3.3 `void defineGlitchVal (glitchS * glitch, int nglt)`
- 12.82.3.4 `void doPlot (double * epoch, double * f0, double * f0e, double * f1, double * f1e, int fitf1, int * nFit, int * id, int n, float * gt, int ngt, int * plotType, int nplot, double plotOffset, double * plotResX, double * plotResY, double * plotResE, int nplotVal, int combine, float fontSize, char * title, float * yscale_min, float * yscale_max, int * yscale_set, int interactive)`
- 12.82.3.5 `void drawMenu (float minx, float maxx, float miny, float maxy, glitchS * glitch, int nglt, int fitf0, int fitf1)`

12.82.3.6 void fitFuncs (double *x*, double * *p*, int *m*)

12.82.3.7 int graphicalInterface (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)

12.82.3.8 void help ()

12.82.3.9 void interactivePlot (double * *epoch*, double * *f0*, double * *f0e*, int * *nFit*, int * *id*, int *n*)

12.82.3.10 double lm_enorm (int *n*, const double * *x*)

sum squares.

calculation of norm.

12.82.3.11 void lm_lmdif (int *m*, int *n*, double * *x*, double * *fvec*, double *ftol*, double *xtol*, double *gtol*, int *maxfev*, double *epsfcn*, double * *diag*, int *mode*, double *factor*, int * *info*, int * *nfev*, double * *fjac*, int * *ipvt*, double * *qtf*, double * *wa1*, double * *wa2*, double * *wa3*, double * *wa4*, void(*) (const double * *par*, int *m_dat*, const void * *data*, double * *fvec*, int * *info*) *evaluate*, void(*) (int *n_par*, const double * *par*, int *m_dat*, const void * *data*, const double * *fvec*, int *printflags*, int *iflag*, int *iter*, int *nfev*) *printout*, int *printflags*, const void * *data*)

Legacy low-level interface.

12.82.3.12 void lm_lmpar (int *n*, double * *r*, int *ldr*, int * *ipvt*, double * *diag*, double * *qtb*, double *delta*, double * *par*, double * *x*, double * *sdiag*, double * *aux*, double * *xdi*)

evaluate the function at the current value of par.

if the function is small enough, accept the current value of par. Also test for the exceptional cases where par1 is zero or the number of iterations has reached 10.

compute the Newton correction.

depending on the sign of the function, update par1 or paru.

compute an improved estimate for par.

12.82.3.13 void lm_printout_std (int *n_par*, const double * *par*, int *m_dat*, const void * *data*, const double * *fvec*, int *printflags*, int *iflag*, int *iter*, int *nfev*)

12.82.3.14 void lm_qrfac (int *m*, int *n*, double * *a*, int *pivot*, int * *ipvt*, double * *rdiag*, double * *acnorm*, double * *wa*)

bring the column of largest norm into the pivot position.

compute the Householder transformation to reduce the j-th column of a to a multiple of the j-th unit vector.

apply the transformation to the remaining columns and update the norms.

12.82.3.15 void lm_qrsolv (int *n*, double * *r*, int *ldr*, int * *ipvt*, double * *diag*, double * *qtb*, double * *x*, double * *sdiag*, double * *wa*)

determine a Givens rotation which eliminates the appropriate element in the current row of d.

compute the modified diagonal element of r and the modified element of ((q transpose)*b,0).

accumulate the tranformation in the row of s.

store the diagonal element of s and restore the corresponding diagonal element of r.

- 12.82.3.16 `void lmcurve_evaluate (const double * par, int m_dat, const void * data, double * fvec, int * info)`
- 12.82.3.17 `void lmcurve_fit (int n_par, double * par, int m_dat, const double * t, const double * y, double(*) (double t, const double * par, int obsNum) f, const lm_control_struct * control, lm_status_struct * status)`
- 12.82.3.18 `void lmmmin (int n_par, double * par, int m_dat, const void * data, void(*) (const double * par, int m_dat, const void * data, double * fvec, int * info) evaluate, const lm_control_struct * control, lm_status_struct * status, void(*) (int n_par, const double * par, int m_dat, const void * data, const double * fvec, int printflags, int iflag, int iter, int nfev) printout)`
- 12.82.3.19 `double nonlinearFunc (double t, const double * par, int obsNum)`
- 12.82.3.20 `void plot1 (double * epoch, double * f0, float * yerr1, float * yerr2, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.21 `void plot2 (double * epoch, double * f0, double * f0e, int * nFit, int * id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.22 `void plot3 (double * epoch, double * f1, double * f1e, int * nFit, int * id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.23 `void plot4 (double * epoch, double * f0, double * f0e, int * nFit, int * id, int n, double plotOffset, float * gt, int ngt, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.24 `void plot5 (double * epoch, double * f0, double * f0e, int * nFit, int * id, int n, double plotOffset, float * gt, int ngt, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.25 `void plot6 (double * epoch, double * f1, double * f1e, int * nFit, int * id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.26 `void plot7 (double * plotResX, double * plotResY, double * plotResE, int nplotVal, double plotOffset, int combine, int pos, int nplot, double start, double end, double psrF0, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.27 `void plot8 (double * epoch, double * f0, float * yerr1, float * yerr2, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`
- 12.82.3.28 `void plot9 (double * epoch, double * f0, double * f0e, int * nFit, int * id, int n, double plotOffset, int combine, int pos, int nplot, float yscale_min, float yscale_max, int yscale_set, float minx, float maxx)`

12.82.4 Variable Documentation

- 12.82.4.1 `int global_fitf0`
- 12.82.4.2 `int global_fitf1`
- 12.82.4.3 `double global_footer`
- 12.82.4.4 `glitchS* global_glitch`
- 12.82.4.5 `double global_header`
- 12.82.4.6 `int global_nglt`
- 12.82.4.7 `double global_valf0`

12.82.4.8 double global_valf1

12.82.4.9 const lm_control_struct lm_control_double

Initial value:

```
= {
    LM_USERTOL, LM_USERTOL, LM_USERTOL, LM_USERTOL, 100., 100, 1, 0
}
```

12.82.4.10 const lm_control_struct lm_control_float

Initial value:

```
= {
    1.e-7, 1.e-7, 1.e-7, 1.e-7, 100., 100, 0, 0 }
```

12.82.4.11 const char* lm_infmsg[]

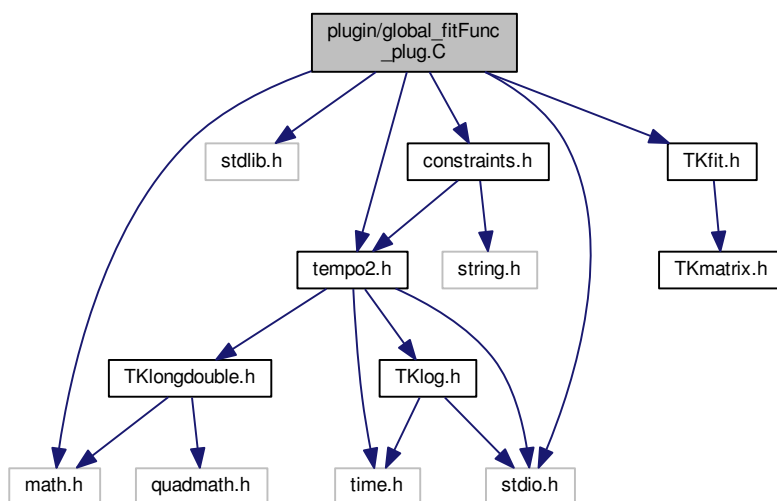
12.82.4.12 const char* lm_shortmsg[]

12.82.4.13 const char* plugVersionCheck = (char *)TEMPO2_h_VER

12.83 plugin/global_fitFunc_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <tempo2.h>
#include <math.h>
#include "TKfit.h"
#include "constraints.h"
```

Include dependency graph for global_fitFunc_plug.C:



Functions

- void [globalFITfuncs](#) (double x, double afunc[], int ma, [pulsar](#) *psr, int ipos)
- int [pluginFitFunc](#) ([pulsar](#) *psr, int npsr, int writeModel)

Variables

- int [gnpsr](#)
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.83.1 Function Documentation

12.83.1.1 void [globalFITfuncs](#) (double x, double afunc[], int ma, [pulsar](#) * psr, int ipos)

12.83.1.2 int [pluginFitFunc](#) ([pulsar](#) * psr, int npsr, int writeModel)

12.83.2 Variable Documentation

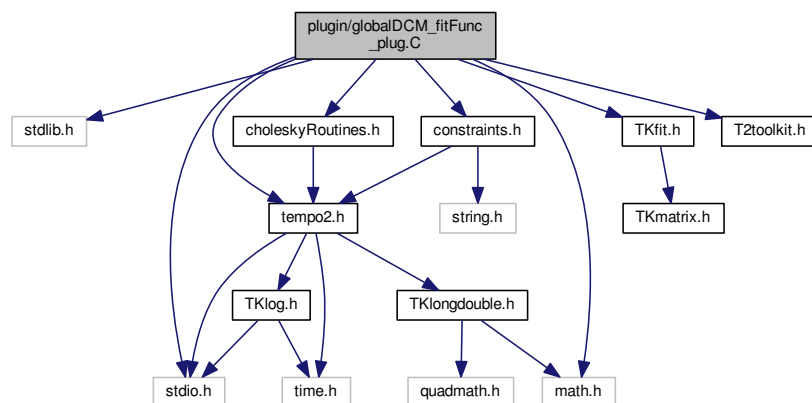
12.83.2.1 int [gnpsr](#)

12.83.2.2 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.84 plugin/globalDCM_fitFunc_plug.C File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <tempo2.h>
#include <math.h>
#include "TKfit.h"
#include "T2toolkit.h"
#include "choleskyRoutines.h"
#include "constraints.h"
```

Include dependency graph for globalDCM_fitFunc_plug.C:



Functions

- void [globalFITfuncs](#) (double x, double afunc[], int ma, [pulsar](#) *psr, int ipos)

- void [multMatrix2](#) (double **idcm, double **u, int ndata, int npol, double **uout)
- void [multMatrixVec2](#) (double **idcm, double *b, int ndata, double *bout)
- void [TKsingularValueDecomposition_Isq2](#) (double **designMatrix, int n, int [nf](#), double **v, double *w, double **u)
- void [TKbacksubstitution_svd2](#) (double **V, double *w, double **U, double *b, double *x, int n, int [nf](#))
- double [TKpythag2](#) (double a, double b)
- void [readUinv](#) (int p, double **uinv, [pulsar](#) *psr, double *x, double *y, double *sig, int count, int nconstraints, int *ip)
- void [TKbidiagonal2](#) (double **a, double *anorm, int ndata, int nfit, double **v, double *w, double **u, double *rv1)
- void [formCholeskyMatrix2](#) (double *c, double *resx, double *resy, double *rese, int np, int nconstraints, double **uinv)
- int [pluginFitFunc](#) ([pulsar](#) *psr, int npsr, int writeModel)

Variables

- int [gnpsr](#)
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.84.1 Function Documentation

- 12.84.1.1 void [formCholeskyMatrix2](#) (double * *c*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nconstraints*, double ** *uinv*)
- 12.84.1.2 void [globalFITfuncs](#) (double *x*, double *afunc*[], int *ma*, [pulsar](#) * *psr*, int *ipos*)
- 12.84.1.3 void [multMatrix2](#) (double ** *idcm*, double ** *u*, int *ndata*, int *npol*, double ** *uout*)
- 12.84.1.4 void [multMatrixVec2](#) (double ** *idcm*, double * *b*, int *ndata*, double * *bout*)
- 12.84.1.5 int [pluginFitFunc](#) ([pulsar](#) * *psr*, int *npsr*, int *writeModel*)
- 12.84.1.6 void [readUinv](#) (int *p*, double ** *uinv*, [pulsar](#) * *psr*, double * *x*, double * *y*, double * *sig*, int *count*, int *nconstraints*, int * *ip*)
- 12.84.1.7 void [TKbacksubstitution_svd2](#) (double ** *V*, double * *w*, double ** *U*, double * *b*, double * *x*, int *n*, int *nf*)
- 12.84.1.8 void [TKbidiagonal2](#) (double ** *a*, double * *anorm*, int *ndata*, int *nfit*, double ** *v*, double * *w*, double ** *u*, double * *rv1*)
- 12.84.1.9 double [TKpythag2](#) (double *a*, double *b*)
- 12.84.1.10 void [TKsingularValueDecomposition_Isq2](#) (double ** *designMatrix*, int *n*, int *nf*, double ** *v*, double * *w*, double ** *u*)

12.84.2 Variable Documentation

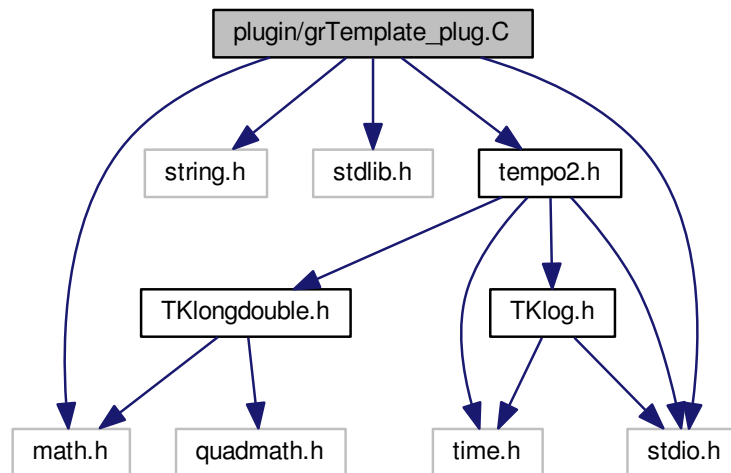
- 12.84.2.1 int [gnpsr](#)
- 12.84.2.2 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.85 plugin/grTemplate_plug.C File Reference

```
#include <stdio.h>
```

```
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for grTemplate_plug.C:



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.85.1 Function Documentation

12.85.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.85.1.2 void [help](#) ()

12.85.2 Variable Documentation

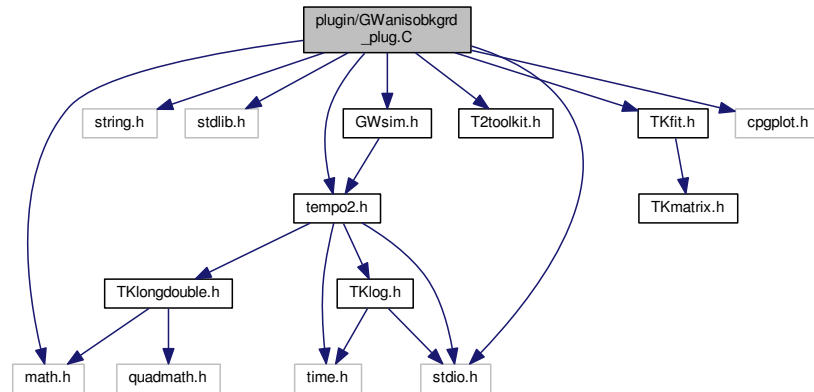
12.85.2.1 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.86 plugin/GWanisobkgrd_plug.C File Reference

```
#include <stdio.h>
```

```
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "GWsim.h"
#include "TKfit.h"
#include <cpgplot.h>
```

Include dependency graph for GWanisobkgrd_plug.C:



Functions

- void `doPlot` (`pulsar *psr`, int `npsr`, `gwSrc *gw`, `longdouble **gwRes`, `longdouble` `timeOffset`, int `ngw`, `longdouble` `tspan`, `longdouble` `alpha`, `longdouble` `gwAmp`)
- `longdouble` `getTspan` (`pulsar *psr`, int `npsr`)
- void `plotResiduals` (`pulsar *psr`, `longdouble **gwRes`, int `p`, `longdouble` `timeOffset`, int `plotType`)
- void `plotSpectrum` (`gwSrc *gw`, int `ngw`, `longdouble` `tspan`, `longdouble` `alpha`, `longdouble` `gwAmp`)
- void `plotPosn` (`pulsar *psr`, int `npsr`, `gwSrc *gw`, int `ngw`)
- void `draw_grid` (double `start_gl`, double `end_gl`, double `start_gb`, double `end_gb`, double `gstep`, double `bstep`, int `celestialCoords`)
- void `convertXY_celestial` (double `raj`, double `decj`, double `*retx`, double `*rety`)
- void `help` ()
- int `graphicalInterface` (int `argc`, char `*argv[]`, `pulsar *psr`, int `*npsr`)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.86.1 Function Documentation

12.86.1.1 void `convertXY_celestial` (double *raj*, double *decj*, double * *retx*, double * *rety*)

12.86.1.2 void `doPlot` (`pulsar *psr`, int *npsr*, `gwSrc *gw`, `longdouble ** gwRes`, `longdouble` *timeOffset*, int *ngw*, `longdouble` *tspan*, `longdouble` *alpha*, `longdouble` *gwAmp*)

12.86.1.3 void `draw_grid` (double *start_gl*, double *end_gl*, double *start_gb*, double *end_gb*, double *gstep*, double *bstep*, int *celestialCoords*)

12.86.1.4 `longdouble getTspan (pulsar * psr, int npsr)`

12.86.1.5 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.86.1.6 `void help ()`

12.86.1.7 `void plotPosn (pulsar * psr, int npsr, gwSrc * gw, int ngw)`

12.86.1.8 `void plotResiduals (pulsar * psr, longdouble ** gwRes, int p, longdouble timeOffset, int plotType)`

12.86.1.9 `void plotSpectrum (gwSrc * gw, int ngw, longdouble tspan, longdouble alpha, longdouble gwAmp)`

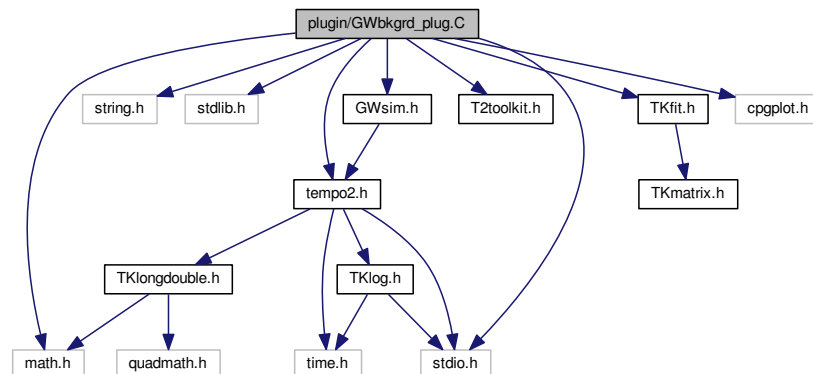
12.86.2 Variable Documentation

12.86.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.87 plugin/GWbkgrd_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "GWsim.h"
#include "TKfit.h"
#include <cpgplot.h>
```

Include dependency graph for GWbkgrd_plug.C:



Functions

- `void doPlot (pulsar *psr, int npsr, gwSrc *gw, longdouble **gwRes, longdouble timeOffset, int ngw, longdouble tspan, longdouble alpha, longdouble gwAmp)`
- `longdouble getTspan (pulsar *psr, int npsr)`
- `void plotResiduals (pulsar *psr, longdouble **gwRes, int p, longdouble timeOffset, int plotType)`
- `void plotSpectrum (gwSrc *gw, int ngw, longdouble tspan, longdouble alpha, longdouble gwAmp)`
- `void plotPosn (pulsar *psr, int npsr, gwSrc *gw, int ngw)`

- void `draw_grid` (double `start_gl`, double `end_gl`, double `start_gb`, double `end_gb`, double `gstep`, double `bstep`, int `celestialCoords`)
- void `convertXY_celestial` (double `raj`, double `decj`, double `*retx`, double `*rety`)
- void `help` ()
- int `graphicalInterface` (int `argc`, char `*argv[]`, `pulsar` `*psr`, int `*npsr`)

Variables

- const char `* plugVersionCheck` = `TEMPO2_h_VER`

12.87.1 Function Documentation

12.87.1.1 void `convertXY_celestial` (double *raj*, double *decj*, double ** retx*, double ** rety*)

12.87.1.2 void `doPlot` (`pulsar` ** psr*, int *npsr*, `gwSrc` ** gw*, `longdouble` *** gwRes*, `longdouble` *timeOffset*, int *ngw*, `longdouble` *tspan*, `longdouble` *alpha*, `longdouble` *gwAmp*)

12.87.1.3 void `draw_grid` (double *start_gl*, double *end_gl*, double *start_gb*, double *end_gb*, double *gstep*, double *bstep*, int *celestialCoords*)

12.87.1.4 `longdouble` `getTspan` (`pulsar` ** psr*, int *npsr*)

12.87.1.5 int `graphicalInterface` (int *argc*, char ** argv[]*, `pulsar` ** psr*, int ** npsr*)

12.87.1.6 void `help` ()

12.87.1.7 void `plotPosn` (`pulsar` ** psr*, int *npsr*, `gwSrc` ** gw*, int *ngw*)

12.87.1.8 void `plotResiduals` (`pulsar` ** psr*, `longdouble` *** gwRes*, int *p*, `longdouble` *timeOffset*, int *plotType*)

12.87.1.9 void `plotSpectrum` (`gwSrc` ** gw*, int *ngw*, `longdouble` *tspan*, `longdouble` *alpha*, `longdouble` *gwAmp*)

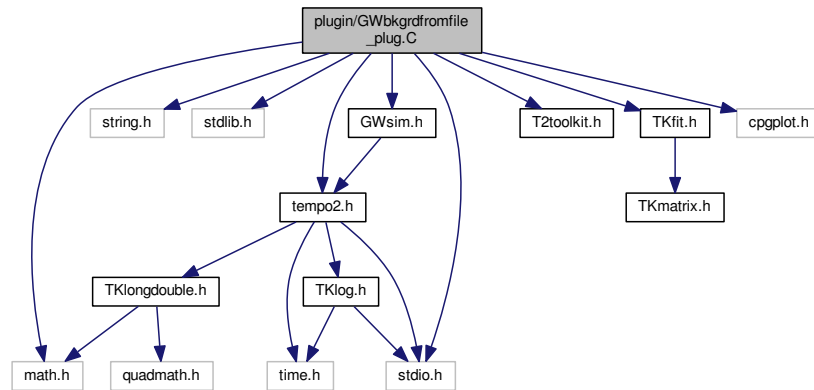
12.87.2 Variable Documentation

12.87.2.1 const char`* plugVersionCheck` = `TEMPO2_h_VER`

12.88 plugin/GWbkgrdfromfile_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include <cpgplot.h>
```

Include dependency graph for GWbkgrdfromfile_plug.C:



Functions

- void **doPlot** (**pulsar** *psr, int npsr, **gwSrc** *gw, **longdouble** **gwRes, **longdouble** timeOffset, int ngw, **longdouble** tspan)
- void **doGenPlot** (**pulsar** *psr, int npsr, **gwgeneralSrc** *gw, **longdouble** **gwRes, **longdouble** timeOffset, int ngw, **longdouble** tspan)
- **longdouble** **getTspan** (**pulsar** *psr, int npsr)
- void **plotResiduals** (**pulsar** *psr, **longdouble** **gwRes, int p, **longdouble** timeOffset, int plotType)
- void **plotSpectrum** (**gwSrc** *gw, int ngw, **longdouble** tspan)
- void **plotGenSpectrum** (**gwgeneralSrc** *gw, int ngw, **longdouble** tspan)
- void **plotPosn** (**pulsar** *psr, int npsr, **gwSrc** *gw, int ngw)
- void **plotGenPosn** (**pulsar** *psr, int npsr, **gwgeneralSrc** *gw, int ngw)
- void **draw_grid** (double start_gl, double end_gl, double start_gb, double end_gb, double gstep, double bstep, int celestialCoords)
- void **convertXY_celestial** (double raj, double decj, double *retx, double *rety)
- void **help** ()
- int **graphicalInterface** (int argc, char *argv[], **pulsar** *psr, int *npsr)

Variables

- int **NGWmax** =10000
- const char * **plugVersionCheck** = **TEMPO2_h_VER**

12.88.1 Function Documentation

12.88.1.1 void **convertXY_celestial** (double *raj*, double *decj*, double * *retx*, double * *rety*)

12.88.1.2 void **doGenPlot** (**pulsar** * *psr*, int *npsr*, **gwgeneralSrc** * *gw*, **longdouble** ** *gwRes*, **longdouble** *timeOffset*, int *ngw*, **longdouble** *tspan*)

12.88.1.3 void **doPlot** (**pulsar** * *psr*, int *npsr*, **gwSrc** * *gw*, **longdouble** ** *gwRes*, **longdouble** *timeOffset*, int *ngw*, **longdouble** *tspan*)

12.88.1.4 void **draw_grid** (double *start_gl*, double *end_gl*, double *start_gb*, double *end_gb*, double *gstep*, double *bstep*, int *celestialCoords*)

- 12.88.1.5 `longdouble getTspan (pulsar * psr, int npsr)`
- 12.88.1.6 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`
- 12.88.1.7 `void help ()`
- 12.88.1.8 `void plotGenPosn (pulsar * psr, int npsr, gwgeneralSrc * gw, int ngw)`
- 12.88.1.9 `void plotGenSpectrum (gwgeneralSrc * gw, int ngw, longdouble tspan)`
- 12.88.1.10 `void plotPosn (pulsar * psr, int npsr, gwSrc * gw, int ngw)`
- 12.88.1.11 `void plotResiduals (pulsar * psr, longdouble ** gwRes, int p, longdouble timeOffset, int plotType)`
- 12.88.1.12 `void plotSpectrum (gwSrc * gw, int ngw, longdouble tspan)`

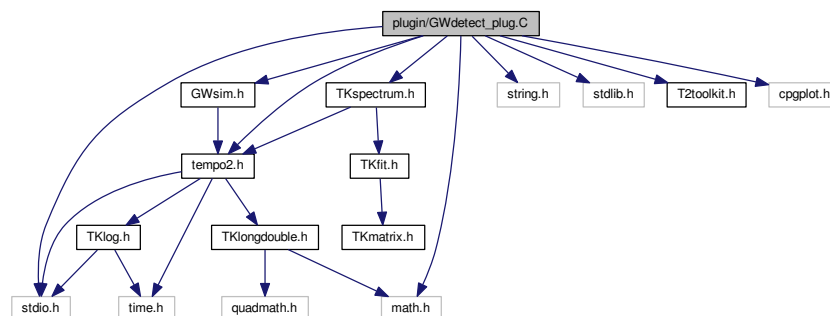
12.88.2 Variable Documentation

- 12.88.2.1 `int NGWmax = 10000`
- 12.88.2.2 `const char* plugVersionCheck = TEMPO2_h_VER`

12.89 plugin/GWdetect_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "GWsim.h"
#include "T2toolkit.h"
#include <cpgplot.h>
#include "tempo2.h"
#include "TKspectrum.h"
```

Include dependency graph for GWdetect_plug.C:



Functions

- void [searchGridPos](#) (double *dlat*, double *dlong*, int *gridPos*, pulsar * *psr*, int *npsr*, char * *addname*)
- void [help](#) ()
- int [graphicalInterface](#) (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)

12.89.1 Function Documentation

12.89.1.1 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

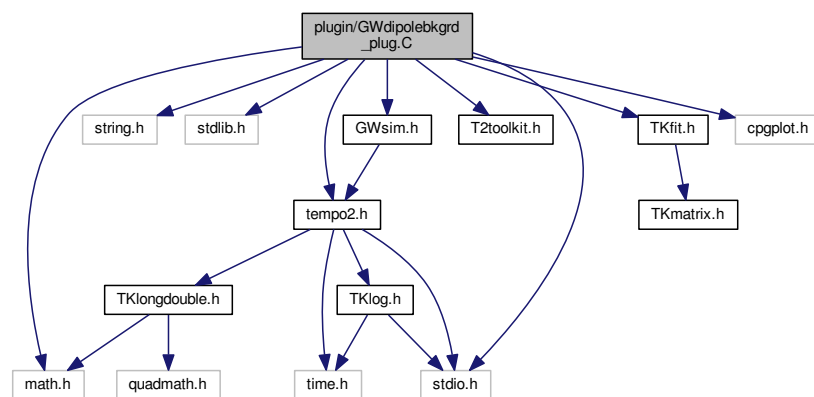
12.89.1.2 `void help ()`

12.89.1.3 `void searchGridPos (double dlat, double dlong, int gridPos, pulsar * psr, int npsr, char * addname)`

12.90 plugin/GWdipolebkgrd_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "GWsim.h"
#include "TKfit.h"
#include <cpgplot.h>
```

Include dependency graph for GWdipolebkgrd_plug.C:



Functions

- void `doPlot` (`pulsar *psr`, `int npsr`, `gwSrc *gw`, `longdouble **gwRes`, `longdouble timeOffset`, `int ngw`, `longdouble tspan`, `longdouble alpha`, `longdouble gwAmp`)
- `longdouble getTspan` (`pulsar *psr`, `int npsr`)
- void `plotResiduals` (`pulsar *psr`, `longdouble **gwRes`, `int p`, `longdouble timeOffset`, `int plotType`)
- void `plotSpectrum` (`gwSrc *gw`, `int ngw`, `longdouble tspan`, `longdouble alpha`, `longdouble gwAmp`)
- void `plotPosn` (`pulsar *psr`, `int npsr`, `gwSrc *gw`, `int ngw`)
- void `draw_grid` (`double start_gl`, `double end_gl`, `double start_gb`, `double end_gb`, `double gstep`, `double bstep`, `int celestialCoords`)
- void `convertXY_celestial` (`double raj`, `double decj`, `double *retx`, `double *rety`)
- void `help` ()
- int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.90.1 Function Documentation

12.90.1.1 void convertXY_celestial (double *raj*, double *decj*, double * *retx*, double * *rety*)

12.90.1.2 void doPlot (pulsar * *psr*, int *npsr*, gwSrc * *gw*, longdouble ** *gwRes*, longdouble *timeOffset*, int *ngw*, longdouble *tspan*, longdouble *alpha*, longdouble *gwAmp*)

12.90.1.3 void draw_grid (double *start_gl*, double *end_gl*, double *start_gb*, double *end_gb*, double *gstep*, double *bstep*, int *celestialCoords*)

12.90.1.4 longdouble getTspan (pulsar * *psr*, int *npsr*)

12.90.1.5 int graphicalInterface (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)

12.90.1.6 void help ()

12.90.1.7 void plotPosn (pulsar * *psr*, int *npsr*, gwSrc * *gw*, int *ngw*)

12.90.1.8 void plotResiduals (pulsar * *psr*, longdouble ** *gwRes*, int *p*, longdouble *timeOffset*, int *plotType*)

12.90.1.9 void plotSpectrum (gwSrc * *gw*, int *ngw*, longdouble *tspan*, longdouble *alpha*, longdouble *gwAmp*)

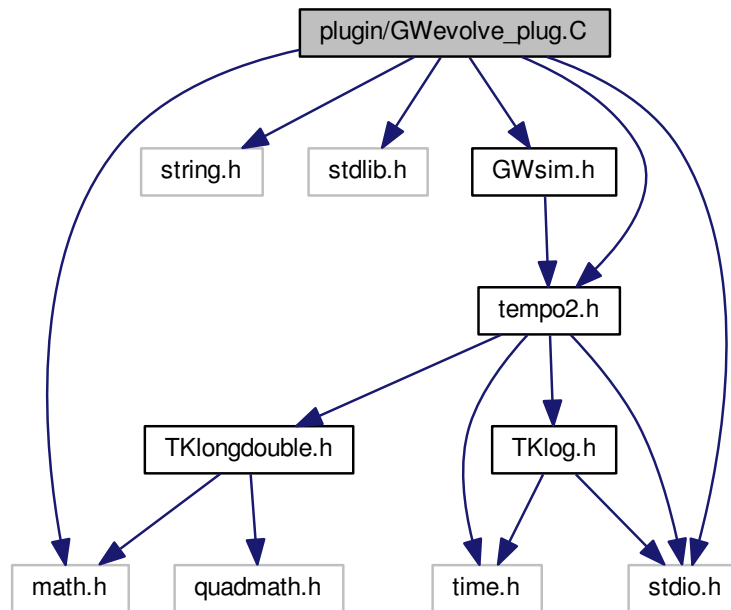
12.90.2 Variable Documentation

12.90.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.91 plugin/GWevolve_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
```

Include dependency graph for GWevolve_plug.C:



Macros

- `#define SPEED_LIGHT 299792458.0 /* Speed of light (m/s) */`
- `#define SOLAR_MASS 1.98892e30 /* Mass of Sun (kg) */`
- `#define BIG_G 6.673e-11 /* Gravitational constant */`
- `#define PCM 3.08568025e16 /* one parsec in meters */`
- `#define MAX_VAL 6000`
- `#define MAXSTP 10000`
- `#define TINY 1.0e-30`
- `#define SIGN(a, b) ((b) >= 0.0 ? fabs(a) : -fabs(a))`
- `#define SAFETY 0.9`
- `#define PGROW -0.2`
- `#define PSHRNK -0.25`
- `#define ERRCON 1.89e-4`
- `#define NR_END 1`
- `#define FMAX(x, y) ((x < y) ? y : x)`
- `#define FREE_ARG char*`

Functions

- `void ThetaEderivs (double t, double *vals, double *derivs)`
- `void setup3C66B (double *e0, double *theta0, double *mjdOmega0, double *mjdObs0, double *mjdLast, int *nObs, double *psrdist, double *dist, double *omega0, double *mu, double *mc, double *phi)`
- `double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)`
- `void setupTest (double *e0, double *theta0, double *mjdOmega0, double *mjdObs0, double *mjdLast, int *nObs, double *psrdist, double *dist, double *omega0, double *mu, double *mc, double *phi)`

- void [RungeKuttaStep](#) (double *y, double *dydx, int n, double *x, double htry, double eps, double *yscal, double *hdid, double *hnext, void(*[derivs](#))(double, double *, double *))
- void [RungeKuttaCashKarp](#) (double *y, double *dydx, int n, double x, double h, double *yout, double *yerr, void(*[derivs](#))(double, double *, double *))
- void [ode](#) (double ystart[], int nvar, double x1, double x2, double eps, double h1, double hmin, int *nok, int *nbad, void(*[derivs](#))(double, double[], double[]), void(*[RungeKuttaStep](#))(double[], double[], int, double *, double, double, double[], double *, double *, void(*) (double, double[], double[])))
- void [help](#) ()
- [longdouble calcAmp](#) (gwSrc *gw)
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)
- void [nrerror](#) (char error_text[])
- void [free_vector](#) (double *v, long nl, long nh)
- double * [vector](#) (long nl, long nh)
- void [ode](#) (double *ystart, int nvar, double x1, double x2, double eps, double h1, double hmin, int *nok, int *nbad, void(*[derivs](#))(double, double *, double *), void(*[RungeKuttaStep](#))(double *, double *, int, double *, double, double, double *, double *, double *, void(*) (double, double *, double *)))

Variables

- double [constA0](#)
- double [const2](#)
- int [kmax](#)
- int [kount](#)
- double * [xp](#)
- double ** [yp](#)
- double [dxsav](#)
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.91.1 Macro Definition Documentation

- 12.91.1.1 `#define BIG_G 6.673e-11 /* Gravitational constant */`
- 12.91.1.2 `#define ERRCON 1.89e-4`
- 12.91.1.3 `#define FMAX(x, y) ((x<y)?y:x)`
- 12.91.1.4 `#define FREE_ARG char*`
- 12.91.1.5 `#define MAX_VAL 6000`
- 12.91.1.6 `#define MAXSTP 10000`
- 12.91.1.7 `#define NR_END 1`
- 12.91.1.8 `#define PCM 3.08568025e16 /* one parsec in meters */`
- 12.91.1.9 `#define PGROW -0.2`
- 12.91.1.10 `#define PSHRNK -0.25`
- 12.91.1.11 `#define SAFETY 0.9`
- 12.91.1.12 `#define SIGN(a, b) ((b) >= 0.0 ? fabs(a) : -fabs(a))`

12.91.1.13 `#define SOLAR_MASS 1.98892e30 /* Mass of Sun (kg) */`

12.91.1.14 `#define SPEED_LIGHT 299792458.0 /* Speed of light (m/s) */`

12.91.1.15 `#define TINY 1.0e-30`

12.91.2 Function Documentation

12.91.2.1 `longdouble calcAmp (gwSrc * gw)`

12.91.2.2 `void free_vector (double * v, long nl, long nh)`

12.91.2.3 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.91.2.4 `void help ()`

12.91.2.5 `void nrerror (char error_text[])`

12.91.2.6 `void ode (double ystart[], int nvar, double x1, double x2, double eps, double h1, double hmin, int * nok, int * nbad, void(*) (double, double[], double[]) derivs, void(*) (double[], double[], int, double *, double, double, double[], double *, double *, void(*) (double, double[], double[])) RungeKuttaStep)`

12.91.2.7 `void ode (double * ystart, int nvar, double x1, double x2, double eps, double h1, double hmin, int * nok, int * nbad, void(*) (double, double *, double *) derivs, void(*) (double *, double *, int, double *, double, double, double *, double *, double *, void(*) (double, double *, double *)) RungeKuttaStep)`

12.91.2.8 `double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)`

12.91.2.9 `void RungeKuttaCashKarp (double * y, double * dydx, int n, double x, double h, double * youf, double * yerr, void(*) (double, double *, double *) derivs)`

12.91.2.10 `void RungeKuttaStep (double * y, double * dydx, int n, double * x, double htry, double eps, double * yscal, double * hdid, double * hnext, void(*) (double, double *, double *) derivs)`

12.91.2.11 `void setup3C66B (double * e0, double * theta0, double * mjdOmega0, double * mjdObs0, double * mjdLast, int * nObs, double * psrdist, double * dist, double * omega0, double * mu, double * mc, double * phi)`

12.91.2.12 `void setupTest (double * e0, double * theta0, double * mjdOmega0, double * mjdObs0, double * mjdLast, int * nObs, double * psrdist, double * dist, double * omega0, double * mu, double * mc, double * phi)`

12.91.2.13 `void ThetaEderivs (double t, double * vals, double * derivs)`

12.91.2.14 `double* vector (long nl, long nh)`

12.91.3 Variable Documentation

12.91.3.1 `double const2`

12.91.3.2 `double constA0`

12.91.3.3 `double dxsav`

12.91.3.4 `int kmax`

12.91.3.5 `int kount`

12.91.3.6 `const char* plugVersionCheck = TEMPO2_h_VER`

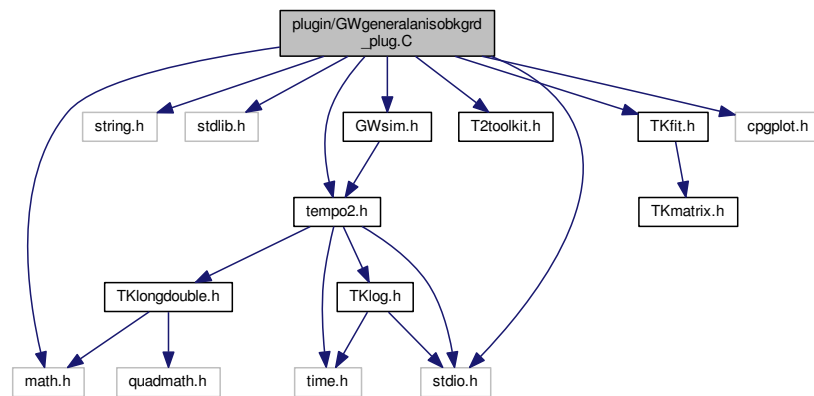
12.91.3.7 `double* xp`

12.91.3.8 `double ** yp`

12.92 plugin/GWgeneralanisobkgrd_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "GWsim.h"
#include "TKfit.h"
#include <cpgplot.h>
```

Include dependency graph for GWgeneralanisobkgrd_plug.C:



Functions

- void `doPlot` (`pulsar *psr`, `int npsr`, `gwgeneralSrc *gw`, `longdouble **gwRes`, `longdouble timeOffset`, `int *numsources`, `longdouble tspan`, `gwgenSpec gwAmps`)
- `longdouble getTspan` (`pulsar *psr`, `int npsr`)
- void `plotResiduals` (`pulsar *psr`, `longdouble **gwRes`, `int p`, `longdouble timeOffset`, `int plotType`)
- void `plotSpectrum` (`gwgeneralSrc *gw`, `int *numsources`, `longdouble tspan`, `gwgenSpec gwAmps`)
- void `plotPosn` (`pulsar *psr`, `int npsr`, `gwgeneralSrc *gw`, `int *numsources`)
- void `draw_grid` (`double start_gl`, `double end_gl`, `double start_gb`, `double end_gb`, `double gstep`, `double bstep`, `int celestialCoords`)
- void `convertXY_celestial` (`double raj`, `double decj`, `double *retx`, `double *rety`)
- void `help` ()
- int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

Variables

- `const char * plugVersionCheck = TEMPO2_h_VER`

12.92.1 Function Documentation

12.92.1.1 void convertXY_celestial (double *raj*, double *decj*, double * *retx*, double * *rety*)

12.92.1.2 void doPlot (pulsar * *psr*, int *npsr*, gwgeneralSrc * *gw*, longdouble ** *gwRes*, longdouble *timeOffset*, int * *numsources*, longdouble *tspan*, gwgenSpec *gwAmps*)

12.92.1.3 void draw_grid (double *start_gl*, double *end_gl*, double *start_gb*, double *end_gb*, double *gstep*, double *bstep*, int *celestialCoords*)

12.92.1.4 longdouble getTspan (pulsar * *psr*, int *npsr*)

12.92.1.5 int graphicalInterface (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)

12.92.1.6 void help ()

12.92.1.7 void plotPosn (pulsar * *psr*, int *npsr*, gwgeneralSrc * *gw*, int * *numsources*)

12.92.1.8 void plotResiduals (pulsar * *psr*, longdouble ** *gwRes*, int *p*, longdouble *timeOffset*, int *plotType*)

12.92.1.9 void plotSpectrum (gwgeneralSrc * *gw*, int * *numsources*, longdouble *tspan*, gwgenSpec *gwAmps*)

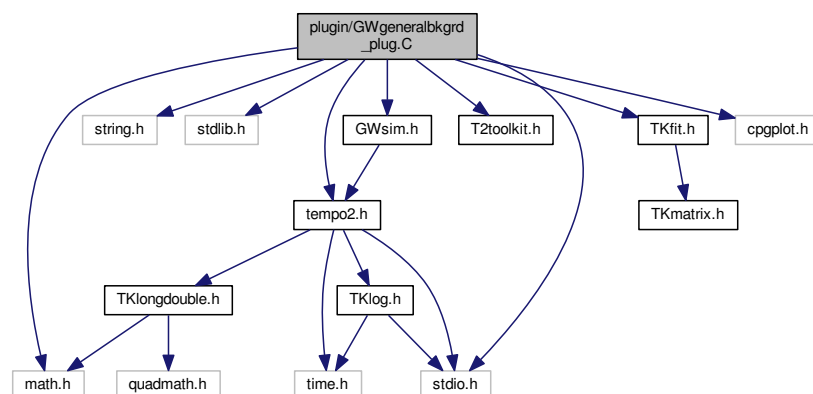
12.92.2 Variable Documentation

12.92.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.93 plugin/GWgeneralbkgrd_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "GWsim.h"
#include "TKfit.h"
#include <cpgplot.h>
```

Include dependency graph for GWgeneralbkgrd_plug.C:



Functions

- void `doPlot` (`pulsar *psr`, `int npsr`, `gwgeneralSrc *gw`, `longdouble **gwRes`, `longdouble timeOffset`, `int *numsources`, `longdouble tspan`, `gwgenSpec gwAmps`)
- `longdouble` `getTspan` (`pulsar *psr`, `int npsr`)
- void `plotResiduals` (`pulsar *psr`, `longdouble **gwRes`, `int p`, `longdouble timeOffset`, `int plotType`)
- void `plotSpectrum` (`gwgeneralSrc *gw`, `int *numsources`, `longdouble tspan`, `gwgenSpec gwAmps`)
- void `plotPosn` (`pulsar *psr`, `int npsr`, `gwgeneralSrc *gw`, `int *numsources`)
- void `draw_grid` (`double start_gl`, `double end_gl`, `double start_gb`, `double end_gb`, `double gstep`, `double bstep`, `int celestialCoords`)
- void `convertXY_celestial` (`double raj`, `double decj`, `double *retx`, `double *rety`)
- void `help` ()
- int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.93.1 Function Documentation

- 12.93.1.1 void `convertXY_celestial` (`double raj`, `double decj`, `double * retx`, `double * rety`)
- 12.93.1.2 void `doPlot` (`pulsar * psr`, `int npsr`, `gwgeneralSrc * gw`, `longdouble ** gwRes`, `longdouble timeOffset`, `int * numsources`, `longdouble tspan`, `gwgenSpec gwAmps`)
- 12.93.1.3 void `draw_grid` (`double start_gl`, `double end_gl`, `double start_gb`, `double end_gb`, `double gstep`, `double bstep`, `int celestialCoords`)
- 12.93.1.4 `longdouble` `getTspan` (`pulsar * psr`, `int npsr`)
- 12.93.1.5 int `graphicalInterface` (`int argc`, `char * argv[]`, `pulsar * psr`, `int * npsr`)
- 12.93.1.6 void `help` ()
- 12.93.1.7 void `plotPosn` (`pulsar * psr`, `int npsr`, `gwgeneralSrc * gw`, `int * numsources`)
- 12.93.1.8 void `plotResiduals` (`pulsar * psr`, `longdouble ** gwRes`, `int p`, `longdouble timeOffset`, `int plotType`)
- 12.93.1.9 void `plotSpectrum` (`gwgeneralSrc * gw`, `int * numsources`, `longdouble tspan`, `gwgenSpec gwAmps`)

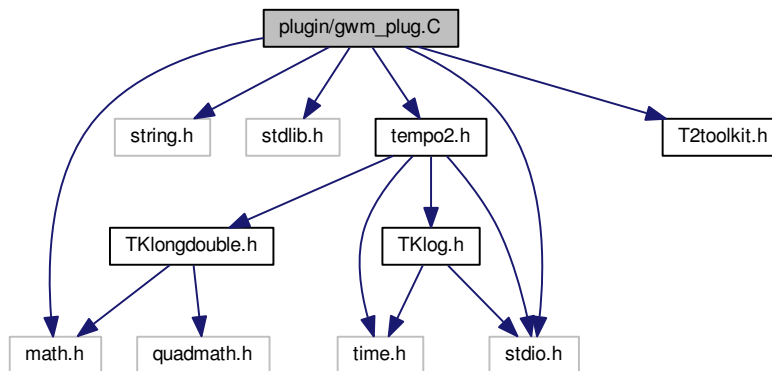
12.93.2 Variable Documentation

- 12.93.2.1 const char* `plugVersionCheck` = `TEMPO2_h_VER`

12.94 plugin/gwm_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
```

Include dependency graph for gwm_plug.C:



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = (char *)[TEMPO2_h_VER](#)

12.94.1 Function Documentation

12.94.1.1 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.94.1.2 void [help](#) ()

12.94.2 Variable Documentation

12.94.2.1 const char* [plugVersionCheck](#) = (char *)[TEMPO2_h_VER](#)

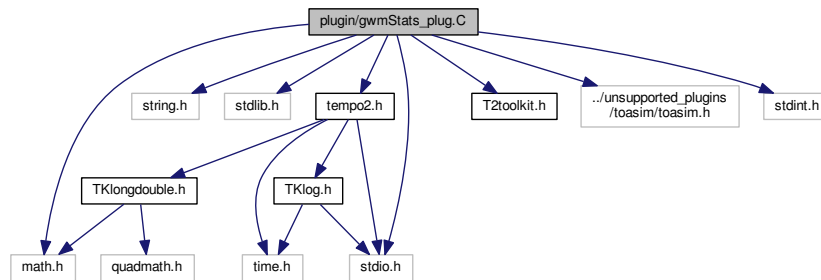
12.95 plugin/gwmStats_plug.C File Reference

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "../unsupported_plugins/toasim/toasim.h"

```


Include dependency graph for gwmStats_plug.C:



Macros

- `#define MAX_CORR 3`

Functions

- void `calculateAngularFactors` (`pulsar *psr`)
- double `calculatedD` (`pulsar *psr`)
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], `pulsar *psr`, int *npsr)

Variables

- const char * `plugVersionCheck` = (char *)TEMPO2_h_VER

12.95.1 Macro Definition Documentation

12.95.1.1 `#define MAX_CORR 3`

12.95.2 Function Documentation

12.95.2.1 void `calculateAngularFactors` (`pulsar * psr`)

12.95.2.2 double `calculatedD` (`pulsar * psr`)

12.95.2.3 int `graphicalInterface` (int *argc*, char * *argv*[], `pulsar * psr`, int * *npsr*)

12.95.2.4 void `help` ()

12.95.3 Variable Documentation

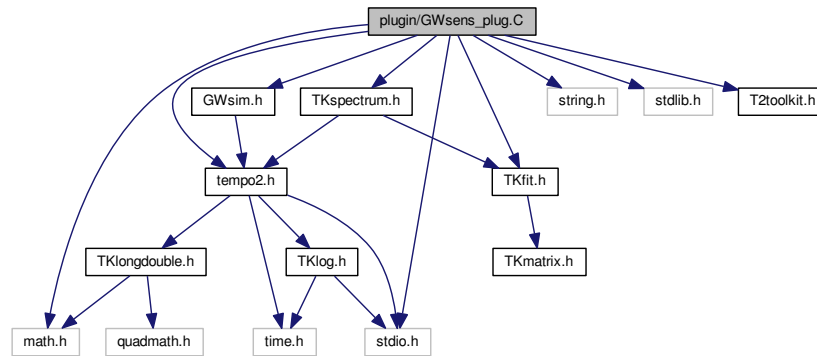
12.95.3.1 const char* `plugVersionCheck` = (char *)TEMPO2_h_VER

12.96 plugin/GWsens_plug.C File Reference

```
#include <stdio.h>
```

```
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKspectrum.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include "GWsim.h"
```

Include dependency graph for GWsens_plug.C:



Functions

- void `doPlugin` (`pulsar *psr`, `int npsr`, `int doFitV`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`)
- void `getSensCurv` (`pulsar *psr`, `int npsr`, `double **resX`, `double **resY`, `double **resE`, `int *nObs`, `int doFitV`)
- int `detectSource` (`pulsar *psr`, `int npsr`, `double **resX`, `double **resY`, `double **resE`)
- void `help` ()
- int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

Variables

- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.96.1 Function Documentation

12.96.1.1 int `detectSource` (`pulsar *psr`, `int npsr`, `double **resX`, `double **resY`, `double **resE`)

12.96.1.2 void `doPlugin` (`pulsar *psr`, `int npsr`, `int doFitV`, `char parFile[MAX_PSR_VAL][MAX_FILELEN]`, `char timFile[MAX_PSR_VAL][MAX_FILELEN]`)

!!!!temporary marker for finding this location in file

12.96.1.3 void `getSensCurv` (`pulsar *psr`, `int npsr`, `double **resX`, `double **resY`, `double **resE`, `int *nObs`, `int doFitV`)

12.96.1.4 int `graphicalInterface` (`int argc`, `char *argv[]`, `pulsar *psr`, `int *npsr`)

12.96.1.5 void help ()

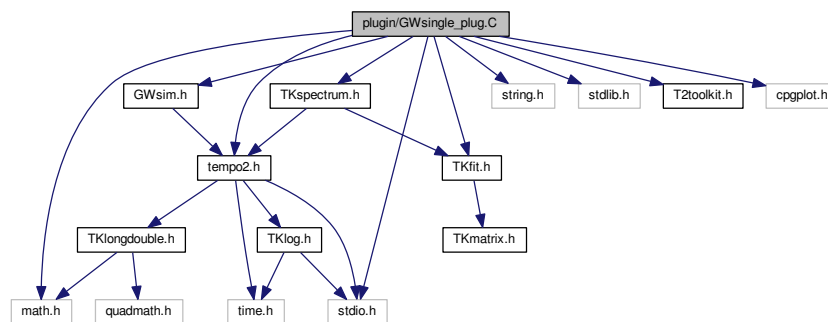
12.96.2 Variable Documentation

12.96.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.97 plugin/GWsingle_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include "TKspectrum.h"
#include <cpgplot.h>
```

Include dependency graph for GWsingle_plug.C:



Functions

- void `doPlot` (pulsar *psr, int npsr, gwSrc gw, longdouble **gwRes, longdouble timeOffset, longdouble tspan)
- longdouble `getTspan` (pulsar *psr, int npsr)
- void `plotResiduals` (pulsar *psr, longdouble **gwRes, int p, longdouble timeOffset, int plotType)
- void `plotSpectrum` (pulsar *psr, int p, longdouble timeOffset)
- void `plotPosn` (pulsar *psr, int npsr, gwSrc gw)
- void `draw_grid` (double start_gl, double end_gl, double start_gb, double end_gb, double gstep, double bstep, int celestialCoords)
- void `convertXY_celestial` (double raj, double decj, double *retx, double *rety)
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], pulsar *psr, int *npsr)

Variables

- const char * `plugVersionCheck` = TEMPO2_h_VER

12.97.1 Function Documentation

12.97.1.1 void convertXY_celestial (double *raj*, double *decj*, double * *retx*, double * *rety*)

12.97.1.2 void doPlot (pulsar * *psr*, int *npsr*, gwSrc *gw*, longdouble ** *gwRes*, longdouble *timeOffset*, longdouble *tspan*)

12.97.1.3 void draw_grid (double *start_gl*, double *end_gl*, double *start_gb*, double *end_gb*, double *gstep*, double *bstep*, int *celestialCoords*)

12.97.1.4 longdouble getTspan (pulsar * *psr*, int *npsr*)

12.97.1.5 int graphicalInterface (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)

12.97.1.6 void help ()

12.97.1.7 void plotPosn (pulsar * *psr*, int *npsr*, gwSrc *gw*)

12.97.1.8 void plotResiduals (pulsar * *psr*, longdouble ** *gwRes*, int *p*, longdouble *timeOffset*, int *plotType*)

12.97.1.9 void plotSpectrum (pulsar * *psr*, int *p*, longdouble *timeOffset*)

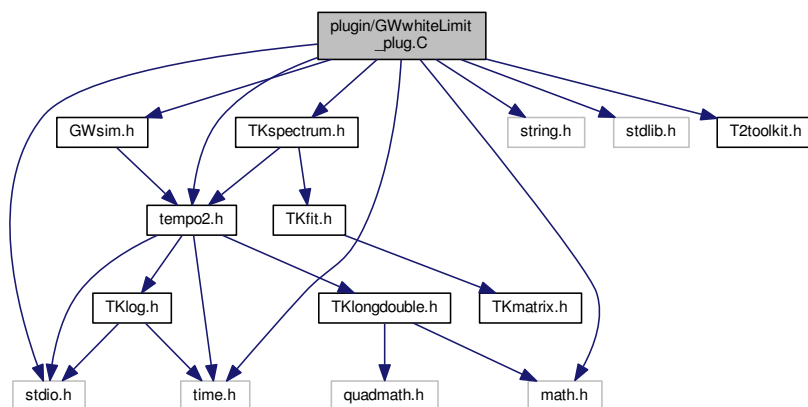
12.97.2 Variable Documentation

12.97.2.1 const char* plugVersionCheck = TEMPO2_h_VER

12.98 plugin/GWwhiteLimit_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "GWsim.h"
#include "T2toolkit.h"
#include "TKspectrum.h"
#include <time.h>
```

Include dependency graph for GWwhiteLimit_plug.C:



Macros

- `#define SIGN(a, b) ((b) >= 0.0 ? fabs(a) : -fabs(a))`
- `#define MAX_POLY 8`
- `#define MAX_FLAG 10`
- `#define MAX_FREQ 1000`
- `#define MAX_ITERATION 25000`

Functions

- void `getThreshold` (`pulsar` *psr, double *freqVal, int *nFreq, double *threshold, long *idum, int fast, int npsr, `longdouble` **Func)
- void `cumulativeHistogram` (double val[`MAX_ITERATION`][`MAX_FREQ`], int nval, double *freqVal, int nFreq, double limit, double *threshold)
- void `cumulativeHistogram2` (double *val, int nval, double limit, double *threshold)
- void `sortit` (int n, double array[], double rasort[])
- void `shuffle` (`longdouble` *R, double *toaE, `longdouble` *R2, double *toaE2, int N, long *idum)
- void `checkReal` (`pulsar` psr, double *freqVal, int *nFreq, double *threshold, double `alpha`)
- void `getLimits` (`pulsar` *psr, double *freqVal, int *nFreq, double *threshold, long *idum, int checkBackground, double `alpha`, double *dist, int distNum, double maxAmp, int fast, int npsr, int numberGW, `longdouble alpha`, `longdouble` **Func)
- void `setupPulsar` (`longdouble` ra_p, `longdouble` dec_p, `longdouble` *kp)
- void `GramSchmidt` (`longdouble` x[], `longdouble` y[], `longdouble` err[], int ObsAmt, int Npoly, `longdouble` CoeffArray[], int wtyn)
- void `writeCommands` (int argc, char *argv[])
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], `pulsar` *psr, int *npsr)
- void `GramSchmidt` (`longdouble` *x, `longdouble` *y, `longdouble` *err, int ObsAmt, int Npoly, `longdouble` *CoeffArray, int wtyn)

Variables

- double `storeVal` [`MAX_ITERATION`][`MAX_FREQ`]
- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.98.1 Macro Definition Documentation

12.98.1.1 `#define MAX_FLAG 10`

12.98.1.2 `#define MAX_FREQ 1000`

12.98.1.3 `#define MAX_ITERATION 25000`

12.98.1.4 `#define MAX_POLY 8`

12.98.1.5 `#define SIGN(a, b) ((b) >= 0.0 ? fabs(a) : -fabs(a))`

12.98.2 Function Documentation

12.98.2.1 void `checkReal` (`pulsar` *psr*, double * *freqVal*, int * *nFreq*, double * *threshold*, double *alpha*)

12.98.2.2 void `cumulativeHistogram` (double val[`MAX_ITERATION`][`MAX_FREQ`], int *nval*, double * *freqVal*, int *nFreq*, double *limit*, double * *threshold*)

- 12.98.2.3 void cumulativeHistogram2 (double * *val*, int *nval*, double *limit*, double * *threshold*)
- 12.98.2.4 void getLimits (pulsar * *psr*, double * *freqVal*, int * *nFreq*, double * *threshold*, long * *idum*, int *checkBackground*, double *alpha*, double * *dist*, int *distNum*, double *maxAmp*, int *fast*, int *npsr*, int *numberGW*, longdouble *alpha*, longdouble ** *Func*)
- 12.98.2.5 void getThreshold (pulsar * *psr*, double * *freqVal*, int * *nFreq*, double * *threshold*, long * *idum*, int *fast*, int *npsr*, longdouble ** *Func*)
- 12.98.2.6 void GramSchmidt (longdouble *x*[], longdouble *y*[], longdouble *err*[], int *ObsAmt*, int *Npoly*, longdouble *CoeffArray*[], int *wtyn*)
- 12.98.2.7 void GramSchmidt (longdouble * *x*, longdouble * *y*, longdouble * *err*, int *ObsAmt*, int *Npoly*, longdouble * *CoeffArray*, int *wtyn*)
- 12.98.2.8 int graphicalInterface (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)
- 12.98.2.9 void help ()
- 12.98.2.10 void setupPulsar (longdouble *ra_p*, longdouble *dec_p*, longdouble * *kp*)
- 12.98.2.11 void shuffle (longdouble * *R*, double * *toaE*, longdouble * *R2*, double * *toaE2*, int *N*, long * *idum*)
- 12.98.2.12 void sortit (int *n*, double *array*[], double *rasort*[])
- 12.98.2.13 void writeCommands (int *argc*, char * *argv*[])

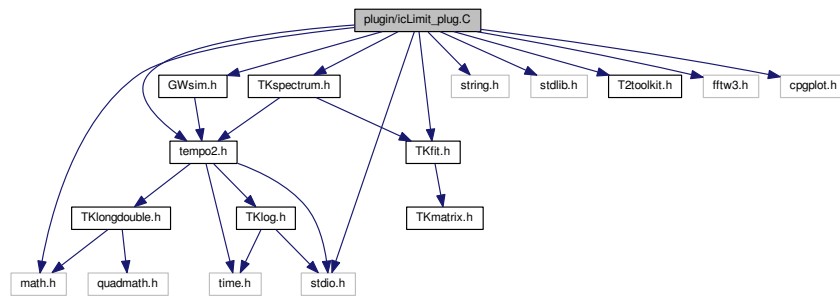
12.98.3 Variable Documentation

- 12.98.3.1 const char* plugVersionCheck = TEMPO2_h_VER
- 12.98.3.2 double storeVal[MAX_ITERATION][MAX_FREQ]

12.99 plugin/icLimit_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "TKspectrum.h"
#include "T2toolkit.h"
#include "GWsim.h"
#include "fftw3.h"
#include "TKfit.h"
#include <cpgplot.h>
```

Include dependency graph for icLimit_plug.C:



Functions

- double [getSpectra](#) ([pulsar](#) *psr, int npsr, char *[covarFuncFile](#), double **specX, double **specY, int *nSpec)
- double [getStatPS](#) ([pulsar](#) *psr, int npsr, double gwAmp, double gwAlpha, int it, char *[covarFuncFile](#), double noise, int plot, double *specX, double *specY, int *nSpec)
- void [calculateWeighting](#) (double *avSpecY, double *specX, int nSpec, double noiseLevel, double *weighting, double gwAmp, double gwAlpha)
- double [calculateStatistic](#) (double **specY, double **weighting, int *nSpec, int npsr)
- longdouble [getTspan](#) ([pulsar](#) *psr, int npsr)
- void [formCholeskyMatrixPlugin](#) (double *c, double *resx, double *resy, double *rese, int np, double **uinv)
- void [calculateGWCholesky](#) (double modelAlpha, double modelFc, double fitVar, double *covFunc, double dspan)
- void [createGWcovarianceFunction](#) (char *file, double gwAmp, double gwAlpha, [pulsar](#) *psr, int npsr, double *gwVar)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.99.1 Function Documentation

12.99.1.1 void [calculateGWCholesky](#) (double *modelAlpha*, double *modelFc*, double *fitVar*, double * *covFunc*, double *dspan*)

12.99.1.2 double [calculateStatistic](#) (double ** *specY*, double ** *weighting*, int * *nSpec*, int *npsr*)

12.99.1.3 void [calculateWeighting](#) (double * *avSpecY*, double * *specX*, int *nSpec*, double *noiseLevel*, double * *weighting*, double *gwAmp*, double *gwAlpha*)

12.99.1.4 void [createGWcovarianceFunction](#) (char * *file*, double *gwAmp*, double *gwAlpha*, [pulsar](#) * *psr*, int *npsr*, double * *gwVar*)

12.99.1.5 void [formCholeskyMatrixPlugin](#) (double * *c*, double * *resx*, double * *resy*, double * *rese*, int *np*, double ** *uinv*)

12.99.1.6 double [getSpectra](#) ([pulsar](#) * *psr*, int *npsr*, char * *covarFuncFile*, double ** *specX*, double ** *specY*, int * *nSpec*)

12.99.1.7 `double getStatPS (pulsar * psr, int npsr, double gwAmp, double gwAlpha, int it, char * covarFuncFile, double noise, int plot, double * specX, double * specY, int * nSpec)`

12.99.1.8 `longdouble getTspan (pulsar * psr, int npsr)`

12.99.1.9 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.99.1.10 `void help ()`

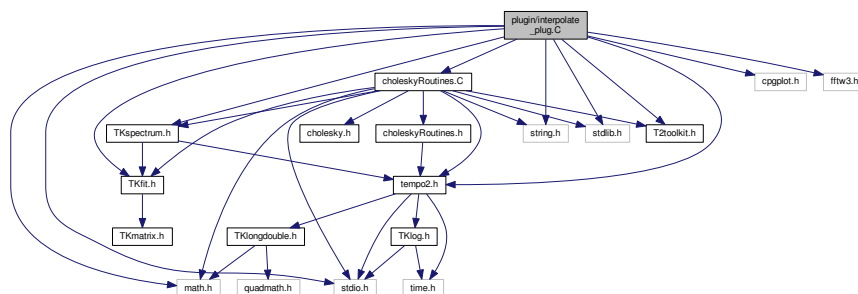
12.99.2 Variable Documentation

12.99.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.100 plugin/interpolate_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cgplot.h>
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "TKfit.h"
#include "fftw3.h"
#include "choleskyRoutines.C"
```

Include dependency graph for interpolate_plug.C:



Classes

- struct [sample](#)

Macros

- #define [MAX_SAMPLES](#) 25000
- #define [NRANSI](#)
- #define [TINY](#) 1.0e-20;
- #define [NR_END](#) 1
- #define [FREE_ARG](#) char*

Typedefs

- typedef struct [sample](#) [sample](#)

Functions

- void [plotResiduals](#) ([pulsar](#) *psr, [sample](#) *samples, int nSample, int drawFig)
- void [plotModel](#) ([pulsar](#) *psr, double startSample, double endSample, double spacingSample, [sample](#) *samples, int nSamples, int actualSamples, int drawFig)
- void [getPowerSpectra](#) ([pulsar](#) *psr, double modelA, double modelFc, double modelAlpha, double startSample, double endSample, double *covFunc, int *nCovFunc, [sample](#) *samples, int nSampleTimes, int gw, int drawFig)
- void [sortSamples](#) ([sample](#) *s, int n)
- void [choldc](#) (double **a, int n, double *p)
- void [lubksb](#) (double **a, int n, int *indx, double b[])
- void [ludcmp](#) (double **a, int n, int *indx, double *d)
- void [matrixMult](#) (double **m1, double **m2, int n)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)
- void [free_vector](#) (double *v, long nl, long nh)
- void [nrerror](#) (const char *error_text)
- double * [vector](#) (long nl, long nh)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.100.1 Macro Definition Documentation

12.100.1.1 `#define FREE_ARG char*`

12.100.1.2 `#define MAX_SAMPLES 25000`

12.100.1.3 `#define NR_END 1`

12.100.1.4 `#define NRANSI`

12.100.1.5 `#define TINY 1.0e-20;`

12.100.2 Typedef Documentation

12.100.2.1 typedef struct [sample](#) [sample](#)

12.100.3 Function Documentation

12.100.3.1 void [choldc](#) (double ** *a*, int *n*, double * *p*)

12.100.3.2 void [free_vector](#) (double * *v*, long *nl*, long *nh*)

12.100.3.3 void [getPowerSpectra](#) ([pulsar](#) * *psr*, double *modelA*, double *modelFc*, double *modelAlpha*, double *startSample*, double *endSample*, double * *covFunc*, int * *nCovFunc*, [sample](#) * *samples*, int *nSampleTimes*, int *gw*, int *drawFig*)

12.100.3.4 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.100.3.5 `void help ()`

12.100.3.6 `void lubksb (double ** a, int n, int * indx, double b[])`

12.100.3.7 `void ludcmp (double ** a, int n, int * indx, double * d)`

12.100.3.8 `void matrixMult (double ** m1, double ** m2, int n)`

12.100.3.9 `void nerror (const char * error_text)`

12.100.3.10 `void plotModel (pulsar * psr, double startSample, double endSample, double spacingSample, sample * samples, int nSamples, int actualSamples, int drawFig)`

12.100.3.11 `void plotResiduals (pulsar * psr, sample * samples, int nSample, int drawFig)`

12.100.3.12 `void sortSamples (sample * s, int n)`

12.100.3.13 `double* vector (long nl, long nh)`

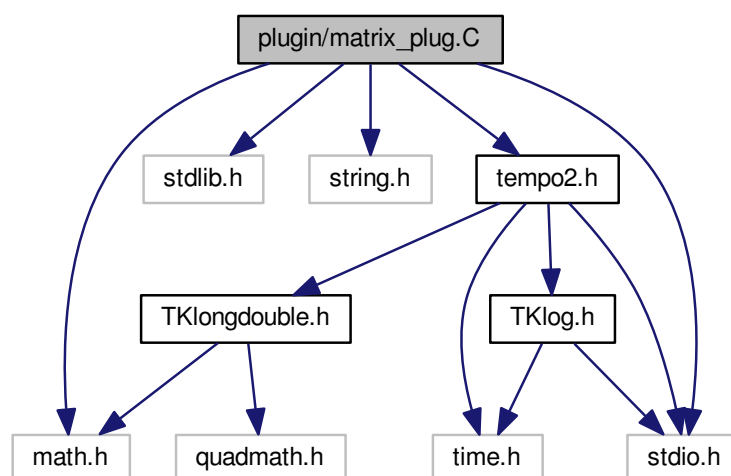
12.100.4 Variable Documentation

12.100.4.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.101 plugin/matrix_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for matrix_plug.C:



Functions

- void [getLabel](#) ([pulsar](#) *psr, char *lab, int i)
- int [tempoOutput](#) (int argc, char *argv[], [pulsar](#) *psr, int npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.101.1 Function Documentation

12.101.1.1 void [getLabel](#) ([pulsar](#) * *psr*, char * *lab*, int *i*)

12.101.1.2 int [tempoOutput](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int *npsr*)

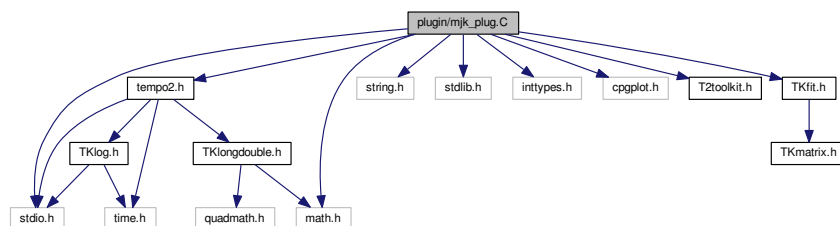
12.101.2 Variable Documentation

12.101.2.1 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.102 plugin/mjk_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <inttypes.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
#include "T2toolkit.h"
#include "TKfit.h"
```

Include dependency graph for mjk_plug.C:



Macros

- `#define` [NIT](#) 4

Functions

- double [iterativeFit](#) ([pulsar](#) *psr, double pb, double x, int *npsr)
- void [_itt](#) ([pulsar](#) *psr, int *npsr)
- void [saveparams](#) ([pulsar](#) *from, [pulsar](#) *to)
- void [help](#) ()

- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

12.102.1 Macro Definition Documentation

12.102.1.1 `#define NIT 4`

12.102.2 Function Documentation

12.102.2.1 `void _itt (pulsar * psr, int * npsr)`

12.102.2.2 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.102.2.3 `void help ()`

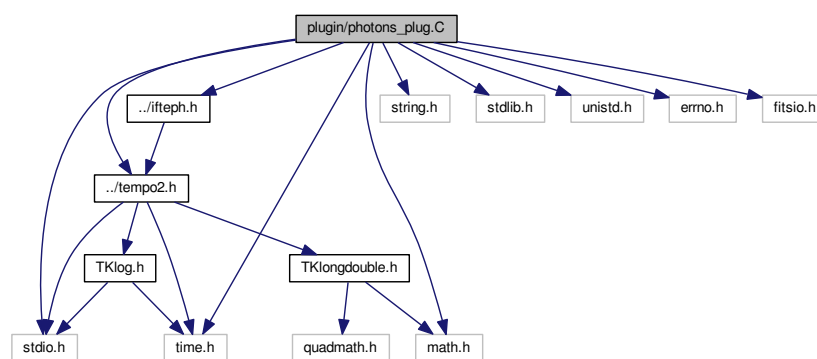
12.102.2.4 `double iterativeFit (pulsar * psr, double pb, double x, int * npsr)`

12.102.2.5 `void saveparams (pulsar * from, pulsar * to)`

12.103 plugin/photons_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <math.h>
#include "../tempo2.h"
#include "../ifteph.h"
#include <fitsio.h>
#include <time.h>
```

Include dependency graph for photons_plug.C:



Macros

- `#define IAU_TEPHO (-6.55e-5/86400) /* s */`
- `#define IAU_K 1.550519768e-8`
- `#define IAU_KINV 1.55051974395888e-8 /* 1 - 1/(1-IAU_K) */`

Functions

- [longdouble tcb2tdb](#) ([longdouble](#) *mjd*)
- [longdouble tdb2tcb](#) ([longdouble](#) *mjd*)
- [int find_event_hdu](#) ([fitsfile](#) **ft_in*)
- [longdouble get_mjdref](#) ([fitsfile](#) **ft_in*)
- [void check_barycentered](#) ([fitsfile](#) **ft_in*, [int](#) *event_hdu*)
- [int graphicalInterface](#) ([int](#) *argc*, [char](#) **argv*[], [pulsar](#) **psr*, [int](#) **npsr*)

12.103.1 Macro Definition Documentation

12.103.1.1 `#define IAU_K 1.550519768e-8`

12.103.1.2 `#define IAU_KINV 1.55051974395888e-8 /* 1 - 1/(1-IAU_K) */`

12.103.1.3 `#define IAU_TEPH0 (-6.55e-5/86400) /* s */`

12.103.2 Function Documentation

12.103.2.1 `void check_barycentered (fitsfile * ft_in, int event_hdu)`

12.103.2.2 `int find_event_hdu (fitsfile * ft_in)`

12.103.2.3 `longdouble get_mjdref (fitsfile * ft_in)`

12.103.2.4 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

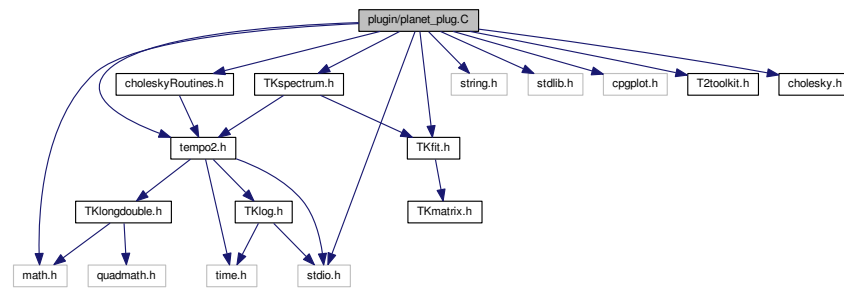
12.103.2.5 `longdouble tcb2tdb (longdouble mjd)`

12.103.2.6 `longdouble tdb2tcb (longdouble mjd)`

12.104 plugin/planet_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cpqplot.h>
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "TKfit.h"
#include "choleskyRoutines.h"
#include "cholesky.h"
```

Include dependency graph for planet_plug.C:



Functions

- void [plot6](#) (double *cholSpecX, double *cholSpecY, int nCholSpec, double *cholWspecX, double *cholWspecY, int nCholWspec, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, int makeps)
- void [doPlugin](#) ([pulsar](#) *psr, double idt, int ipw, double ifc, double iexp, int inpt, int makeps, double amp, char *dcf_file, int *npsr, char *argv[], int argc, char [parFile](#)[[MAX_FILELEN], char [timFile](#)[[MAX_FILELEN], int [nit](#))
- int [obtainTimingResiduals](#) ([pulsar](#) *psr, double *resx, double *resy, double *rese, int *ip)
- void [fitSineFunc](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival)
- void [plot1](#) (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *smoothModel, double *highFreqRes, double *hfNormCovar, int *hfNormCovarNpts, double hfZerolagNormCovar)
- void [removeMean](#) (double *resx, double *resy, int n)
- void [fileOutput3](#) (const char *fname, double *x, double *y, double *z, int n)
- void [fileOutput2](#) (const char *fname, double *x, double *y, int n)
- void [findSmoothCurve](#) (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *smoothModel, double expSmooth)
- void [getHighFreqRes](#) (double *resy, double *smoothModel, int nres, double *highFreqRes)
- void [getHighFreqCovar](#) (double *resx, double *rese, double *highFreqRes, int nres, double *hfNormCovar, int *hfNormCovarNpts, double *hfZerolagNormCovar)
- void [calculateDailyCovariance](#) (double *x, double *y, double *e, int n, double *cv, int *in, double *zl, int usew)
- int [calculateSpectra](#) (double *x, double *y, double *e, int n, int useErr, int preWhite, int specType, double *specX, double *specY)
- void [plot2](#) (double *origSpecX, double *origSpecY, int nOrigSpec, double *smoothSpecX0, double *smoothSpecY0, int nSmoothSpec0, double *smoothSpecX1, double *smoothSpecY1, int nSmoothSpec1, double *smoothSpecX2, double *smoothSpecY2, int nSmoothSpec2, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, int makeps)
- void [plot3](#) (double *preWhiteSpecX, double *preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, int closeit, float *minx, float *maxx, float wn)
- void [plot3a](#) (double *resx, double *resy, int nres, double *rawCovar, int *rawCovarNpts, double zerolagRawCovar, double *ampFit, double *chisqFit, int nGridFit, double bestAmp, double bestLag, double bestChisq, int makeps)
- void [plot4](#) (double *resx, double *resy, double *rese, int nres, double *cholWhiteY, double *whiteCovar, int *whiteCovarNpts, double zerolagWhiteCovar)
- void [plot5](#) (double *preWhiteSpecX, double *preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, double nmodelScale, double *cholSpecX, double *cholSpecY, int nCholSpec, double *cholWspecX, double *cholWspecY, int nCholWspec, int makeps, double wn, double pb)
- void [outputMatrix](#) (double **uinv, int nres)
- void [fitExponential](#) (double *resx, int nres, double *rawCovar, int *rawCovarNpts, double *ampFit, double *chisqFit, double *bestAmp, double *bestLag, double *bestChisq, int *nGridFit)

- void [calculateCholeskyCovarFunc](#) (double bestAmp, double bestLag, int nGridFit, double **uinv, double *resx, double *resy, double *rese, int nres, double *covarFunc)
- void [outputCovarianceFunction](#) (double *covFunc, int n, double errorScaleFactor, [pulsar](#) *psr)
- double [modelfcn](#) (double freq, double nmodelScale, double modelFc, double modelAlpha, double wn)
- int [T2fitSpectraRMS](#) (double *preWhiteSpecX, double *preWhiteSpecY, int nPreWhiteSpec, double *modelAlpha, double *modelFc, int *modelNfit, double *modelScale, double *fitVar, int aval, int ipw, double ifc, double iexp, int inpt, double amp, double *wn)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- char [pgdevice](#) [80]
- double [G_OMEGA](#)
- char [skipstep2](#) =0
- bool [writeFiles](#) =true
- int [skipprocess](#) =0
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.104.1 Function Documentation

- 12.104.1.1 void [calculateCholeskyCovarFunc](#) (double *bestAmp*, double *bestLag*, int *nGridFit*, double ** *uinv*, double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *covarFunc*)
- 12.104.1.2 void [calculateDailyCovariance](#) (double * *x*, double * *y*, double * *e*, int *n*, double * *cv*, int * *in*, double * *zl*, int *usew*)
- 12.104.1.3 int [calculateSpectra](#) (double * *x*, double * *y*, double * *e*, int *n*, int *useErr*, int *preWhite*, int *specType*, double * *specX*, double * *specY*)
- 12.104.1.4 void [doPlugin](#) ([pulsar](#) * *psr*, double *idt*, int *ipw*, double *ifc*, double *iexp*, int *inpt*, int *makeps*, double *amp*, char * *dcf_file*, int * *npsr*, char * *argv*[], int *argc*, char *parFile*[(MAX_FILELEN)], char *timFile*[(MAX_FILELEN)], int *nit*)
- 12.104.1.5 void [fileOutput2](#) (const char * *fname*, double * *x*, double * *y*, int *n*)
- 12.104.1.6 void [fileOutput3](#) (const char * *fname*, double * *x*, double * *y*, double * *z*, int *n*)
- 12.104.1.7 void [findSmoothCurve](#) (double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *cubicVal*, double * *smoothModel*, double *expSmooth*)
- 12.104.1.8 void [fitExponential](#) (double * *resx*, int *nres*, double * *rawCovar*, int * *rawCovarNpts*, double * *ampFit*, double * *chisqFit*, double * *bestAmp*, double * *bestLag*, double * *bestChisq*, int * *nGridFit*)
- 12.104.1.9 void [fitSineFunc](#) (double *x*, double * *v*, int *nfit*, [pulsar](#) * *psr*, int *ival*)
- 12.104.1.10 void [getHighFreqCovar](#) (double * *resx*, double * *rese*, double * *highFreqRes*, int *nres*, double * *hfNormCovar*, int * *hfNormCovarNpts*, double * *hfZeroLagNormCovar*)
- 12.104.1.11 void [getHighFreqRes](#) (double * *resy*, double * *smoothModel*, int *nres*, double * *highFreqRes*)
- 12.104.1.12 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)
- 12.104.1.13 void [help](#) ()

- 12.104.1.14 `double modelFcn (double freq, double nmodelScale, double modelFc, double modelAlpha, double wn)`
- 12.104.1.15 `int obtainTimingResiduals (pulsar * psr, double * resx, double * resy, double * rese, int * ip)`
- 12.104.1.16 `void outputCovarianceFunction (double * covFunc, int n, double errorScaleFactor, pulsar * psr)`
- 12.104.1.17 `void outputMatrix (double ** uin, int nres)`
- 12.104.1.18 `void plot1 (double * resx, double * resy, double * rese, int nres, double * cubicVal, double * smoothModel, double * highFreqRes, double * hfNormCovar, int * hfNormCovarNpts, double hfZeroLagNormCovar)`
- 12.104.1.19 `void plot2 (double * origSpecX, double * origSpecY, int nOrigSpec, double * smoothSpecX0, double * smoothSpecY0, int nSmoothSpec0, double * smoothSpecX1, double * smoothSpecY1, int nSmoothSpec1, double * smoothSpecX2, double * smoothSpecY2, int nSmoothSpec2, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, int makeps)`
- 12.104.1.20 `void plot3 (double * preWhiteSpecX, double * preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, int closeit, float * minx, float * maxx, float wn)`
- 12.104.1.21 `void plot3a (double * resx, double * resy, int nres, double * rawCovar, int * rawCovarNpts, double zerolagRawCovar, double * ampFit, double * chisqFit, int nGridFit, double bestAmp, double bestLag, double bestChisq, int makeps)`
- 12.104.1.22 `void plot4 (double * resx, double * resy, double * rese, int nres, double * cholWhiteY, double * whiteCovar, int * whiteCovarNpts, double zerolagWhiteCovar)`
- 12.104.1.23 `void plot5 (double * preWhiteSpecX, double * preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, double nmodelScale, double * cholSpecX, double * cholSpecY, int nCholSpec, double * cholWspecX, double * cholWspecY, int nCholWspec, int makeps, double wn, double pb)`
- 12.104.1.24 `void plot6 (double * cholSpecX, double * cholSpecY, int nCholSpec, double * cholWspecX, double * cholWspecY, int nCholWspec, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, int makeps)`
- 12.104.1.25 `void removeMean (double * resx, double * resy, int n)`
- 12.104.1.26 `int T2fitSpectraRMS (double * preWhiteSpecX, double * preWhiteSpecY, int nPreWhiteSpec, double * modelAlpha, double * modelFc, int * modelNfit, double * modelScale, double * fitVar, int aval, int ipw, double ifc, double iexp, int inpt, double amp, double * wn)`

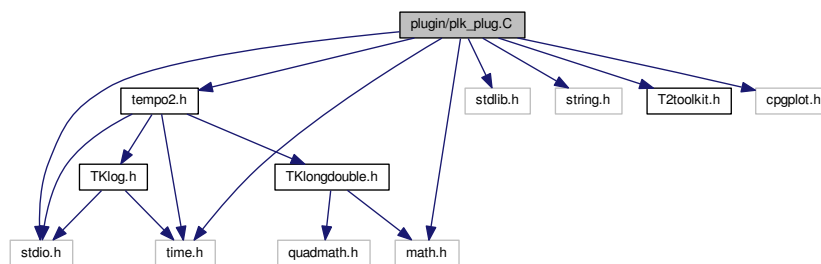
12.104.2 Variable Documentation

- 12.104.2.1 `double G_OMEGA`
- 12.104.2.2 `char pgdevice[80]`
- 12.104.2.3 `const char* plugVersionCheck = TEMPO2_h_VER`
- 12.104.2.4 `int skipprocess =0`
- 12.104.2.5 `char skipstep2 =0`
- 12.104.2.6 `bool writeFiles =true`

12.105 plugin/plk_plug.C File Reference

```
#include "tempo2.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "T2toolkit.h"
#include <time.h>
#include <cpgplot.h>
```

Include dependency graph for plk_plug.C:



Functions

- void [overPlotN](#) (int overN, float overX[], float overY[], float overYe[])
- void [doPlot](#) (pulsar *psr, int npsr, char *gr, double unitFlag, char parFile[][MAX_FILELEN], char timFile[][MAX_FILELEN], float lockx1, float lockx2, float locky1, float locky2, int xplot, int yplot, int publish, int argc, char *argv[], int menu, char *setupFile, int showChisq, int nohead, char *flagColour, char *bandsFile, int displayPP)
- int [setPlot](#) (float *x, int count, pulsar *psr, int iobs, double unitFlag, int plotPhase, int plot, int *userValChange, char *userCMD, char *userValStr, float *userX, longdouble centreEpoch, int log, char *flagStr)
- void [drawAxisSel](#) (float x, float y, const char *str, int sel1, int sel2)
- float [findMinY](#) (float *y, float *x, int count, float xmin, float xmax)
- float [findMaxY](#) (float *y, float *x, int count, float xmin, float xmax)
- float [findMean](#) (float *x, pulsar *psr, int i1, int i2)
- double [findMeanD](#) (float *x, pulsar *psr, int i1, int i2)
- void [callFit](#) (pulsar *psr, int npsr)
- void [reFit](#) (int fitFlag, int setZoomX1, int setZoomX2, float zoomX1, float zoomX2, longdouble origStart, longdouble origFinish, longdouble centreEpoch, pulsar *psr, int npsr, int plotX, char *dcmFile, char *covarFuncFile, int zoom)
- float [deletePoint](#) (pulsar *psr, float *x, float *y, int *id, int count, float mouseX, float mouseY)
- void [displayStatistics](#) (float *x, float *y, int count, float plotx1, float plotx2, float ploty1, float ploty2)
- int [idPoint](#) (pulsar *psr, float *x, float *y, int *id, int count, float mouseX, float mouseY)
- double [fortranMod](#) (double a, double p)
- void [sort](#) (float *x, float *y, float *yerr1, float *yerr2, float *freq, int *id, int count)
- void [changeParameters](#) (pulsar *psr)
- void [changeFitParameters](#) (pulsar *psr)
- void [averagePts](#) (float *x, float *y, int n, int width, float *meanX, float *meanY, int *nMean)
- void [overPlotShapiro](#) (pulsar *psr, float offset, longdouble centreEpoch)
- void [binResiduals](#) (pulsar *psr, int npsr, float *x, float *y, int count, int *id, int *overN, float overX[], float overY[], float overYe[], int xplot, int yplot, float errBar[], double unitFlag, int plotPhase, double centreEpoch)
- void [drawMenu](#) (pulsar *psr, float plotx1, float plotx2, float ploty1, float ploty2, int menu, int paramOffset)

- void [drawMenu3](#) ([pulsar](#) *psr, float plotx1, float plotx2, float ploty1, float ploty2, int menu, int xplot, int yplot)
- void [slaClyd](#) (int iy, int im, int id, int *ny, int *nd, int *jstat)
- void [slaCalyd](#) (int iy, int im, int id, int *ny, int *nd, int *j)
- void [drawMenu3_2](#) ([pulsar](#) *psr, float plotx1, float plotx2, float ploty1, float ploty2, int menu, int xplot, int yplot, int jumpOffset, int iFlagColour, int nFlags)
- void [checkMenu](#) ([pulsar](#) *psr, float mx, float my, int button, int fitFlag, int setZoomX1, int setZoomX2, float zoomX1, float zoomX2, [longdouble](#) origStart, [longdouble](#) origFinish, [longdouble](#) centreEpoch, int menu, int plotx, char [parFile](#)[][MAX_FILELEN], char [timFile](#)[][MAX_FILELEN], int argc, char *argv[], int *xplot, int *yplot, int *graphics, char highlightID[100][100], char highlightVal[100][100], int *highlightNum, float aspect, int fontType, int lineWidth, char *bkgrdColour, char *lineColour, int *jumpOffset, int zoom, int *paramOffset)
- void [checkMenu3](#) ([pulsar](#) *psr, float mx, float my, int button, int fitFlag, int setZoomX1, int setZoomX2, float zoomX1, float zoomX2, [longdouble](#) origStart, [longdouble](#) origFinish, [longdouble](#) centreEpoch, int menu, int plotx, char [parFile](#)[][MAX_FILELEN], char [timFile](#)[][MAX_FILELEN], int argc, char *argv[], int *xplot, int *yplot, int *graphics, char highlightID[100][100], char highlightVal[100][100], int *highlightNum, float aspect, int fontType, int lineWidth, char *bkgrdColour, char *lineColour, int *jumpOffset)
- void [setLabel](#) (char *ystr, int yplot, int plotPhase, double unitFlag, [longdouble](#) centreEpoch, char *userValStr, char *flagStr)
- void [drawOption](#) (float x, float y, const char *str, int fit)
- void [swapFit](#) ([pulsar](#) *psr, int par, int k, int button)
- void [newTim](#) ([pulsar](#) *psr)
- void [plotFITWAVES_spec](#) ()
- void [viewModels](#) ([pulsar](#) *psr, float x1, float x2, [longdouble](#) centreEpoch, int [removeMean](#), double [mean](#), int count, int *id, int fitFlag, float *x, float *y)
- double [lmst2](#) (double mjd, double olong, double *tsid, double *tsid_der)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- char [dcmFile](#) [MAX_FILELEN]
- char [covarFuncFile](#) [MAX_FILELEN]
- bool [cholmode](#) =false
- double [FITWAVES_omega](#)
- int [FITWAVES_n](#)
- int [FITWAVES_harmonicStep](#)
- double [FITWAVES_par](#) [1000]
- char [flagStore](#) [100][100]
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.105.1 Function Documentation

- 12.105.1.1 void [averagePts](#) (float * *x*, float * *y*, int *n*, int *width*, float * *meanX*, float * *meanY*, int * *nMean*)
- 12.105.1.2 void [binResiduals](#) ([pulsar](#) * *psr*, int *npsr*, float * *x*, float * *y*, int *count*, int * *id*, int * *overN*, float *overX*[], float *overY*[], float *overYe*[], int *xplot*, int *yplot*, float *errBar*[], double *unitFlag*, int *plotPhase*, double *centreEpoch*)
- 12.105.1.3 void [callFit](#) ([pulsar](#) * *psr*, int *npsr*)
- 12.105.1.4 void [changeFitParameters](#) ([pulsar](#) * *psr*)
- 12.105.1.5 void [changeParameters](#) ([pulsar](#) * *psr*)

- 12.105.1.6 void checkMenu (pulsar * *psr*, float *mx*, float *my*, int *button*, int *fitFlag*, int *setZoomX1*, int *setZoomX2*, float *zoomX1*, float *zoomX2*, longdouble *origStart*, longdouble *origFinish*, longdouble *centreEpoch*, int *menu*, int *plotx*, char *parFile*[[MAX_FILELEN], char *timFile*[[MAX_FILELEN], int *argc*, char * *argv*[], int * *xplot*, int * *yplot*, int * *graphics*, char *highlightID*[100][100], char *highlightVal*[100][100], int * *highlightNum*, float *aspect*, int *fontType*, int *lineWidth*, char * *bkgrdColour*, char * *lineColour*, int * *jumpOffset*, int *zoom*, int * *paramOffset*)
- 12.105.1.7 void checkMenu3 (pulsar * *psr*, float *mx*, float *my*, int *button*, int *fitFlag*, int *setZoomX1*, int *setZoomX2*, float *zoomX1*, float *zoomX2*, longdouble *origStart*, longdouble *origFinish*, longdouble *centreEpoch*, int *menu*, int *plotx*, char *parFile*[[MAX_FILELEN], char *timFile*[[MAX_FILELEN], int *argc*, char * *argv*[], int * *xplot*, int * *yplot*, int * *graphics*, char *highlightID*[100][100], char *highlightVal*[100][100], int * *highlightNum*, float *aspect*, int *fontType*, int *lineWidth*, char * *bkgrdColour*, char * *lineColour*, int * *jumpOffset*)
- 12.105.1.8 float deletePoint (pulsar * *psr*, float * *x*, float * *y*, int * *id*, int *count*, float *mouseX*, float *mouseY*)
- 12.105.1.9 void displayStatistics (float * *x*, float * *y*, int *count*, float *plotx1*, float *plotx2*, float *ploty1*, float *ploty2*)
- 12.105.1.10 void doPlot (pulsar * *psr*, int *npsr*, char * *gr*, double *unitFlag*, char *parFile*[[MAX_FILELEN], char *timFile*[[MAX_FILELEN], float *lockx1*, float *lockx2*, float *locky1*, float *locky2*, int *xplot*, int *yplot*, int *publish*, int *argc*, char * *argv*[], int *menu*, char * *setupFile*, int *showChisq*, int *nohead*, char * *flagColour*, char * *bandsFile*, int *displayPP*)
- 12.105.1.11 void drawAxisSel (float *x*, float *y*, const char * *str*, int *sel1*, int *sel2*)
- 12.105.1.12 void drawMenu (pulsar * *psr*, float *plotx1*, float *plotx2*, float *ploty1*, float *ploty2*, int *menu*, int *paramOffset*)
- 12.105.1.13 void drawMenu3 (pulsar * *psr*, float *plotx1*, float *plotx2*, float *ploty1*, float *ploty2*, int *menu*, int *xplot*, int *yplot*)
- 12.105.1.14 void drawMenu3_2 (pulsar * *psr*, float *plotx1*, float *plotx2*, float *ploty1*, float *ploty2*, int *menu*, int *xplot*, int *yplot*, int *jumpOffset*, int *iFlagColour*, int *nFlags*)
- 12.105.1.15 void drawOption (float *x*, float *y*, const char * *str*, int *fit*)
- 12.105.1.16 float findMaxY (float * *y*, float * *x*, int *count*, float *xmin*, float *xmax*)
- 12.105.1.17 float findMean (float * *x*, pulsar * *psr*, int *i1*, int *i2*)
- 12.105.1.18 double findMeanD (float * *x*, pulsar * *psr*, int *i1*, int *i2*)
- 12.105.1.19 float findMinY (float * *y*, float * *x*, int *count*, float *xmin*, float *xmax*)
- 12.105.1.20 double fortranMod (double *a*, double *p*)
- 12.105.1.21 int graphicalInterface (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)
- 12.105.1.22 void help ()
- 12.105.1.23 int idPoint (pulsar * *psr*, float * *x*, float * *y*, int * *id*, int *count*, float *mouseX*, float *mouseY*)
- 12.105.1.24 double lmst2 (double *mjd*, double *olong*, double * *tsid*, double * *tsid_der*)
- 12.105.1.25 void newTim (pulsar * *psr*)
- 12.105.1.26 void overPlotN (int *overN*, float *overX*[], float *overY*[], float *overYe*[])
- 12.105.1.27 void overPlotShapiro (pulsar * *psr*, float *offset*, longdouble *centreEpoch*)

- 12.105.1.28 `void plotFITWAVES_spec ()`
- 12.105.1.29 `void reFit (int fitFlag, int setZoomX1, int setZoomX2, float zoomX1, float zoomX2, longdouble origStart, longdouble origFinish, longdouble centreEpoch, pulsar * psr, int npsr, int plotX, char * dcmFile, char * covarFuncFile, int zoom)`
- 12.105.1.30 `void setLabel (char * ystr, int yplot, int plotPhase, double unitFlag, longdouble centreEpoch, char * userValStr, char * flagStr)`
- 12.105.1.31 `int setPlot (float * x, int count, pulsar * psr, int iobs, double unitFlag, int plotPhase, int plot, int * userValChange, char * userCMD, char * userValStr, float * userX, longdouble centreEpoch, int log, char * flagStr)`
- 12.105.1.32 `void slaCalyd (int iy, int im, int id, int * ny, int * nd, int * j)`
- 12.105.1.33 `void slaClyd (int iy, int im, int id, int * ny, int * nd, int * jstat)`
- 12.105.1.34 `void sort (float * x, float * y, float * yerr1, float * yerr2, float * freq, int * id, int count)`
- 12.105.1.35 `void swapFit (pulsar * psr, int par, int k, int button)`
- 12.105.1.36 `void viewModels (pulsar * psr, float x1, float x2, longdouble centreEpoch, int removeMean, double mean, int count, int * id, int fitFlag, float * x, float * y)`

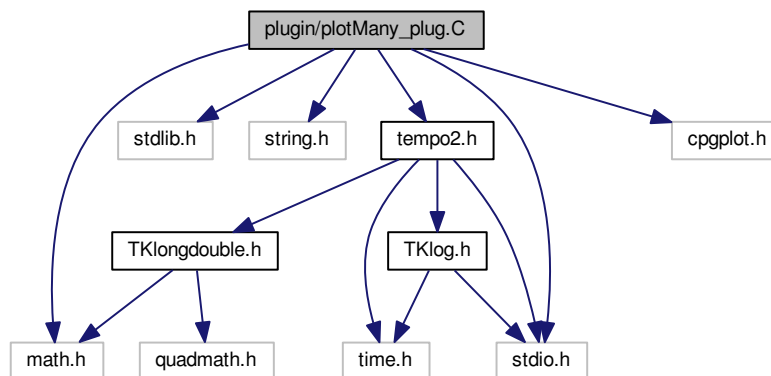
12.105.2 Variable Documentation

- 12.105.2.1 `bool cholmode = false`
- 12.105.2.2 `char covarFuncFile[MAX_FILELEN]`
- 12.105.2.3 `char dcmFile[MAX_FILELEN]`
- 12.105.2.4 `int FITWAVES_harmonicStep`
- 12.105.2.5 `int FITWAVES_n`
- 12.105.2.6 `double FITWAVES_omega`
- 12.105.2.7 `double FITWAVES_par[1000]`
- 12.105.2.8 `char flagStore[100][100]`
- 12.105.2.9 `const char* plugVersionCheck = TEMPO2_h_VER`

12.106 plugin/plotMany_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
```

Include dependency graph for plotMany_plug.C:



Functions

- void [doPlot](#) ([pulsar](#) *psr, int npsr, float *scale, int nScale, char *grDev, int plotUs, float fontsize, float centreMJD, int ptStyle, float ptSize, int error, float miny, float maxy, float minx, float maxx, int nOverlay, float labelsize, float fracX)
- float [findMin](#) (float *x, [pulsar](#) *psr, int p, int i1, int i2)
- float [findMax](#) (float *x, [pulsar](#) *psr, int p, int i1, int i2)
- float [findMean](#) (float *x, [pulsar](#) *psr, int p, int i1, int i2)
- void [callFit](#) ([pulsar](#) *psr, int npsr)
- float [findMinVal](#) (float *a, int n)
- float [findMaxVal](#) (float *a, int n)
- double [fortranMod](#) (double a, double p)
- double [lmst2](#) (double mjd, double olong, double *tsid, double *tsid_der)
- void [slaClyd](#) (int iy, int im, int id, int *ny, int *nd, int *jstat)
- void [slaCalyd](#) (int iy, int im, int id, int *ny, int *nd, int *j)
- float [calcYr](#) (float mjd)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- char [covarFuncFile2](#) [MAX_FILELEN]

12.106.1 Function Documentation

12.106.1.1 float [calcYr](#) (float *mjd*)

12.106.1.2 void [callFit](#) ([pulsar](#) * *psr*, int *npsr*)

12.106.1.3 void [doPlot](#) ([pulsar](#) * *psr*, int *npsr*, float * *scale*, int *nScale*, char * *grDev*, int *plotUs*, float *fontsize*, float *centreMJD*, int *ptStyle*, float *ptSize*, int *error*, float *miny*, float *maxy*, float *minx*, float *maxx*, int *nOverlay*, float *labelsize*, float *fracX*)

12.106.1.4 float [findMax](#) (float * *x*, [pulsar](#) * *psr*, int *p*, int *i1*, int *i2*)

- 12.106.1.5 `float findMaxVal (float * a, int n)`
- 12.106.1.6 `float findMean (float * x, pulsar * psr, int p, int i1, int i2)`
- 12.106.1.7 `float findMin (float * x, pulsar * psr, int p, int i1, int i2)`
- 12.106.1.8 `float findMinVal (float * a, int n)`
- 12.106.1.9 `double fortranMod (double a, double p)`
- 12.106.1.10 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`
- 12.106.1.11 `void help ()`
- 12.106.1.12 `double lmst2 (double mjd, double along, double * tsid, double * tsid_der)`
- 12.106.1.13 `void slaCalyd (int iy, int im, int id, int * ny, int * nd, int * j)`
- 12.106.1.14 `void slaClyd (int iy, int im, int id, int * ny, int * nd, int * jstat)`

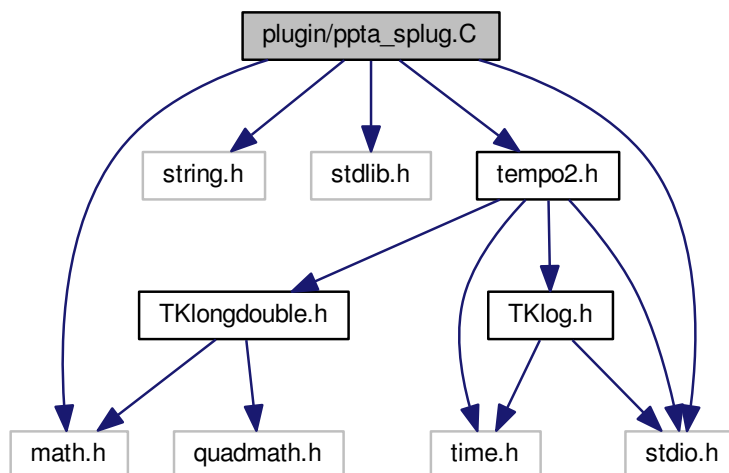
12.106.2 Variable Documentation

- 12.106.2.1 `char covarFuncFile2[MAX_FILELEN]`

12.107 plugin/ppta_splug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for ppta_splug.C:



Functions

- int [selectInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

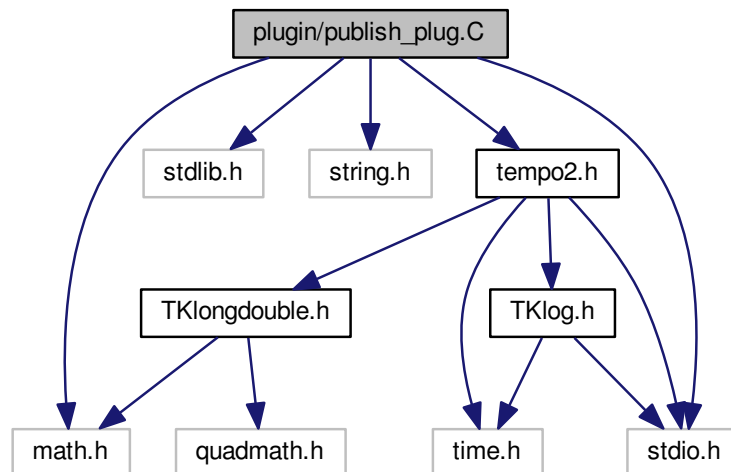
12.107.1 Function Documentation

12.107.1.1 int selectInterface (int argc, char * argv[], [pulsar](#) * psr, int * npsr)

12.108 plugin/publish_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for publish_plug.C:



Functions

- void [dispParameter](#) (int i, int k, [pulsar](#) *psr, FILE *fout, int err, double efac, int useCompare, char *compare↵
File, int *useDagger)
- int [nint_derived](#) (double x)
- int [rnd8](#) (double rval, double rerr, int ifac, char *cval, int *lv, char *cerr, int *le, char *msg)
- void [parseMinus](#) (char *str)
- void [parseExp](#) (char *str)
- double [fixRA](#) (char *tstr, double err, char *valStr)
- double [fixDec](#) (char *tstr, double err, char *valStr)
- int [tempoOutput](#) (int argc, char *argv[], [pulsar](#) *psr, int npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.108.1 Function Documentation

12.108.1.1 `void dispParameter (int i, int k, pulsar * psr, FILE * fout, int err, double efac, int useCompare, char * compareFile, int * useDagger)`

12.108.1.2 `double fixDec (char * tstr, double err, char * valStr)`

12.108.1.3 `double fixRA (char * tstr, double err, char * valStr)`

12.108.1.4 `int nint_derived (double x)`

12.108.1.5 `void parseExp (char * str)`

12.108.1.6 `void parseMinus (char * str)`

12.108.1.7 `int rnd8 (double rval, double rerr, int ifac, char * cval, int * lv, char * cerr, int * le, char * msg)`

12.108.1.8 `int tempoOutput (int argc, char * argv[], pulsar * psr, int npsr)`

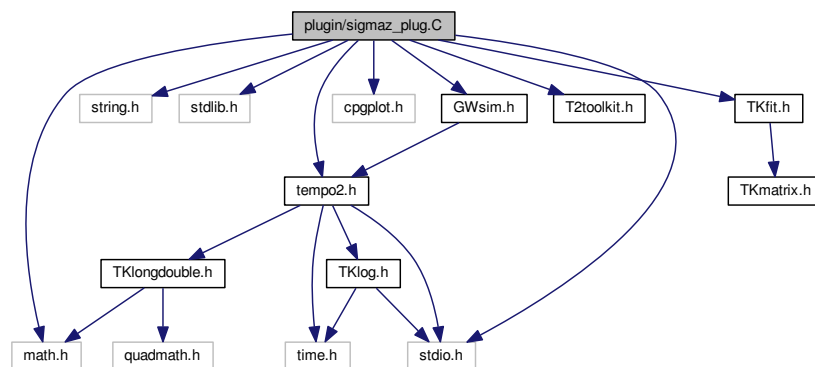
12.108.2 Variable Documentation

12.108.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.109 plugin/sigmaz_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <cpgplot.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include "GWsim.h"
```

Include dependency graph for sigmaz_plug.C:



Classes

- struct [XY](#)

Macros

- #define `MAX_GWS` 10000

Typedefs

- typedef struct `XY XY`

Functions

- void `readin` (`pulsar` psr)
- void `getprtj` (int n)
- void `indexx8` (int n, double *arrin, int *indx)
- void `getweights` (int n, double *wt)
- void `fit4` (int *nfit, double *p4, double *cov4, int ndostats, double *chidf, double *avewt)
- void `mat20` (double sam[21][21], double a[21][21], int n, double *determ, int *nbad)
- void `simWhiteFunc` (`pulsar` *psr, long *idum, char `parFile`[`MAX_PSR_VAL`][`MAX_FILELEN`], char `timFile`[`MAX_PSR_VAL`][`MAX_FILELEN`], int weights, double mintau)
- void `calcSigmaz` (`pulsar` psr, int weights, double *tau, double *szbias, double *e1, double *e2, int *nval, double mintau)
- void `doplot` (`pulsar` *psr, int npsr, char *grDev, float mint, float maxt, float minsz, float maxsz, int style, int `average`, double tau[`MAX_PSR_VAL`][100], double szbias[`MAX_PSR_VAL`][100], double e1[`MAX_PSR_VAL`][100], double e2[`MAX_PSR_VAL`][100], int nval[`MAX_PSR_VAL`], int nWhite, float *white, char *cline, int slopes, int bound)
- void `sortTimes` (`pulsar` psr, int *nobs, double *times, double *resid, double *error)
- void `fitv` (double x, double afunc[], int ma, `pulsar` *psr, int ipos)
- void `plotOmega_g` (double omega, float *px, float *py)
- void `plotA_g` (double a, double `alpha`, float *px, float *py)
- void `shufflePts` (`longdouble` *R, double *toaE, `longdouble` *R2, double *toaE2, int N, long *idum)
- void `convert_gravWaveBackground_noFit` (`pulsar` *psr, int npsr, double convertGW, long *idum, int same↵ Background)
- void `convert_gravWaveBackground_fit` (`pulsar` *psr, int npsr, double convertGW, long *idum, int same↵ Background, char `parFile`[`MAX_PSR_VAL`][`MAX_FILELEN`], char `timFile`[`MAX_PSR_VAL`][`MAX_FILELEN`])
- void `calcSpline` (float *px, float *py, int count)
- float `SplineBlend` (int k, int t, int *u, float v)
- void `calculateGWlwm` (`pulsar` *psr, long *idum, double obs, char `parFile`[`MAX_PSR_VAL`][`MAX_FILELEN`], char `timFile`[`MAX_PSR_VAL`][`MAX_FILELEN`], int weights, double mintau, `longdouble` gwamp, float mint, float maxt, float minsz, float maxsz, double *szbias, double *e1obs, double *e2obs, int nit, int ngw, `longdouble` lowAmp, `longdouble` hiAmp)
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], `pulsar` *psr, int *npsr)
- void `SplinePoint` (int *u, int n, int t, float v, `XY` *control, `XY` *output)
- void `SplineKnots` (int *u, int n, int t)
- void `SplineCurve` (`XY` *inp, int n, int *knots, int t, `XY` *outp, int res)

Variables

- int `npt`
- int `nusewt`
- int `nxunits`
- int `ntunits`
- int `nformat`
- int `nwriteres`
- int `nbintype`

- int [npt1last](#)
- int [npt2last](#)
- int [ncubic](#)
- int [ncubics](#)
- int [ntau](#)
- int [linfile](#)
- int [indx](#) [90000]
- int [ndim](#)
- double [data](#) [90000]
- double [utjd](#) [90000]
- double [taumin](#)
- double [sigmai](#) [90000]
- double [permax](#)
- double [root2](#)
- double [utjd1](#)
- double [utjd2](#)
- double [tmin](#)
- double [tmax](#)
- double [xmin](#)
- double [xmax](#)
- double [utjdlast](#)
- double [tausec](#)
- double [taumax](#)
- double [tauday](#)
- double [prtl](#) [5]
- double [utmean](#)
- double [secyear](#)
- double [taulog](#)
- double [addvar](#)
- double [tauyear](#)
- double [tauensure](#)
- double [tdiffmin](#)
- double [utfirst](#)
- double [utlast](#)
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.109.1 Macro Definition Documentation

12.109.1.1 `#define MAX_GWS 10000`

12.109.2 Typedef Documentation

12.109.2.1 `typedef struct XY XY`

12.109.3 Function Documentation

12.109.3.1 `void calcSigmaz (pulsar psr, int weights, double * tau, double * szbias, double * e1, double * e2, int * nval, double mintau)`

12.109.3.2 `void calcSpline (float * px, float * py, int count)`

12.109.3.3 `void calculateGWlim (pulsar * psr, long * idum, double obs, char parFile[MAX_PSR_VAL][MAX_FILELEN], char timFile[MAX_PSR_VAL][MAX_FILELEN], int weights, double mintau, longdouble gwamp, float mint, float maxt, float minsz, float maxsz, double * szbias, double * e1obs, double * e2obs, int nit, int ngw, longdouble lowAmp, longdouble hiAmp)`

- 12.109.3.4 void convert_gravWaveBackground_fit (pulsar * *psr*, int *npsr*, double *convertGW*, long * *idum*, int *sameBackground*, char *parFile*[MAX_PSR_VAL][MAX_FILELEN], char *timFile*[MAX_PSR_VAL][MAX_FILELEN])
- 12.109.3.5 void convert_gravWaveBackground_noFit (pulsar * *psr*, int *npsr*, double *convertGW*, long * *idum*, int *sameBackground*)
- 12.109.3.6 void doplot (pulsar * *psr*, int *npsr*, char * *grDev*, float *mint*, float *maxt*, float *minsz*, float *maxsz*, int *style*, int *average*, double *tau*[MAX_PSR_VAL][100], double *szbias*[MAX_PSR_VAL][100], double *e1*[MAX_PSR_VAL][100], double *e2*[MAX_PSR_VAL][100], int *nval*[MAX_PSR_VAL], int *nWhite*, float * *white*, char * *cline*, int *slopes*, int *bound*)
- 12.109.3.7 void fit4 (int * *nfit*, double * *p4*, double * *cov4*, int *ndostats*, double * *chidf*, double * *avewt*)
- 12.109.3.8 void fitv (double *x*, double *afunc*[], int *ma*, pulsar * *psr*, int *ipos*)
- 12.109.3.9 void getprtj (int *n*)
- 12.109.3.10 void getweights (int *n*, double * *wt*)
- 12.109.3.11 int graphicalInterface (int *argc*, char * *argv*[], pulsar * *psr*, int * *npsr*)
- 12.109.3.12 void help ()
- 12.109.3.13 void indexx8 (int *n*, double * *arrin*, int * *indx*)
- 12.109.3.14 void mat20 (double *sam*[21][21], double *a*[21][21], int *n*, double * *determ*, int * *nbad*)
- 12.109.3.15 void plotA_g (double *a*, double *alpha*, float * *px*, float * *py*)
- 12.109.3.16 void plotOmega_g (double *omega*, float * *px*, float * *py*)
- 12.109.3.17 void readin (pulsar *psr*)
- 12.109.3.18 void shufflePts (longdouble * *R*, double * *toaE*, longdouble * *R2*, double * *toaE2*, int *N*, long * *idum*)
- 12.109.3.19 void simWhiteFunc (pulsar * *psr*, long * *idum*, char *parFile*[MAX_PSR_VAL][MAX_FILELEN], char *timFile*[MAX_PSR_VAL][MAX_FILELEN], int *weights*, double *mintau*)
- 12.109.3.20 void sortTimes (pulsar *psr*, int * *nobs*, double * *times*, double * *resid*, double * *error*)
- 12.109.3.21 float SplineBlend (int *k*, int *t*, int * *u*, float *v*)
- 12.109.3.22 void SplineCurve (XY * *inp*, int *n*, int * *knots*, int *t*, XY * *outp*, int *res*)
- 12.109.3.23 void SplineKnots (int * *u*, int *n*, int *t*)
- 12.109.3.24 void SplinePoint (int * *u*, int *n*, int *t*, float *v*, XY * *control*, XY * *output*)
- 12.109.4 Variable Documentation
- 12.109.4.1 double addvar
- 12.109.4.2 double data[90000]
- 12.109.4.3 int indx[90000]

12.109.4.4 int linfile

12.109.4.5 int nbintype

12.109.4.6 int ncubic

12.109.4.7 int ncubics

12.109.4.8 int ndim

12.109.4.9 int nformat

12.109.4.10 int npt

12.109.4.11 int npt1last

12.109.4.12 int npt2last

12.109.4.13 int ntau

12.109.4.14 int ntunits

12.109.4.15 int nusewt

12.109.4.16 int nwriteres

12.109.4.17 int nxunits

12.109.4.18 double permax

12.109.4.19 const char* plugVersionCheck = TEMPO2_h_VER

12.109.4.20 double prtl[5]

12.109.4.21 double root2

12.109.4.22 double secyear

12.109.4.23 double sigma[90000]

12.109.4.24 double tauday

12.109.4.25 double tauensure

12.109.4.26 double tauog

12.109.4.27 double taumax

12.109.4.28 double taumin

12.109.4.29 double tausec

12.109.4.30 double tauyear

12.109.4.31 double tdiffmin

12.109.4.32 double tmax

12.109.4.33 double tmin

12.109.4.34 double utfirst

12.109.4.35 double utjd[90000]

12.109.4.36 double utjd1

12.109.4.37 double utjd2

12.109.4.38 double utjdlast

12.109.4.39 double utlast

12.109.4.40 double utmean

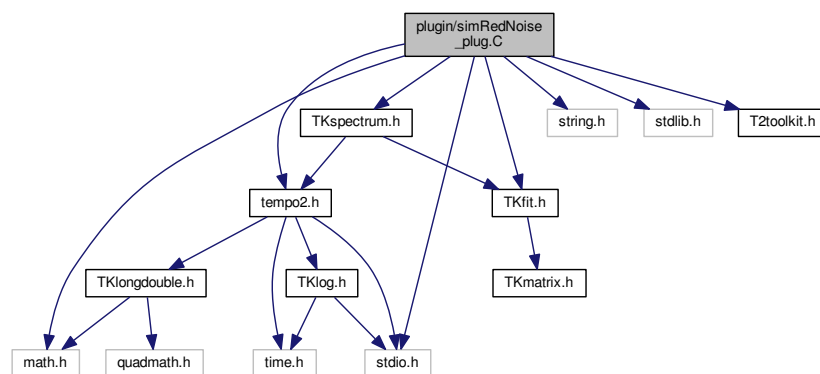
12.109.4.41 double xmax

12.109.4.42 double xmin

12.110 plugin/simRedNoise_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include "TKspectrum.h"
```

Include dependency graph for simRedNoise_plug.C:



Functions

- void [doPlugin](#) ([pulsar](#) *psr, int npsr, double amp, double [alpha](#), double fc, int removeQuad)
- void [getRedNoiseRealisation](#) ([pulsar](#) psr, double amp, double [alpha](#), double fc, long *seed, double *redNoise, int *nRedNoise, double *minx, double *delta)

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.110.1 Function Documentation

12.110.1.1 void [doPlugin](#) ([pulsar](#) * *psr*, int *npsr*, double *amp*, double *alpha*, double *fc*, int *removeQuad*)

12.110.1.2 void [getRedNoiseRealisation](#) ([pulsar](#) *psr*, double *amp*, double *alpha*, double *fc*, long * *seed*, double * *redNoise*, int * *nRedNoise*, double * *minx*, double * *delta*)

12.110.1.3 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.110.1.4 void [help](#) ()

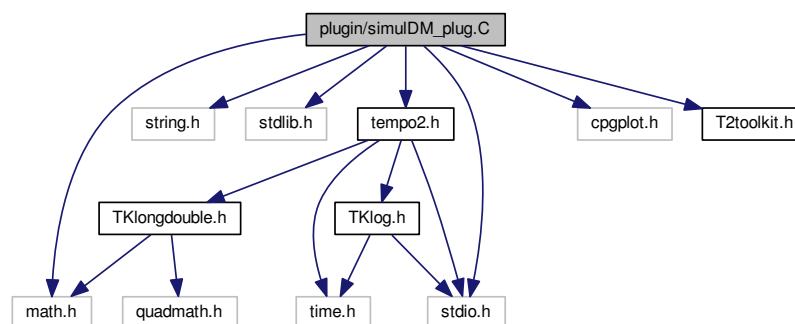
12.110.2 Variable Documentation

12.110.2.1 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.111 plugin/simulDM_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
#include "T2toolkit.h"
```

Include dependency graph for simulDM_plug.C:



Macros

- #define [MAX_DM](#) 1000

Functions

- void [help](#) ()
- void [doplugin](#) ([pulsar](#) *psr)
- void [doPlot](#) ([pulsar](#) *psr, int *highFreq_id, int *lowFreq_id, double *dmVal, double *timeVal, int nDM, double *dmValE)
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.111.1 Macro Definition Documentation

12.111.1.1 `#define MAX_DM 1000`

12.111.2 Function Documentation

12.111.2.1 void [doPlot](#) ([pulsar](#) * *psr*, int * *highFreq_id*, int * *lowFreq_id*, double * *dmVal*, double * *timeVal*, int *nDM*, double * *dmValE*)

12.111.2.2 void [doplugin](#) ([pulsar](#) * *psr*)

12.111.2.3 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)

12.111.2.4 void [help](#) ()

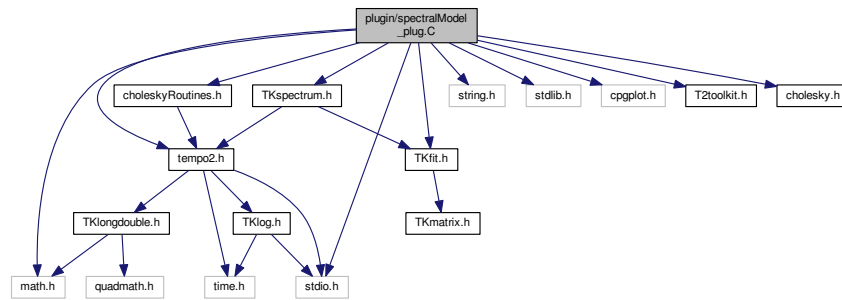
12.111.3 Variable Documentation

12.111.3.1 const char* [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.112 plugin/spectralModel_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include <cpqplot.h>
#include "T2toolkit.h"
#include "TKspectrum.h"
#include "TKfit.h"
#include "choleskyRoutines.h"
#include "cholesky.h"
```

Include dependency graph for spectralModel_plug.C:



Functions

- void [plot6](#) (double *cholSpecX, double *cholSpecY, int nCholSpec, double *cholWspecX, double *cholWspecY, int nCholWspec, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, int makeps)
- void [doPlugin](#) ([pulsar](#) *psr, double idt, int ipw, double ifc, double iexp, int inpt, int makeps, double amp, const char *dcf_file, int fc_auto, int nfit_auto, int exp_auto, int useBeta)
- int [obtainTimingResiduals](#) ([pulsar](#) *psr, double *resx, double *resy, double *rese, int *ip)
- void [fitSineFunc](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival)
- void [plot1](#) (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *smoothModel, double *highFreqRes, double *hfNormCovar, int *hfNormCovarNpts, double hfZerolagNormCovar)
- void [removeMean](#) (double *resx, double *resy, int n)
- void [fileOutput3](#) (const char *fname, double *x, double *y, double *z, int n)
- void [fileOutput2](#) (const char *fname, double *x, double *y, int n)
- void [findSmoothCurve](#) (double *resx, double *resy, double *rese, int nres, double *cubicVal, double *smoothModel, double expSmooth)
- void [getHighFreqRes](#) (double *resy, double *smoothModel, int nres, double *highFreqRes)
- void [getHighFreqCovar](#) (double *resx, double *rese, double *highFreqRes, int nres, double *hfNormCovar, int *hfNormCovarNpts, double *hfZerolagNormCovar)
- void [calculateDailyCovariance](#) (double *x, double *y, double *e, int n, double *cv, int *in, double *zl, int usew)
- int [calculateSpectra](#) (double *x, double *y, double *e, int n, int useErr, int preWhite, int specType, double *specX, double *specY)
- void [plot2](#) (double *origSpecX, double *origSpecY, int nOrigSpec, double *smoothSpecX0, double *smoothSpecY0, int nSmoothSpec0, double *smoothSpecX1, double *smoothSpecY1, int nSmoothSpec1, double *smoothSpecX2, double *smoothSpecY2, int nSmoothSpec2, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, int makeps)
- void [plot3](#) (double *preWhiteSpecX, double *preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, int closeit, float *minx, float *maxx, int useBeta, double betaVal)
- void [plot3a](#) (double *resx, double *resy, int nres, double *rawCovar, int *rawCovarNpts, double zerolagRawCovar, double *ampFit, double *chisqFit, int nGridFit, double bestAmp, double bestLag, double bestChisq, int makeps)
- void [plot4](#) (double *resx, double *resy, double *rese, int nres, double *cholWhiteY, double *whiteCovar, int *whiteCovarNpts, double zerolagWhiteCovar)
- void [plot5](#) (double *preWhiteSpecX, double *preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double *highFreqSpecX, double *highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, double nmodelScale, double *cholSpecX, double *cholSpecY, int nCholSpec, double *cholWspecX, double *cholWspecY, int nCholWspec, int makeps, double *cholWspecX2, double *cholWspecY2, int useBeta, double betaVal)
- void [outputMatrix](#) (double **uinv, int nres)

- void [fitExponential](#) (double *resx, int nres, double *rawCovar, int *rawCovarNpts, double *ampFit, double *chisqFit, double *bestAmp, double *bestLag, double *bestChisq, int *nGridFit)
- void [calculateCholeskyCovarFunc](#) (double bestAmp, double bestLag, int nGridFit, double **uinv, double *resx, double *resy, double *rese, int nres, double *covarFunc)
- void [outputCovarianceFunction](#) (double *covFunc, int n, double errorScaleFactor, [pulsar](#) *psr)
- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- char [pgdevice](#) [80]
- double [G_OMEGA](#)
- char [skipstep2](#) =0
- bool [writeFiles](#) =true
- int [skipprocess](#) =0
- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.112.1 Function Documentation

- 12.112.1.1 void [calculateCholeskyCovarFunc](#) (double *bestAmp*, double *bestLag*, int *nGridFit*, double ** *uinv*, double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *covarFunc*)
- 12.112.1.2 void [calculateDailyCovariance](#) (double * *x*, double * *y*, double * *e*, int *n*, double * *cv*, int * *in*, double * *zl*, int *usew*)
- 12.112.1.3 int [calculateSpectra](#) (double * *x*, double * *y*, double * *e*, int *n*, int *useErr*, int *preWhite*, int *specType*, double * *specX*, double * *specY*)
- 12.112.1.4 void [doPlugin](#) ([pulsar](#) * *psr*, double *idt*, int *ipw*, double *ifc*, double *iexp*, int *inpt*, int *makeps*, double *amp*, const char * *dcf_file*, int *fc_auto*, int *nfit_auto*, int *exp_auto*, int *useBeta*)
- 12.112.1.5 void [fileOutput2](#) (const char * *fname*, double * *x*, double * *y*, int *n*)
- 12.112.1.6 void [fileOutput3](#) (const char * *fname*, double * *x*, double * *y*, double * *z*, int *n*)
- 12.112.1.7 void [findSmoothCurve](#) (double * *resx*, double * *resy*, double * *rese*, int *nres*, double * *cubicVal*, double * *smoothModel*, double *expSmooth*)
- 12.112.1.8 void [fitExponential](#) (double * *resx*, int *nres*, double * *rawCovar*, int * *rawCovarNpts*, double * *ampFit*, double * *chisqFit*, double * *bestAmp*, double * *bestLag*, double * *bestChisq*, int * *nGridFit*)
- 12.112.1.9 void [fitSineFunc](#) (double *x*, double * *v*, int *nfit*, [pulsar](#) * *psr*, int *ival*)
- 12.112.1.10 void [getHighFreqCovar](#) (double * *resx*, double * *rese*, double * *highFreqRes*, int *nres*, double * *hfNormCovar*, int * *hfNormCovarNpts*, double * *hfZerolagNormCovar*)
- 12.112.1.11 void [getHighFreqRes](#) (double * *resy*, double * *smoothModel*, int *nres*, double * *highFreqRes*)
- 12.112.1.12 int [graphicalInterface](#) (int *argc*, char * *argv*[], [pulsar](#) * *psr*, int * *npsr*)
- 12.112.1.13 void [help](#) ()
- 12.112.1.14 int [obtainTimingResiduals](#) ([pulsar](#) * *psr*, double * *resx*, double * *resy*, double * *rese*, int * *ip*)

- 12.112.1.15 void outputCovarianceFunction (double * covFunc, int n, double errorScaleFactor, pulsar * psr)
- 12.112.1.16 void outputMatrix (double ** uinv, int nres)
- 12.112.1.17 void plot1 (double * resx, double * resy, double * rese, int nres, double * cubicVal, double * smoothModel, double * highFreqRes, double * hfNormCovar, int * hfNormCovarNpts, double hfZerolagNormCovar)
- 12.112.1.18 void plot2 (double * origSpecX, double * origSpecY, int nOrigSpec, double * smoothSpecX0, double * smoothSpecY0, int nSmoothSpec0, double * smoothSpecX1, double * smoothSpecY1, int nSmoothSpec1, double * smoothSpecX2, double * smoothSpecY2, int nSmoothSpec2, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, int makeps)
- 12.112.1.19 void plot3 (double * preWhiteSpecX, double * preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, int closeit, float * minx, float * maxx, int useBeta, double betaVal)
- 12.112.1.20 void plot3a (double * resx, double * resy, int nres, double * rawCovar, int * rawCovarNpts, double zerolagRawCovar, double * ampFit, double * chisqFit, int nGridFit, double bestAmp, double bestLag, double bestChisq, int makeps)
- 12.112.1.21 void plot4 (double * resx, double * resy, double * rese, int nres, double * cholWhiteY, double * whiteCovar, int * whiteCovarNpts, double zerolagWhiteCovar)
- 12.112.1.22 void plot5 (double * preWhiteSpecX, double * preWhiteSpecY, int nPreWhiteSpec, int usePreWhitening, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, double modelAlpha, double modelFc, int modelNfit, double modelScale, double nmodelScale, double * cholSpecX, double * cholSpecY, int nCholSpec, double * cholWspecX, double * cholWspecY, int nCholWspec, int makeps, double * cholWspecX2, double * cholWspecY2, int useBeta, double betaVal)
- 12.112.1.23 void plot6 (double * cholSpecX, double * cholSpecY, int nCholSpec, double * cholWspecX, double * cholWspecY, int nCholWspec, double * highFreqSpecX, double * highFreqSpecY, int nHighFreqSpec, int makeps)
- 12.112.1.24 void removeMean (double * resx, double * resy, int n)

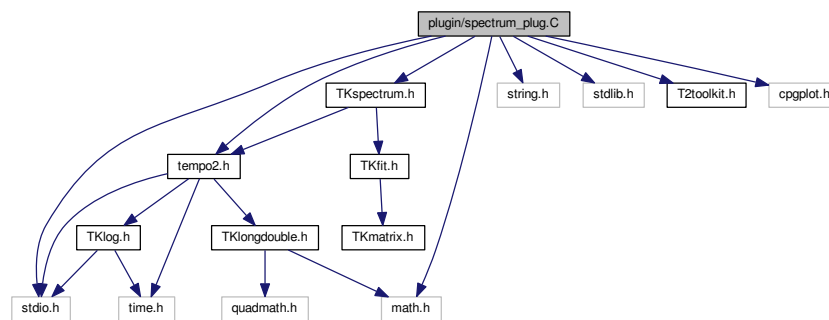
12.112.2 Variable Documentation

- 12.112.2.1 double G_OMEGA
- 12.112.2.2 char pgdevice[80]
- 12.112.2.3 const char* plugVersionCheck = TEMPO2_h_VER
- 12.112.2.4 int skipprocess=0
- 12.112.2.5 char skipstep2=0
- 12.112.2.6 bool writeFiles=true

12.113 plugin/spectrum_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKspectrum.h"
#include <cpgplot.h>
```

Include dependency graph for spectrum_plug.C:



Macros

- #define `MAX_ID` 50

Functions

- void `drawOption` (int on, float x, float y, const char *str)
- void `doPlugin` (pulsar *psr, int npsr, char parFile[MAX_PSR_VAL][MAX_FILELEN], char timFile[MAX_PSR_VAL][MAX_FILELEN], int white, int filter)
- void `drawMenu` (int specType, int xaxis, int logv, int specOut)
- void `checkMenu` (float mx, float my, int *change, int *xaxis, int *logv, int *specType, int *specOut)
- void `identify` (float mx, float my, float px[MAX_PSR_VAL][MAX_OBSN_VAL], float py[MAX_PSR_VAL][MAX_OBSN_VAL], int *sn, int idV[MAX_ID], int idP[MAX_ID], int *iN, int npsr)
- void `model` (pulsar *psr, char parFile[MAX_PSR_VAL][MAX_FILELEN], char timFile[MAX_PSR_VAL][MAX_FILELEN])
- void `help` ()
- int `graphicalInterface` (int argc, char *argv[], pulsar *psr, int *npsr)

Variables

- char `dcmFile` [MAX_FILELEN]
- char `covarFuncFile` [MAX_FILELEN]
- const char * `plugVersionCheck` = `TEMPO2_h_VER`

12.113.1 Macro Definition Documentation

12.113.1.1 `#define MAX_ID 50`

12.113.2 Function Documentation

12.113.2.1 `void checkMenu (float mx, float my, int * change, int * xaxis, int * logv, int * specType, int * specOut)`

12.113.2.2 `void doPlugin (pulsar * psr, int npsr, char parFile[MAX_PSR_VAL][MAX_FILELEN], char timFile[MAX_PSR_VAL][MAX_FILELEN], int white, int filter)`

12.113.2.3 `void drawMenu (int specType, int xaxis, int logv, int specOut)`

12.113.2.4 `void drawOption (int on, float x, float y, const char * str)`

12.113.2.5 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.113.2.6 `void help ()`

12.113.2.7 `void identify (float mx, float my, float px[MAX_PSR_VAL][MAX_OBSN_VAL], float py[MAX_PSR_VAL][MAX_OBSN_VAL], int * sn, int idV[MAX_ID], int idP[MAX_ID], int * iN, int npsr)`

12.113.2.8 `void model (pulsar * psr, char parFile[MAX_PSR_VAL][MAX_FILELEN], char timFile[MAX_PSR_VAL][MAX_FILELEN])`

12.113.3 Variable Documentation

12.113.3.1 `char covarFuncFile[MAX_FILELEN]`

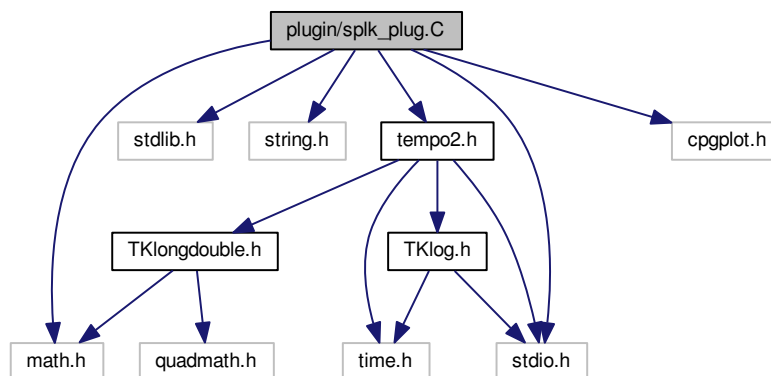
12.113.3.2 `char dcmFile[MAX_FILELEN]`

12.113.3.3 `const char* plugVersionCheck = TEMPO2_h_VER`

12.114 plugin/splk_plug.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
#include <cpgplot.h>
```

Include dependency graph for splk_plug.C:



Functions

- void [doPlot](#) ([pulsar](#) *psr, int npsr, int overlay)
- float [findMin](#) (float *x, [pulsar](#) *psr, int p, int i1, int i2)
- float [findMax](#) (float *x, [pulsar](#) *psr, int p, int i1, int i2)
- float [findMean](#) (float *x, [pulsar](#) *psr, int p, int i1, int i2)
- void [callFit](#) ([pulsar](#) *psr, int npsr)
- float [deletePoint](#) ([pulsar](#) *psr, int npsr, float *x, float *y, float mouseX, float mouseY)
- float [idPoint](#) ([pulsar](#) *psr, int npsr, float *x, float *y, float mouseX, float mouseY)
- double [fortranMod](#) (double a, double p)
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- char [dcmFile](#) [MAX_FILELEN]
- char [covarFuncFile](#) [MAX_FILELEN]
- const char * [plugVersionCheck](#) = TEMPO2_h_VER

12.114.1 Function Documentation

12.114.1.1 void [callFit](#) ([pulsar](#) * *psr*, int *npsr*)

12.114.1.2 float [deletePoint](#) ([pulsar](#) * *psr*, int *npsr*, float * *x*, float * *y*, float *mouseX*, float *mouseY*)

12.114.1.3 void [doPlot](#) ([pulsar](#) * *psr*, int *npsr*, int *overlay*)

12.114.1.4 float [findMax](#) (float * *x*, [pulsar](#) * *psr*, int *p*, int *i1*, int *i2*)

12.114.1.5 float [findMean](#) (float * *x*, [pulsar](#) * *psr*, int *p*, int *i1*, int *i2*)

12.114.1.6 float [findMin](#) (float * *x*, [pulsar](#) * *psr*, int *p*, int *i1*, int *i2*)

12.114.1.7 double [fortranMod](#) (double *a*, double *p*)

12.114.1.8 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.114.1.9 `float idPoint (pulsar * psr, int npsr, float * x, float * y, float mouseX, float mouseY)`

12.114.2 Variable Documentation

12.114.2.1 `char covarFuncFile[MAX_FILELEN]`

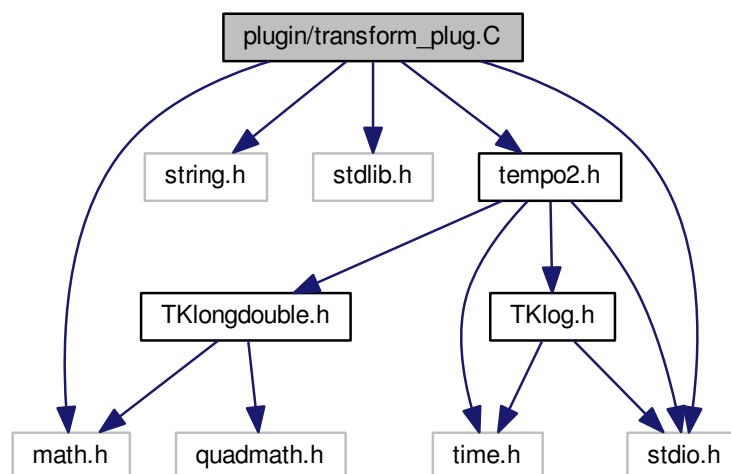
12.114.2.2 `char dcmFile[MAX_FILELEN]`

12.114.2.3 `const char* plugVersionCheck = TEMPO2_h_VER`

12.115 plugin/transform_plug.C File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for transform_plug.C:



Functions

- void [help](#) ()
- int [graphicalInterface](#) (int argc, char *argv[], [pulsar](#) *psr, int *npsr)

Variables

- const char * [plugVersionCheck](#) = [TEMPO2_h_VER](#)

12.115.1 Function Documentation

12.115.1.1 `int graphicalInterface (int argc, char * argv[], pulsar * psr, int * npsr)`

12.115.1.2 `void help ()`

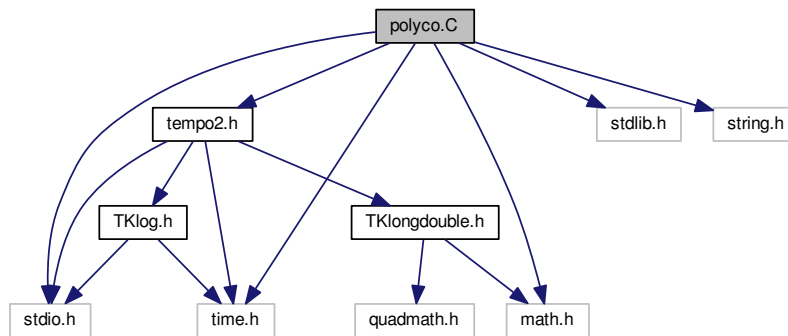
12.115.2 Variable Documentation

12.115.2.1 `const char* plugVersionCheck = TEMPO2_h_VER`

12.116 polyco.C File Reference

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"
#include <time.h>
```

Include dependency graph for polyco.C:



Functions

- void `atimfake` (`pulsar *psr`, `int npsr`, `int tspan`, `int ncoeff`, `longdouble maxha`, `char *sitename`, `longdouble freq`, `longdouble afmjd`, `longdouble *tmin`, `longdouble *tmidMJD`, `int *retTspan`, `int *nsets`, `longdouble *val`)
- void `tzFit` (`pulsar *psr`, `int npsr`, `longdouble *tmin`, `double *doppler`, `double *rms`, `double *utc`, `longdouble tmidMJD`, `int ncoeff`, `longdouble *coeff`, `char *binPhase`, `int nsets`, `longdouble afmjd`, `char *sitename`, `int dspan`, `double obsFreq`, `char *date`, `longdouble *val`, `int trueDM`, `char *polyco_file`)
- void `pcshft` (`longdouble a`, `longdouble b`, `longdouble *d`, `int n`)
- void `chebpc` (`longdouble *c`, `longdouble *d`, `int n`)
- void `polyco` (`pulsar *psr`, `int npsr`, `longdouble polyco_MJD1`, `longdouble polyco_MJD2`, `int nspan`, `int ncoeff`, `longdouble maxha`, `char *sitename`, `longdouble freq`, `longdouble coeff[MAX_COEFF]`, `int trueDM`, `char *polyco_file`)

12.116.1 Function Documentation

12.116.1.1 `void atimfake (pulsar * psr, int npsr, int tspan, int ncoeff, longdouble maxha, char * sitename, longdouble freq, longdouble afmjd, longdouble * tmin, longdouble * tmidMJD, int * retTspan, int * nsets, longdouble * val)`

12.116.1.2 `void chebpc (longdouble * c, longdouble * d, int n)`

12.116.1.3 void pcsht (longdouble *a*, longdouble *b*, longdouble * *d*, int *n*)

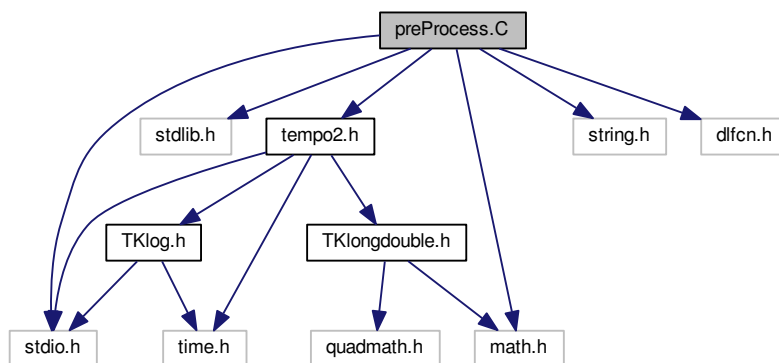
12.116.1.4 void polyco (pulsar * *psr*, int *npsr*, longdouble *polyco_MJD1*, longdouble *polyco_MJD2*, int *nspan*, int *ncoeff*, longdouble *maxha*, char * *sitename*, longdouble *freq*, longdouble *coeff*[MAX_COEFF], int *trueDM*, char * *polyco_file*)

12.116.1.5 void tzFit (pulsar * *psr*, int *npsr*, longdouble * *tmin*, double * *doppler*, double * *rms*, double * *utc*, longdouble *tmidMJD*, int *ncoeff*, longdouble * *coeff*, char * *binPhase*, int *nsets*, longdouble *afmjd*, char * *sitename*, int *dspan*, double *obsFreq*, char * *date*, longdouble * *val*, int *trueDM*, char * *polyco_file*)

12.117 preProcess.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "tempo2.h"
#include <math.h>
#include <string.h>
#include <dlfcn.h>
```

Include dependency graph for preProcess.C:



Functions

- void [readWhiteNoiseModelFile](#) (pulsar *psr, int p)
- void [preProcess](#) (pulsar *psr, int npsr, int argc, char **argv)
- void [useSelectFile](#) (char *fname, pulsar *psr, int npsr)
- void [logicFlag](#) (char *line, pulsar *psr, int npsr)
- void [processFlag](#) (char *line, pulsar *psr, int npsr)
- void [processSimultaneous](#) (char *line, pulsar *psr, int npsr)

12.117.1 Function Documentation

12.117.1.1 void logicFlag (char * *line*, pulsar * *psr*, int *npsr*)

12.117.1.2 void preProcess (pulsar * *psr*, int *npsr*, int *argc*, char ** *argv*)

12.117.1.3 void processFlag (char * *line*, pulsar * *psr*, int *npsr*)

12.117.1.4 void processSimultaneous (char * *line*, pulsar * *psr*, int *npsr*)

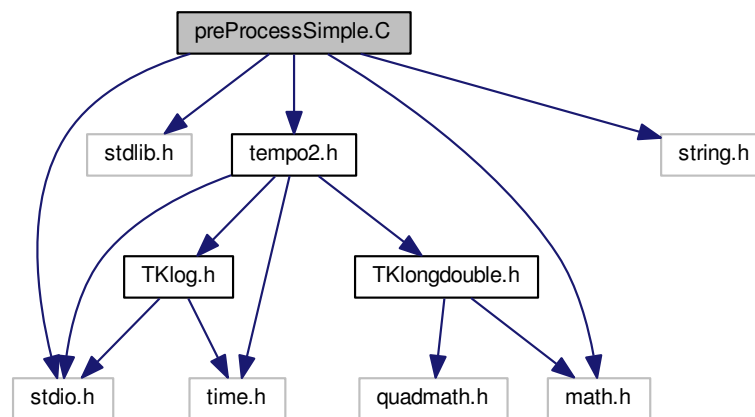
12.117.1.5 void readWhiteNoiseModelFile (pulsar * *psr*, int *p*)

12.117.1.6 void useSelectFile (char * *fname*, pulsar * *psr*, int *npsr*)

12.118 preProcessSimple.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "tempo2.h"
#include <math.h>
#include <string.h>
```

Include dependency graph for preProcessSimple.C:



Functions

- void [preProcessSimple](#) (pulsar *psr)
- void [preProcessSimple1](#) (pulsar *psr, int tempo1, double thelast)
- void [preProcessSimple2](#) (pulsar *psr, float startdmmjd, int ndm, float *dmvals, int trimonly)
- void [preProcessSimple3](#) (pulsar *psr)

12.118.1 Function Documentation

12.118.1.1 void preProcessSimple (pulsar * *psr*)

12.118.1.2 void preProcessSimple1 (pulsar * *psr*, int *tempo1*, double *thelast*)

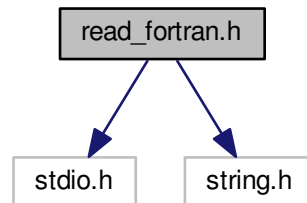
12.118.1.3 void preProcessSimple2 (pulsar * *psr*, float *startdmmjd*, int *ndm*, float * *dmvals*, int *trimonly*)

12.118.1.4 void preProcessSimple3 (pulsar * *psr*)

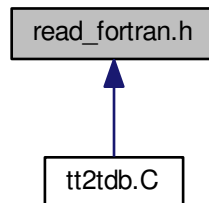
12.119 read_fortran.h File Reference

```
#include <stdio.h>
#include <string.h>
```

Include dependency graph for read_fortran.h:



This graph shows which files directly or indirectly include this file:



Functions

- int [open_file](#) (char *fname)
- void [close_file](#) ()
- void [read_character](#) (int len, char *str)
- char [read_char](#) ()
- int [read_int](#) ()
- float [read_float](#) ()
- double [read_double](#) ()
- int [read_record_int](#) ()

Variables

- FILE * [c_fileptr](#)
- int [swapByte](#)

12.119.1 Function Documentation

12.119.1.1 void close_file ()

12.119.1.2 int open_file (char * *fname*)

12.119.1.3 char read_char ()

12.119.1.4 void read_character (int *len*, char * *str*)

12.119.1.5 double read_double ()

12.119.1.6 float read_float ()

12.119.1.7 int read_int ()

12.119.1.8 int read_record_int ()

12.119.2 Variable Documentation

12.119.2.1 FILE* c_fileptr

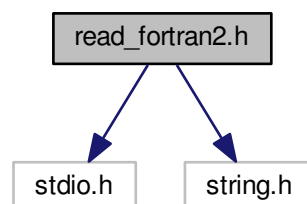
12.119.2.2 int swapByte

12.120 read_fortran2.h File Reference

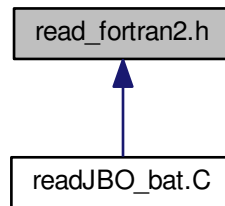
```
#include <stdio.h>
```

```
#include <string.h>
```

Include dependency graph for read_fortran2.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [open_file2](#) (char *fname, int *swap)
- void [close_file2](#) ()
- void [read_character2](#) (int len, char *str)
- int [read_int2](#) ()
- float [read_float2](#) ()
- double [read_double2](#) ()
- int [read_record_int2](#) ()

Variables

- FILE * [c_fileptr2](#)
- int [swapByte2](#)

12.120.1 Function Documentation

12.120.1.1 void [close_file2](#) ()

12.120.1.2 void [open_file2](#) (char * *fname*, int * *swap*)

12.120.1.3 void [read_character2](#) (int *len*, char * *str*)

12.120.1.4 double [read_double2](#) ()

12.120.1.5 float [read_float2](#) ()

12.120.1.6 int [read_int2](#) ()

12.120.1.7 int [read_record_int2](#) ()

12.120.2 Variable Documentation

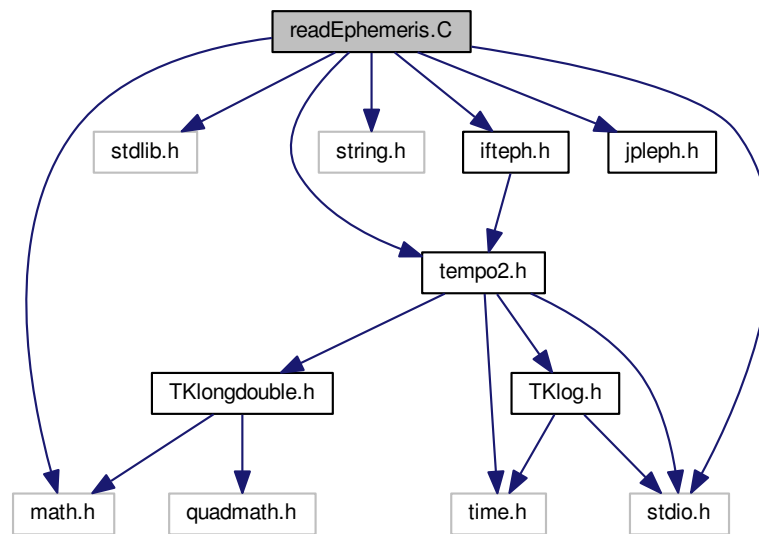
12.120.2.1 FILE* [c_fileptr2](#)

12.120.2.2 int [swapByte2](#)

12.121 readEphemeris.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "tempo2.h"
#include "jpleph.h"
#include "ifteph.h"
```

Include dependency graph for readEphemeris.C:



Macros

- `#define MAX_SHOTS 10000`

Functions

- double `random2` (long *idum)
- double `gasdev` (long *idum)
- void `readEphemeris` (pulsar *psr, int npsr, int addEphemNoise)

12.121.1 Macro Definition Documentation

12.121.1.1 `#define MAX_SHOTS 10000`

12.121.2 Function Documentation

12.121.2.1 double `gasdev` (long * *idum*)

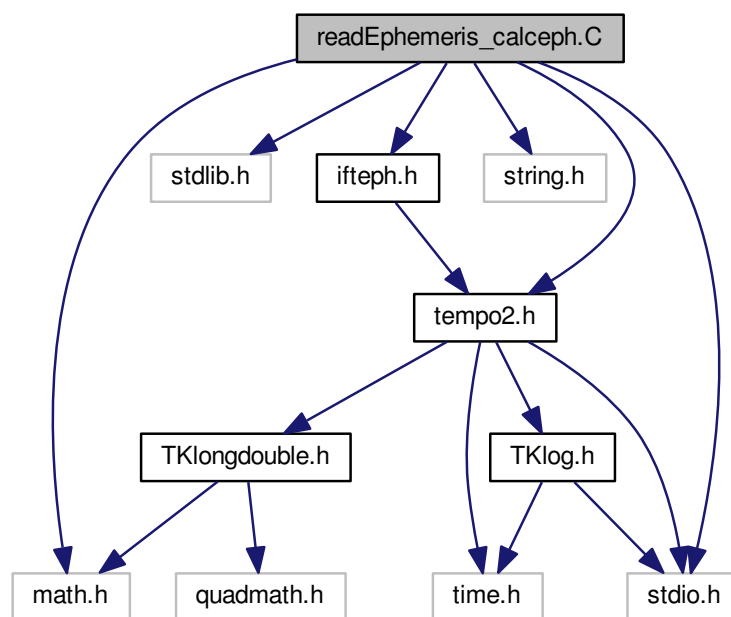
12.121.2.2 double `random2` (long * *idum*)

12.121.2.3 void readEphemeris (pulsar * psr, int npsr, int addEphemNoise)

12.122 readEphemeris_calceph.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "tempo2.h"
#include "ifteph.h"
```

Include dependency graph for readEphemeris_calceph.C:



Functions

- void [convertUnits](#) (double *val, int units)
- void [readEphemeris_calceph](#) (pulsar *psr, int npsr)

12.122.1 Function Documentation

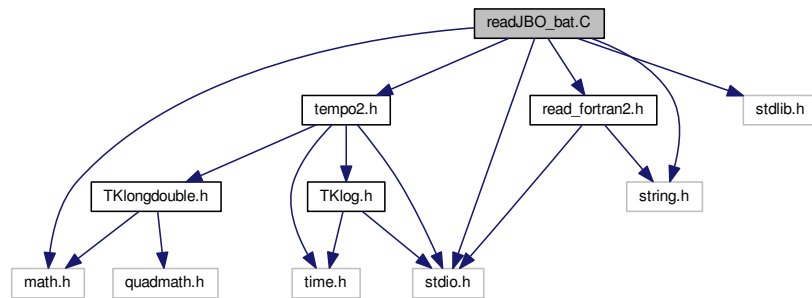
12.122.1.1 void [convertUnits](#) (double * val, int units)

12.122.1.2 void [readEphemeris_calceph](#) (pulsar * psr, int npsr)

12.123 readJBO_bat.C File Reference

```
#include <math.h>
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"
#include "read_fortran2.h"
Include dependency graph for readJBO_bat.C:
```



Functions

- double [date2mjd](#) (int idat)
- void [makechars](#) (char *raw, char cbuf[25][9])
- void [swap4](#) (char *)
- void [swap8](#) (char *)
- void [swap8](#) (double *in)
- void [readJBO_bat](#) (char *fname, [pulsar](#) *psr, int p)

12.123.1 Function Documentation

12.123.1.1 double [date2mjd](#) (int *idat*)

12.123.1.2 void [makechars](#) (char * *raw*, char *cbuf*[25][9])

12.123.1.3 void [readJBO_bat](#) (char * *fname*, [pulsar](#) * *psr*, int *p*)

12.123.1.4 void [swap4](#) (char * *raw*)

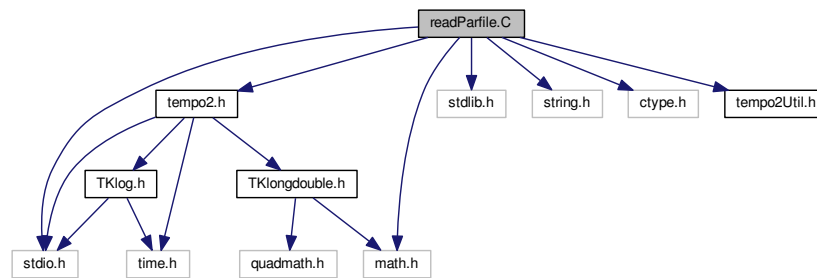
12.123.1.5 void [swap8](#) (char * *raw*)

12.123.1.6 void [swap8](#) (double * *in*)

12.124 readParfile.C File Reference

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "tempo2.h"
#include "tempo2Util.h"
```

Include dependency graph for readParfile.C:



Functions

- int [readValue](#) (pulsar *psr, char *pmtr, FILE *fin, [parameter](#) *parameter, int arr)
- void [getValue](#) (char *str, int v1, int v2, [pulsar](#) *psr, int l, int arr)
- void [removeCR](#) (char *str)
- void [checkLine](#) (pulsar *p, char *str, FILE *fin, [parameter](#) *elong, [parameter](#) *elat)
- void [checkAllSet](#) (pulsar *psr, [parameter](#) elong, [parameter](#) elat, char *filename)
- int [setupParameterFileDefaults](#) (pulsar *psr)
- int [readSimpleParfile](#) (FILE *fin, [pulsar](#) *p)
- void [readParfileGlobal](#) (pulsar *psr, int npsr, char tpar[[MAX_STRLEN](#)][[MAX_FILELEN](#)], char ttim[[MAX_STRLEN](#)][[MAX_FILELEN](#)])
- void [readParfile](#) (pulsar *psr, char parFile[][[MAX_FILELEN](#)], char timFile[][[MAX_FILELEN](#)], int npsr)

12.124.1 Function Documentation

12.124.1.1 void [checkAllSet](#) (pulsar * psr, parameter elong, parameter elat, char * filename)

12.124.1.2 void [checkLine](#) (pulsar * p, char * str, FILE * fin, parameter * elong, parameter * elat)

12.124.1.3 void [getValue](#) (char * str, int v1, int v2, pulsar * psr, int l, int arr)

12.124.1.4 void [readParfile](#) (pulsar * psr, char parFile[][[MAX_FILELEN](#)], char timFile[][[MAX_FILELEN](#)], int npsr)

12.124.1.5 void [readParfileGlobal](#) (pulsar * psr, int npsr, char tpar[[MAX_STRLEN](#)][[MAX_FILELEN](#)], char ttim[[MAX_STRLEN](#)][[MAX_FILELEN](#)])

12.124.1.6 int [readSimpleParfile](#) (FILE * fin, pulsar * p)

12.124.1.7 int [readValue](#) (pulsar * psr, char * pmtr, FILE * fin, parameter * parameter, int arr)

12.124.1.8 void [removeCR](#) (char * str)

12.124.1.9 int [setupParameterFileDefaults](#) (pulsar * psr)

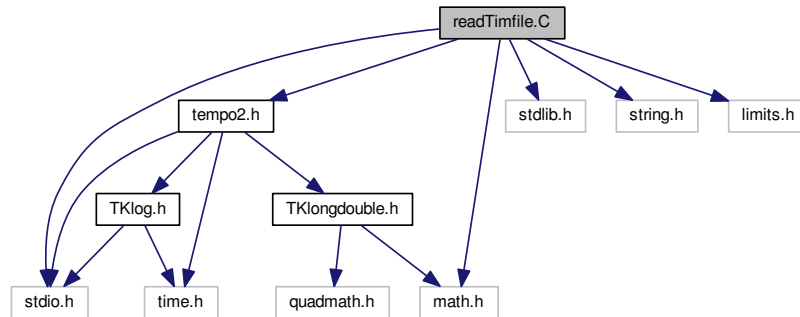
12.125 readTimfile.C File Reference

```
#include <math.h>
```



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
#include "tempo2.h"
```

Include dependency graph for readTimfile.C:



Functions

- void `readTim` (char *timname, [pulsar](#) *psr, int *jumpVal)
- void `removeCR2` (char *str)
- void `readTimfile` ([pulsar](#) *psr, char timFile[][MAX_FILELEN], int npsr)
- void `writeTim` (const char *timname, [pulsar](#) *psr, const char *fileFormat)

12.125.1 Function Documentation

12.125.1.1 void `readTim` (char * *timname*, [pulsar](#) * *psr*, int * *jumpVal*)

12.125.1.2 void `readTimfile` ([pulsar](#) * *psr*, char *timFile*[][MAX_FILELEN], int *npsr*)

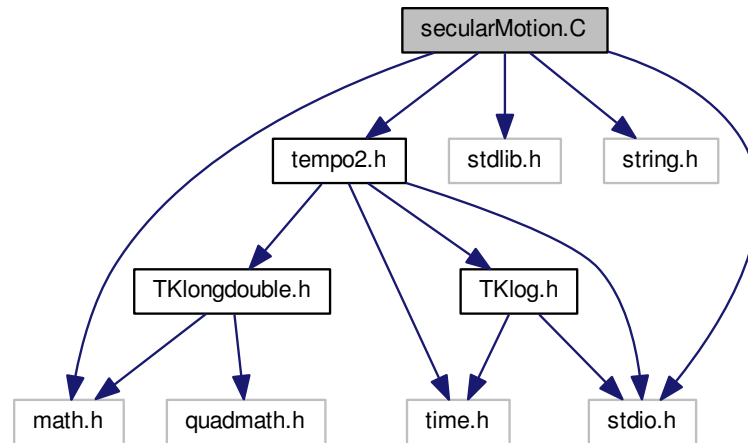
12.125.1.3 void `removeCR2` (char * *str*)

12.125.1.4 void `writeTim` (const char * *timname*, [pulsar](#) * *psr*, const char * *fileFormat*)

12.126 secularMotion.C File Reference

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"
```

Include dependency graph for secularMotion.C:



Functions

- void `secularMotion` (`pulsar` *psr, int npsr)

12.126.1 Function Documentation

12.126.1.1 void `secularMotion` (`pulsar` * *psr*, int *npsr*)

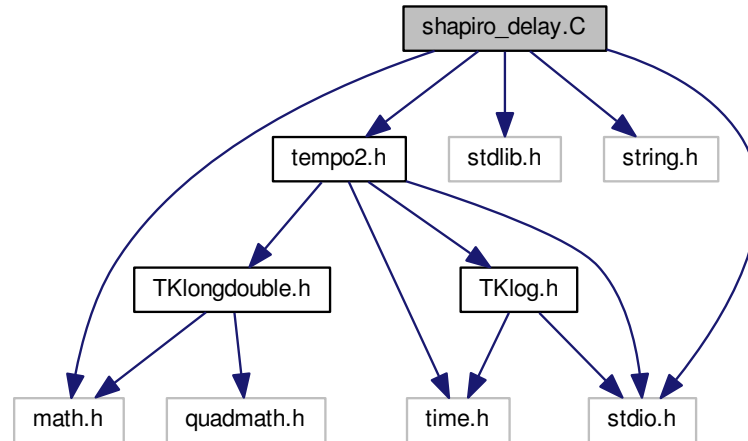
12.127 shapiro_delay.C File Reference

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"

```

Include dependency graph for shapiro_delay.C:



Functions

- void [shapiro_delay](#) ([pulsar](#) *psr, int npsr, int p, int i, double delt, double dt_SSB)

12.127.1 Function Documentation

12.127.1.1 void shapiro_delay ([pulsar](#) * *psr*, int *npsr*, int *p*, int *i*, double *delt*, double *dt_SSB*)

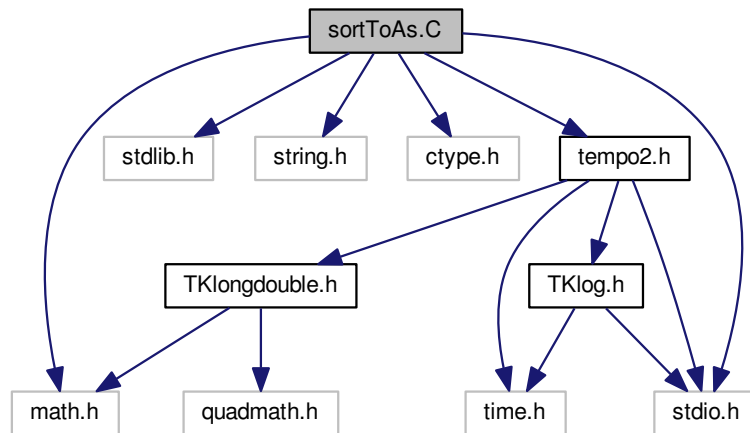
12.128 sortToAs.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include <math.h>
#include <tempo2.h>

```

Include dependency graph for sortToAs.C:



Functions

- int [compareObs](#) (const void *o1, const void *o2)
- void [sortToAs](#) ([pulsar](#) *psr)

12.128.1 Function Documentation

12.128.1.1 int [compareObs](#) (const void * o1, const void * o2)

12.128.1.2 void [sortToAs](#) ([pulsar](#) * psr)

Sort ToAs for one pulsar.

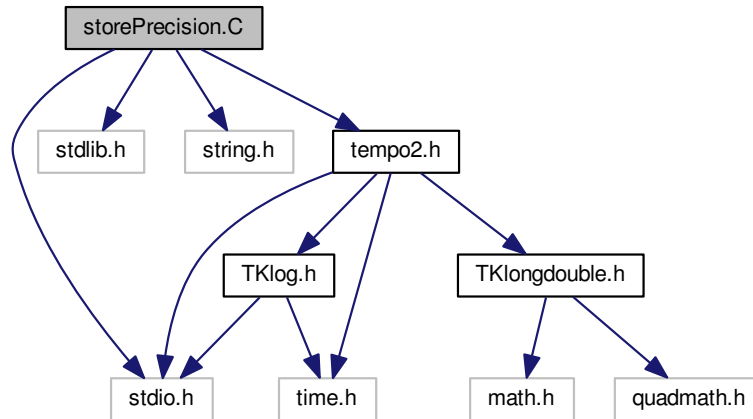
12.129 storePrecision.C File Reference

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"

```

Include dependency graph for storePrecision.C:



Functions

- void [recordPrecision](#) (pulsar *psr, longdouble prec, const char *routine, const char *comment)

12.129.1 Function Documentation

12.129.1.1 void [recordPrecision](#) (pulsar * *psr*, longdouble *prec*, const char * *routine*, const char * *comment*)

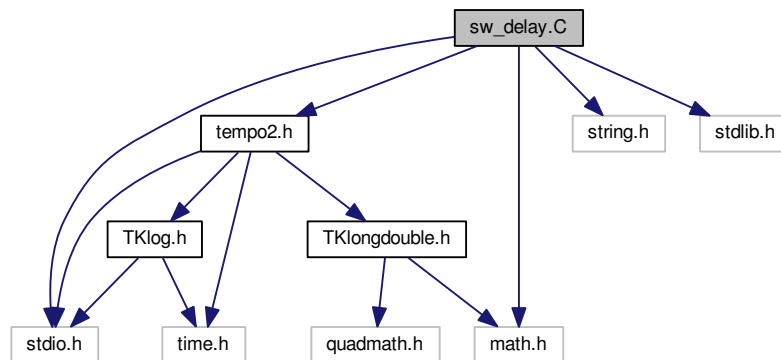
12.130 sw_delay.C File Reference

```

#include <math.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "tempo2.h"

```

Include dependency graph for `sw_delay.C`:



Macros

- `#define MAX_CURRENT 500`

Functions

- void `mcl2` (double `eclon`, double `eclat`, int `iy`, int `iday`, double `secs`, double `vel`, double `helat`[36], double `crlon`[36], double `rots`[36], double `*helate`, double `*crline`, double `*rote`, double `*elong`, double `*beta`, double `*dlon`, double `*delng`, int `*nLineOfSight`)
- double `amod` (double `a`, double `p`)
- double `elsun2` (int `iy`, int `iday`, double `secs`, double `*gst`, double `*sra`, double `*sdec`)
- void `outputResults` (double `*crlon`, double `*helat`)
- void `calcRotN` (double `crline`, double `rote`, int `*irot1`, int `*irot2`, double `*bcrlon`)
- int `readCurrentSheet` (char `*fname`, float `*currentLon`, float `*currentLat`)
- double `findAngle` (double `lon1`, double `lat1`, double `lon2`, double `lat2`)
- void `mjd2date` (int `mjd`, int `*iy`, int `*yy`, int `*mm`, int `*dd`, int `*iday`)
- void `convertEcliptic` (double `raj`, double `decj`, double `*elong`, double `*elat`)
- double `solarWindModel` (pulsar `psr`, int `iobs`)

12.130.1 Macro Definition Documentation

12.130.1.1 `#define MAX_CURRENT 500`

12.130.2 Function Documentation

12.130.2.1 double `amod` (double `a`, double `p`)

12.130.2.2 void `calcRotN` (double `crline`, double `rote`, int `* irot1`, int `* irot2`, double `* bcrlon`)

12.130.2.3 void `convertEcliptic` (double `raj`, double `decj`, double `* elong`, double `* elat`)

12.130.2.4 double `elsun2` (int `iy`, int `iday`, double `secs`, double `* gst`, double `* sra`, double `* sdec`)

12.130.2.5 double `findAngle` (double `lon1`, double `lat1`, double `lon2`, double `lat2`)

12.130.2.6 void mcl2 (double *eclon*, double *eclat*, int *iy*r, int *iday*, double *secs*, double *vel*, double *helat*[36], double *crlon*[36], double *rots*[36], double * *helate*, double * *crline*, double * *rote*, double * *elong*, double * *beta*, double * *dlon*, double * *delng*, int * *nLineOfSight*)

12.130.2.7 void mjd2date (int *mjd*, int * *iy*r, int * *yy*, int * *mm*, int * *dd*, int * *iday*)

12.130.2.8 void outputResults (double * *crlon*, double * *helat*)

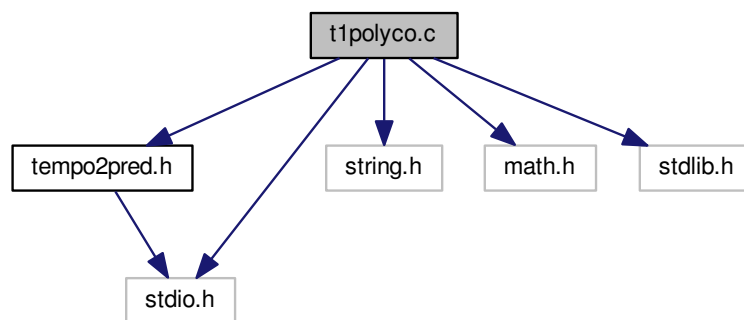
12.130.2.9 int readCurrentSheet (char * *fname*, float * *currentLon*, float * *currentLat*)

12.130.2.10 double solarWindModel (pulsar *psr*, int *iobs*)

12.131 t1polyco.c File Reference

```
#include "tempo2pred.h"
#include <string.h>
#include <math.h>
#include <stdlib.h>
#include <stdio.h>
```

Include dependency graph for t1polyco.c:



Functions

- long double [T1Polyco_GetPhase](#) (T1Polyco *t1p, long double mjd, long double freq)
- long double [T1Polyco_GetFrequency](#) (T1Polyco *t1p, long double mjd, long double freq)
- void [T1Polyco_Write](#) (T1Polyco *t1p, FILE *fout)
- void [T1P_grabString](#) (char *str, int istart, int nchar, char *out)
- long double [T1P_grabLongDouble](#) (char *str, int istart, int nchar)
- long double [T1P_grabInt](#) (char *str, int istart, int nchar)
- int [T1Polyco_Read_NewFormat](#) (T1Polyco *t1p, FILE *f)
- int [T1Polyco_Read](#) (T1Polyco *t1p, FILE *f)
- T1Polyco * [T1PolycoSet_GetNearest](#) (T1PolycoSet *t1ps, long double mjd)
- long double [T1PolycoSet_GetPhase](#) (T1PolycoSet *t1ps, long double mjd, long double freq)
- long double [T1PolycoSet_GetFrequency](#) (T1PolycoSet *t1ps, long double mjd, long double freq)
- void [T1PolycoSet_Write](#) (T1PolycoSet *t1ps, FILE *f)
- int [T1PolycoSet_Read](#) (T1PolycoSet *t1ps, FILE *f)
- void [T1PolycoSet_Destroy](#) (T1PolycoSet *t1ps)

12.131.1 Function Documentation

12.131.1.1 `long double T1P_grabInt (char * str, int istart, int nchar)`

12.131.1.2 `long double T1P_grabLongDouble (char * str, int istart, int nchar)`

12.131.1.3 `void T1P_grabString (char * str, int istart, int nchar, char * out)`

12.131.1.4 `long double T1Polyco_GetFrequency (T1Polyco * t1p, long double mjd, long double freq)`

12.131.1.5 `long double T1Polyco_GetPhase (T1Polyco * t1p, long double mjd, long double freq)`

12.131.1.6 `int T1Polyco_Read (T1Polyco * t1p, FILE * f)`

12.131.1.7 `int T1Polyco_Read_NewFormat (T1Polyco * t1p, FILE * f)`

12.131.1.8 `void T1Polyco_Write (T1Polyco * t1p, FILE * fout)`

12.131.1.9 `void T1PolycoSet_Destroy (T1PolycoSet * t1ps)`

12.131.1.10 `long double T1PolycoSet_GetFrequency (T1PolycoSet * t1ps, long double mjd, long double freq)`

12.131.1.11 `T1Polyco* T1PolycoSet_GetNearest (T1PolycoSet * t1ps, long double mjd)`

12.131.1.12 `long double T1PolycoSet_GetPhase (T1PolycoSet * t1ps, long double mjd, long double freq)`

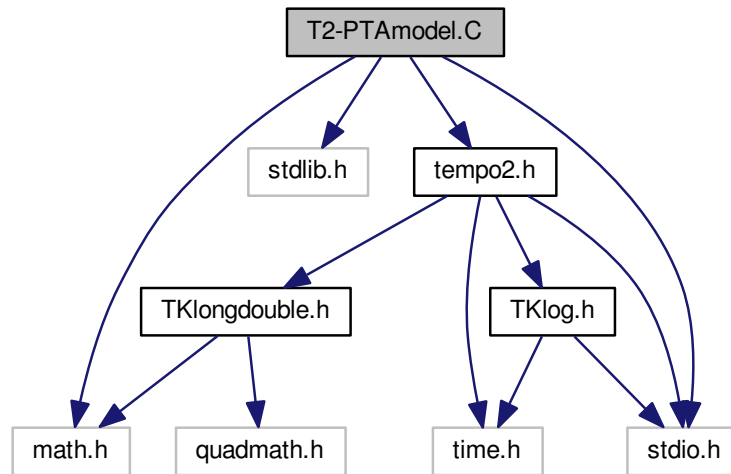
12.131.1.13 `int T1PolycoSet_Read (T1PolycoSet * t1ps, FILE * f)`

12.131.1.14 `void T1PolycoSet_Write (T1PolycoSet * t1ps, FILE * f)`

12.132 T2-PTAmodel.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```


Include dependency graph for T2-PTAmodel.C:



Functions

- [longdouble computeU](#) ([longdouble](#) phase, [longdouble](#) e)
- [double T2_PTAmodel](#) ([pulsar](#) *psr, int p, int ipos, int param, int arr)
- [void updateT2_PTA](#) ([pulsar](#) *psr, double val, double err, int pos, int arr)

12.132.1 Function Documentation

12.132.1.1 [longdouble computeU](#) ([longdouble](#) *phase*, [longdouble](#) *e*)

12.132.1.2 [double T2_PTAmodel](#) ([pulsar](#) * *psr*, int *p*, int *ipos*, int *param*, int *arr*)

12.132.1.3 [void updateT2_PTA](#) ([pulsar](#) * *psr*, double *val*, double *err*, int *pos*, int *arr*)

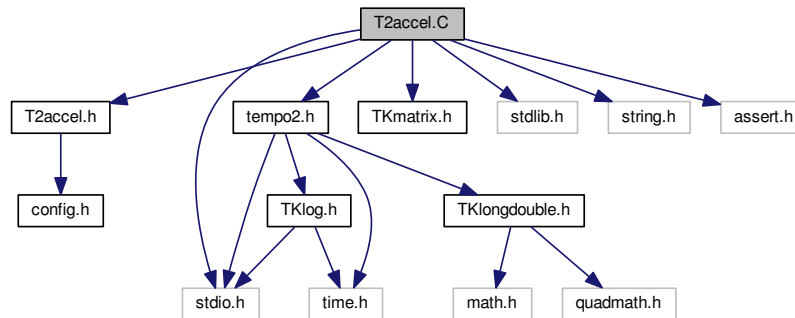
12.133 T2accel.C File Reference

```

#include "T2accel.h"
#include "tempo2.h"
#include "TKmatrix.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <assert.h>

```

Include dependency graph for T2accel.C:



Macros

- `#define F77_dpof2 F77_FUNC (dpof2, DPOTF2)`
- `#define F77_dtptri F77_FUNC (dtptri, DTPTRI)`
- `#define F77_dgels F77_FUNC (dgels, DGELS)`
- `#define F77_dtrmm F77_FUNC (dtrmm, DTRMM)`
- `#define F77_dgemm F77_FUNC(dgemm,DGEMM)`
- `#define F77_dgemv F77_FUNC(dgemv,DGEMV)`

Functions

- void `F77_dgemm` (const char *ta, const char *tb, int *m, int *n, int *k, double *alpha, double *a, int *lda, double *b, int *ldb, double *beta, double *c, int *ldc)
- void `F77_dgemv` (const char *trans, int *m, int *n, double *alpha, double *a, int *lda, double *x, int *incx, double *beta, double *y, int *incy)
- void `F77_dpof2` (const char *uplo, int *n, double *a, int *lda, int *info)
- void `F77_dtptri` (const char *uplo, const char *diag, int *n, double *a, int *info)
- void `F77_dgels` (const char *trans, int *m, int *n, int *nhrs, double *A, int *lda, double *B, int *ldb, double *work, int *lwork, int *info)
- void `F77_dtrmm` (const char *lr, const char *uplo, const char *tr, const char *diag, int *n, int *m, double *alp, double *a, int *lda, double *b, int *ldb)
- int `accel_uinv` (double *_m, int n)
- double `accel_lsqr` (double **A, double *data, double *oparam, int ndata, int nparam, double **Ocvn)
- void `accel_multMatrixVec` (double *m1, double *v, int ndata, int npol, double *out)
- void `accel_multMatrix` (double *m1, double *m2, int ndata, int ndata2, int npol, double *out)

Variables

- char `useT2accel` =1

12.133.1 Macro Definition Documentation

12.133.1.1 `#define F77_dgels F77_FUNC (dgels, DGELS)`

12.133.1.2 `#define F77_dgemm F77_FUNC(dgemm,DGEMM)`

12.133.1.3 `#define F77_dgemv F77_FUNC(dgemv,DGEMV)`

12.133.1.4 `#define F77_dpotf2 F77_FUNC(dpotf2,DPOTF2)`

12.133.1.5 `#define F77_dtptri F77_FUNC(dtptri,DTPTRI)`

12.133.1.6 `#define F77_dtrmm F77_FUNC(dtrmm,DTRMM)`

12.133.2 Function Documentation

12.133.2.1 `double accel_lsqr (double ** A, double * data, double * oparam, int ndata, int nparam, double ** Ocvm)`

Do the least squares using QR decomposition

12.133.2.2 `void accel_multMatrix (double * m1, double * m2, int ndata, int ndata2, int npol, double * out)`

12.133.2.3 `void accel_multMatrixVec (double * m1, double * v, int ndata, int npol, double * out)`

12.133.2.4 `int accel_uinv (double * _m, int n)`

An accelerated cholesky decomposion to form uinv in plac. uinv is a lower triangular, row-major, matrix.

12.133.2.5 `void F77_dgels (const char * trans, int * m, int * n, int * nhrs, double * A, int * lda, double * B, int * ldb, double * work, int * lwork, int * info)`

12.133.2.6 `void F77_dgemm (const char * ta, const char * tb, int * m, int * n, int * k, double * alpha, double * a, int * lda, double * b, int * ldb, double * beta, double * c, int * ldc)`

12.133.2.7 `void F77_dgemv (const char * trans, int * m, int * n, double * alpha, double * a, int * lda, double * x, int * incx, double * beta, double * y, int * incy)`

12.133.2.8 `void F77_dpotf2 (const char * uplo, int * n, double * a, int * lda, int * info)`

12.133.2.9 `void F77_dtptri (const char * uplo, const char * diag, int * n, double * a, int * info)`

12.133.2.10 `void F77_dtrmm (const char * lr, const char * uplo, const char * tr, const char * diag, int * n, int * m, double * alp, double * a, int * lda, double * b, int * ldb)`

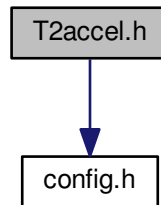
12.133.3 Variable Documentation

12.133.3.1 `char useT2accel =1`

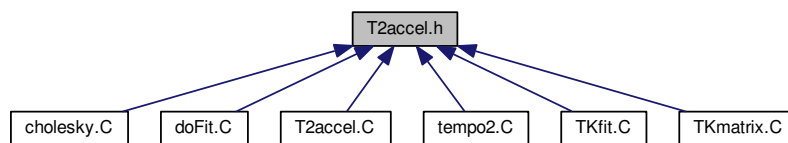
12.134 T2accel.h File Reference

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

Include dependency graph for T2accel.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define ACCEL_UINV`
- `#define ACCEL_LSQ`
- `#define ACCEL_MULTMATRIX`

Functions

- `int accel_uinv (double *_m, int n)`
- `double accel_lsq_qr (double **dm, double *data, double *oparm, int ndata, int nparam, double **Ocvrm)`
- `void accel_multMatrixVec (double *m1, double *v, int ndata, int npol, double *out)`
- `void accel_multMatrix (double *m1, double *m2, int ndata, int ndata2, int npol, double *out)`

Variables

- `char useT2accel`

12.134.1 Macro Definition Documentation

12.134.1.1 `#define ACCEL_LSQ`

12.134.1.2 `#define ACCEL_MULTMATRIX`

12.134.1.3 `#define ACCEL_UINV`

12.134.2 Function Documentation

12.134.2.1 `double accel_lsqr (double ** A, double * data, double * oparam, int ndata, int nparam, double ** Ocvm)`

Do the least squares using QR decomposition

12.134.2.2 `void accel_multMatrix (double * m1, double * m2, int ndata, int ndata2, int npol, double * out)`

12.134.2.3 `void accel_multMatrixVec (double * m1, double * v, int ndata, int npol, double * out)`

12.134.2.4 `int accel_uinv (double * _m, int n)`

An accelerated cholesky decomposition to form uinv in plac. uinv is a lower triangular, row-major, matrix.

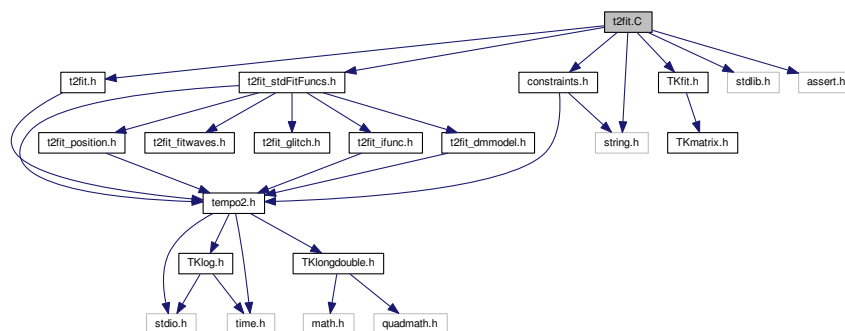
12.134.3 Variable Documentation

12.134.3.1 `char useT2accel`

12.135 t2fit.C File Reference

```
#include "t2fit.h"
#include "t2fit_stdFitFuncs.h"
#include "constraints.h"
#include <TKfit.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
```

Include dependency graph for t2fit.C:



Macros

- `#define T2_SVD_TOL 1e-27`

Functions

- void [t2Fit](#) ([pulsar](#) *psr, unsigned int npsr, const char *covarFuncFile)
- unsigned int [t2Fit_getFitData](#) ([pulsar](#) *psr, double *x, double *y, double *e, int *ip)
- void [t2Fit_buildDesignMatrix](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, double *afunc)
- void [t2Fit_buildConstraintsMatrix](#) ([pulsar](#) *psr, int ipsr, int iconstraint, double *afunc)
- void [t2Fit_updateParameters](#) ([pulsar](#) *psr, int ipsr, double *val, double *error)
- void [t2Fit_fillFitInfo_INNER](#) ([pulsar](#) *psr, [FitInfo](#) &OUT, const int globalflag)
- void [t2Fit_fillGlobalFitInfo](#) ([pulsar](#) *psr, unsigned int npsr, [FitInfo](#) &OUT)
- void [t2Fit_fillFitInfo](#) ([pulsar](#) *psr, [FitInfo](#) &OUT)

12.135.1 Macro Definition Documentation

12.135.1.1 `#define T2_SVD_TOL 1e-27`

12.135.2 Function Documentation

12.135.2.1 void [t2Fit](#) ([pulsar](#) * *psr*, unsigned int *npsr*, const char * *covarFuncFile*)

Find out if there are any global parameters and what they are...

However we are going to do the fit, we want to loop over all the pulsars to get the input data and design matrices etc.

Working out which data contributes to the fit is done in this routine. Basically gets values for all observations within START and FINISH which are not deleted.

returns the number of data points.

Now we work out which parameters are being fit for, how many parameters, and determine the gradient functions for the design matrix and the update functions which update the pulsar struct.

The whitening matrix behaves differently if we have a covariance matrix. If we have a covariance matrix, uinv is an ndata x ndata triangular matrix. Otherwise, it only has diagonal elements, so we efficiently store it as a 1-d ndata array.

Now we form the whitening matrix, uinv. Note that [getCholeskyMatrix\(\)](#) is clever enough to see that we have created a 1 x ndata matrix if we have only diagonal elements.

The design matrix is the matrix of gradients for the least-squares. If the design matrix is M, parameters p, and data d, we are solving $M.p = d$ It is ndata x nparams in size. We also allocate the whitened DM here.

The constraints matrix is similar to the design matrix, but here we are solving: $B.p = 0$ Where B is the constraints matrix and p is the parameters. we solve both this and the DM equation set simultaneously. TKleastSquares will do this for us.

If there are no constraints we leave it as NULL, which is detected in TKfit as no constraints anyway.

Now we multiply the design matrix and the data vector by the whitening matrix. If we just have variances (uinv is diagonal) then we do it traditionally, otherwise we use TKmultMatrix as this is usually backed by LAPACK and so is fast :)

12.135.2.2 void [t2Fit_buildConstraintsMatrix](#) ([pulsar](#) * *psr*, int *ipsr*, int *iconstraint*, double * *afunc*)

12.135.2.3 void [t2Fit_buildDesignMatrix](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, double * *afunc*)

12.135.2.4 void [t2Fit_fillFitInfo](#) ([pulsar](#) * *psr*, [FitInfo](#) & *OUT*)

12.135.2.5 void [t2Fit_fillFitInfo_INNER](#) ([pulsar](#) * *psr*, [FitInfo](#) & *OUT*, const int *globalflag*)

12.135.2.6 void t2Fit_fillGlobalFitInfo (pulsar * *psr*, unsigned int *npsr*, FitInfo & *OUT*)

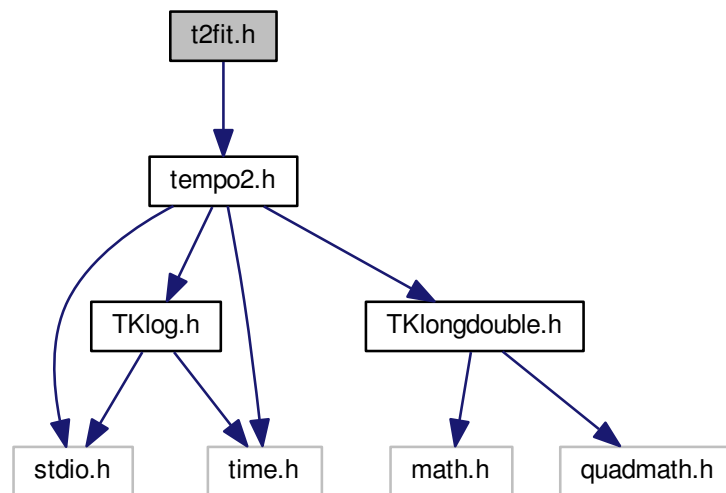
12.135.2.7 unsigned int t2Fit_getFitData (pulsar * *psr*, double * *x*, double * *y*, double * *e*, int * *ip*)

12.135.2.8 void t2Fit_updateParameters (pulsar * *psr*, int *ipsr*, double * *val*, double * *error*)

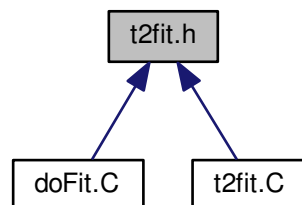
12.136 t2fit.h File Reference

```
#include <tempo2.h>
```

Include dependency graph for t2fit.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [t2Fit](#) (pulsar **psr*, unsigned int *npsr*, const char **covarFuncFile*)
- unsigned int [t2Fit_getFitData](#) (pulsar **psr*, double **x*, double **y*, double **e*, int **ip*)

- void `t2Fit_fillGlobalFitInfo` (`pulsar *psr`, unsigned int `npsr`, `FitInfo` &OUT)
- void `t2Fit_fillFitInfo` (`pulsar *psr`, `FitInfo` &OUT)
- void `t2Fit_buildDesignMatrix` (`pulsar *psr`, int `ipsr`, double `x`, int `ipos`, double `*afunc`)
- void `t2Fit_buildConstraintsMatrix` (`pulsar *psr`, int `ipsr`, int `iconstraint`, double `*afunc`)
- void `t2Fit_updateParameters` (`pulsar *psr`, int `ipsr`, double `*val`, double `*error`)

12.136.1 Function Documentation

12.136.1.1 void `t2Fit` (`pulsar *psr`, unsigned int `npsr`, const char * `covarFuncFile`)

Find out if there are any global parameters and what they are...

However we are going to do the fit, we want to loop over all the pulsars to get the input data and design matrices etc.

Working out which data contributes to the fit is done in this routine. Basically gets values for all observations within START and FINISH which are not deleted.

returns the number of data points.

Now we work out which parameters are being fit for, how many parameters, and determine the gradient functions for the design matrix and the update functions which update the pulsar struct.

The whitening matrix behaves differently if we have a covariance matrix. If we have a covariance matrix, `uinv` is an `ndata x ndata` triangular matrix. Otherwise, it only has diagonal elements, so we efficiently store it as a 1-d `ndata` array.

Now we form the whitening matrix, `uinv`. Note that `getCholeskyMatrix()` is clever enough to see that we have created a 1 x `ndata` matrix if we have only diagonal elements.

The design matrix is the matrix of gradients for the least-squares. If the design matrix is `M`, parameters `p`, and data `d`, we are solving $M.p = d$ It is `ndata x nparams` in size. We also allocate the whitened DM here.

The constraints matrix is similar to the design matrix, but here we are solving: $B.p = 0$ Where `B` is the constraints matrix and `p` is the parameters. we solve both this and the DM equation set simultaneously. `TKleastSquares` will do this for us.

If there are no constraints we leave it as `NULL`, which is detected in `TKfit` as no constraints anyway.

Now we multiply the design matrix and the data vector by the whitening matrix. If we just have variances (`uinv` is diagonal) then we do it traditionally, otherwise we use `TKmultMatrix` as this is usually backed by LAPACK and so is fast :)

12.136.1.2 void `t2Fit_buildConstraintsMatrix` (`pulsar *psr`, int `ipsr`, int `iconstraint`, double * `afunc`)

12.136.1.3 void `t2Fit_buildDesignMatrix` (`pulsar *psr`, int `ipsr`, double `x`, int `ipos`, double * `afunc`)

12.136.1.4 void `t2Fit_fillFitInfo` (`pulsar *psr`, `FitInfo` & `OUT`)

12.136.1.5 void `t2Fit_fillGlobalFitInfo` (`pulsar *psr`, unsigned int `npsr`, `FitInfo` & `OUT`)

12.136.1.6 unsigned int `t2Fit_getFitData` (`pulsar *psr`, double * `x`, double * `y`, double * `e`, int * `ip`)

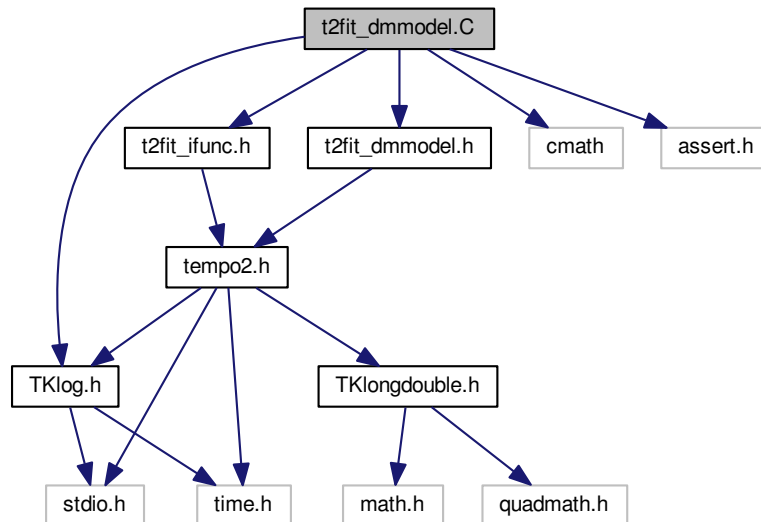
12.136.1.7 void `t2Fit_updateParameters` (`pulsar *psr`, int `ipsr`, double * `val`, double * `error`)

12.137 t2fit_dmmodel.C File Reference

```
#include "t2fit_dmmodel.h"
```



```
#include "t2fit_ifunc.h"
#include "TKlog.h"
#include <cmath>
#include <assert.h>
Include dependency graph for t2fit_dmmodel.C:
```



Functions

- double [t2FitFunc_dmmodelDM](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_dmmodelDM](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)
- double [t2FitFunc_dmmodelCM](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_dmmodelCM](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)

12.137.1 Function Documentation

12.137.1.1 double [t2FitFunc_dmmodelCM](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)

12.137.1.2 double [t2FitFunc_dmmodelDM](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)

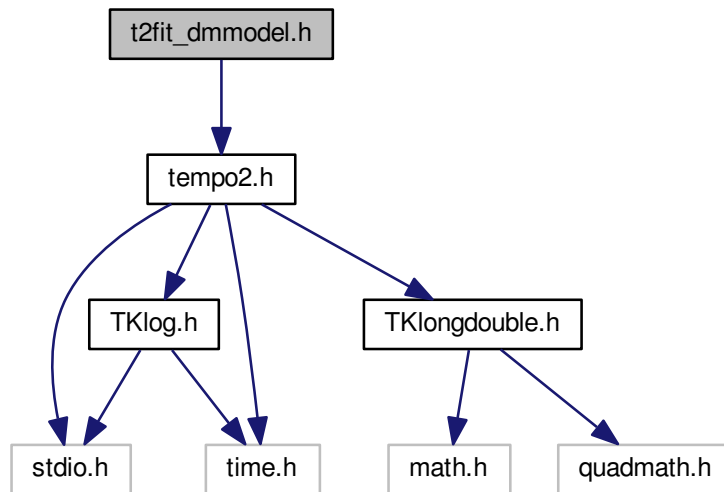
12.137.1.3 void [t2UpdateFunc_dmmodelCM](#) ([pulsar](#) * *psr*, int *ipsr*, [param_label](#) *label*, int *k*, double *val*, double *err*)

12.137.1.4 void [t2UpdateFunc_dmmodelDM](#) ([pulsar](#) * *psr*, int *ipsr*, [param_label](#) *label*, int *k*, double *val*, double *err*)

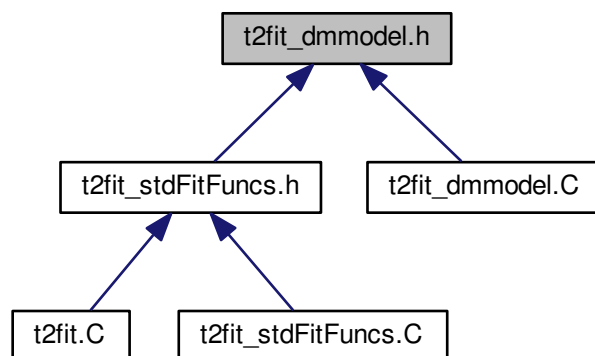
12.138 t2fit_dmmodel.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for t2fit_dmmodel.h:



This graph shows which files directly or indirectly include this file:



Functions

- double `t2FitFunc_dmmodelDM` (`pulsar` *psr, int ipsr, double x, int ipos, `param_label` label, int k)
- void `t2UpdateFunc_dmmodelDM` (`pulsar` *psr, int ipsr, `param_label` label, int k, double val, double err)
- double `t2FitFunc_dmmodelCM` (`pulsar` *psr, int ipsr, double x, int ipos, `param_label` label, int k)
- void `t2UpdateFunc_dmmodelCM` (`pulsar` *psr, int ipsr, `param_label` label, int k, double val, double err)

12.138.1 Function Documentation

12.138.1.1 `double t2FitFunc_dmmodelCM (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.138.1.2 `double t2FitFunc_dmmodelDM (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.138.1.3 `void t2UpdateFunc_dmmodelCM (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

12.138.1.4 `void t2UpdateFunc_dmmodelDM (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

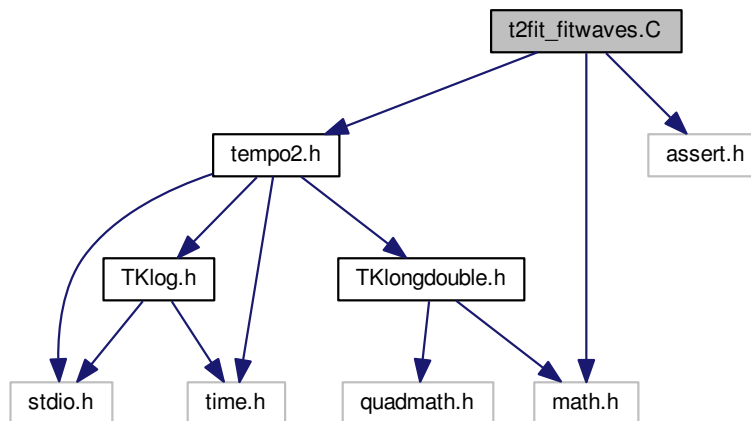
12.139 t2fit_fitwaves.C File Reference

```
#include <tempo2.h>
```

```
#include <math.h>
```

```
#include <assert.h>
```

Include dependency graph for t2fit_fitwaves.C:



Functions

- `double t2FitFunc_fitwaves (pulsar * psr, int ipsr, double x, int ipos, param_label label, int kk)`
- `void t2UpdateFunc_fitwaves (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`

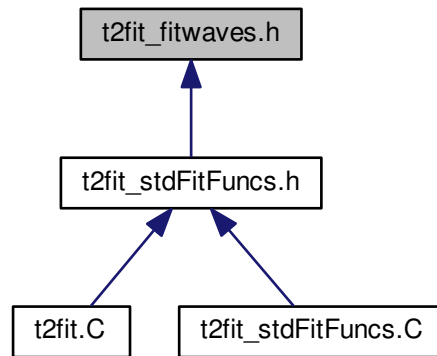
12.139.1 Function Documentation

12.139.1.1 `double t2FitFunc_fitwaves (pulsar * psr, int ipsr, double x, int ipos, param_label label, int kk)`

12.139.1.2 `void t2UpdateFunc_fitwaves (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`

12.140 t2fit_fitwaves.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- double `t2FitFunc_fitwaves` (`pulsar *psr`, int `ipsr`, double `x`, int `ipos`, `param_label label`, int `k`)
- void `t2UpdateFunc_fitwaves` (`pulsar *psr`, int `ipsr`, `param_label label`, int `k`, double `val`, double `err`)

12.140.1 Function Documentation

12.140.1.1 double `t2FitFunc_fitwaves` (`pulsar *psr`, int `ipsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.140.1.2 void `t2UpdateFunc_fitwaves` (`pulsar *psr`, int `ipsr`, `param_label label`, int `k`, double `val`, double `err`)

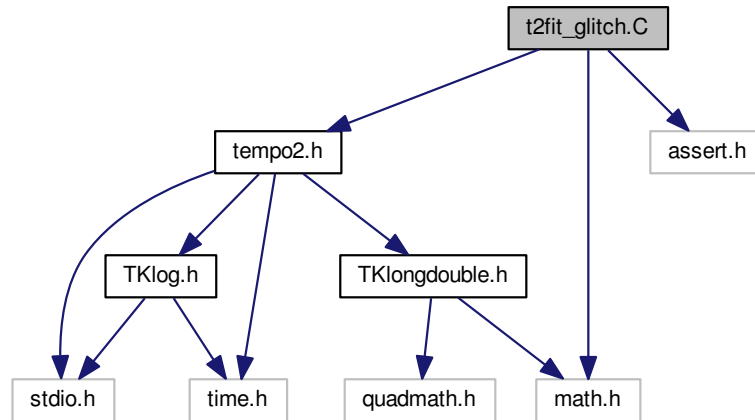
12.141 t2fit_glitch.C File Reference

```

#include <tempo2.h>
#include <math.h>
#include <assert.h>

```

Include dependency graph for t2fit_glitch.C:



Functions

- double [t2FitFunc_stdGlitch](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_stdGlitch](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double error)

12.141.1 Function Documentation

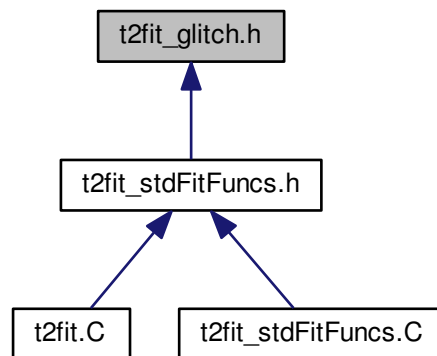
12.141.1.1 double [t2FitFunc_stdGlitch](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)

```
psr[ipsr].param[param_f].val[0];
```

12.141.1.2 void [t2UpdateFunc_stdGlitch](#) ([pulsar](#) * *psr*, int *ipsr*, [param_label](#) *label*, int *k*, double *val*, double *error*)

12.142 t2fit_glitch.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- double [t2FitFunc_stdGlitch](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_stdGlitch](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)

12.142.1 Function Documentation

12.142.1.1 double [t2FitFunc_stdGlitch](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)

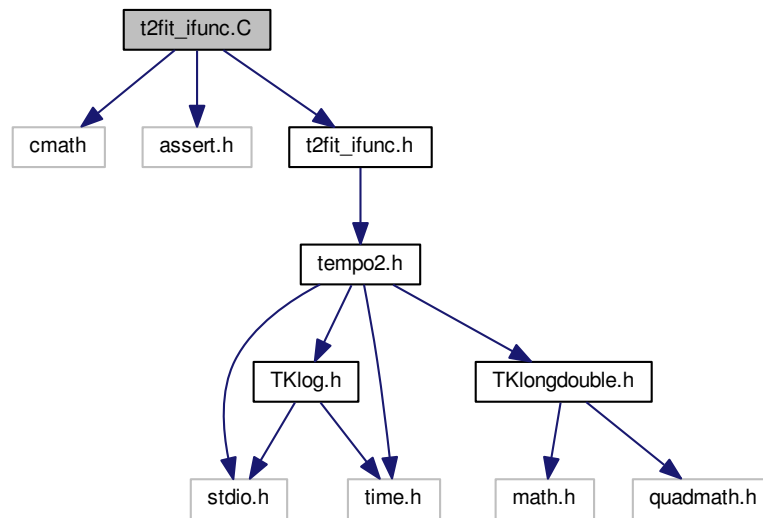
```
psr[ipsr].param[param_f].val[0];
```

12.142.1.2 void [t2UpdateFunc_stdGlitch](#) ([pulsar](#) * *psr*, int *ipsr*, [param_label](#) *label*, int *k*, double *val*, double *err*)

12.143 t2fit_ifunc.C File Reference

```
#include <cmath>
#include <assert.h>
#include "t2fit_ifunc.h"
```

Include dependency graph for t2fit_ifunc.C:



Functions

- double [ifunc](#) (const double *mjd, const double t, const int N, const int k)
- double [sifunc](#) (const double *T, const double bat, const int k)
- double [t2FitFunc_sifunc](#) (pulsar *psr, int ipsr, double x, int ipos, [param_label label](#), int k)
- double [t2FitFunc_ifunc](#) (pulsar *psr, int ipsr, double x, int ipos, [param_label label](#), int k)
- void [t2UpdateFunc_ifunc](#) (pulsar *psr, int ipsr, [param_label label](#), int k, double val, double err)

12.143.1 Function Documentation

12.143.1.1 double ifunc (const double * *mjd*, const double *t*, const int *N*, const int *k*)

12.143.1.2 double sifunc (const double * *T*, const double *bat*, const int *k*)

12.143.1.3 double t2FitFunc_ifunc (pulsar * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label label](#), int *k*)

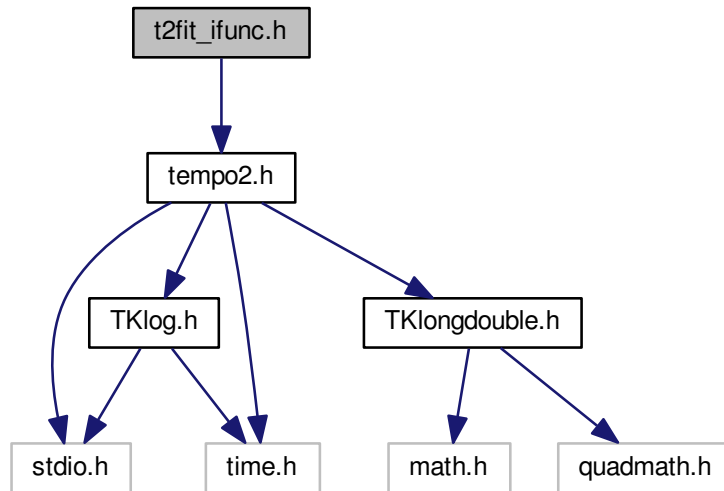
12.143.1.4 double t2FitFunc_sifunc (pulsar * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label label](#), int *k*)

12.143.1.5 void t2UpdateFunc_ifunc (pulsar * *psr*, int *ipsr*, [param_label label](#), int *k*, double *val*, double *err*)

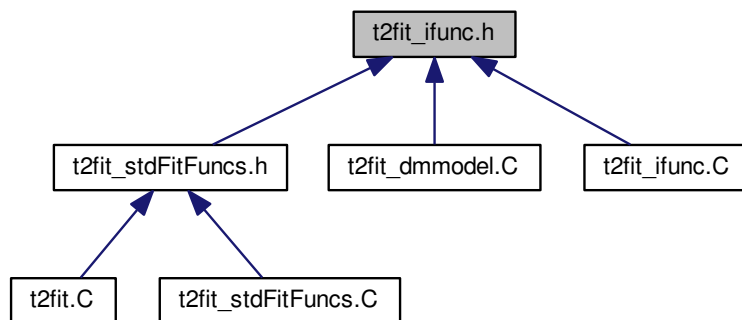
12.144 t2fit_ifunc.h File Reference

```
#include "tempo2.h"
```

Include dependency graph for t2fit_ifunc.h:



This graph shows which files directly or indirectly include this file:



Functions

- double [ifunc](#) (const double *mjd, const double t, const int N, const int k)
- double [sifunc](#) (const double *T, const double t, const int k)
- double [t2FitFunc_sifunc](#) (pulsar *psr, int ipsr, double x, int ipos, [param_label label](#), int k)
- double [t2FitFunc_ifunc](#) (pulsar *psr, int ipsr, double x, int ipos, [param_label label](#), int k)
- void [t2UpdateFunc_ifunc](#) (pulsar *psr, int ipsr, [param_label label](#), int k, double val, double err)

12.144.1 Function Documentation

12.144.1.1 `double ifunc (const double * mjd, const double t, const int N, const int k)`

12.144.1.2 `double sinfunc (const double * T, const double t, const int k)`

12.144.1.3 `double t2FitFunc_ifunc (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.144.1.4 `double t2FitFunc_sifunc (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.144.1.5 `void t2UpdateFunc_ifunc (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

12.145 t2fit_position.C File Reference

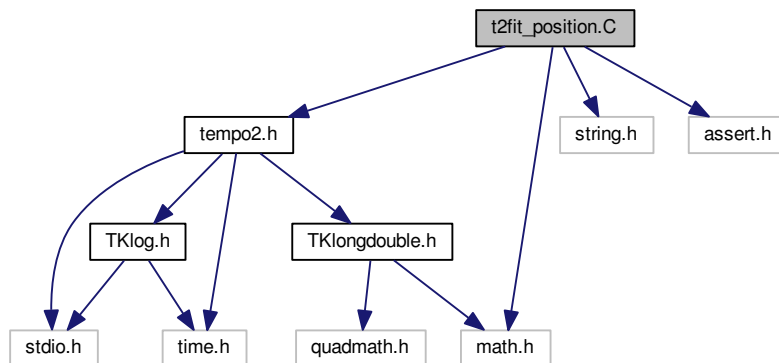
```
#include <tempo2.h>
```

```
#include <math.h>
```

```
#include <string.h>
```

```
#include <assert.h>
```

Include dependency graph for t2fit_position.C:



Functions

- `double t2FitFunc_stdPosition (pulsar * psr, int ipsr, double x, int ipos, param_label i, int k)`
- `void t2UpdateFunc_stdPosition (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

12.145.1 Function Documentation

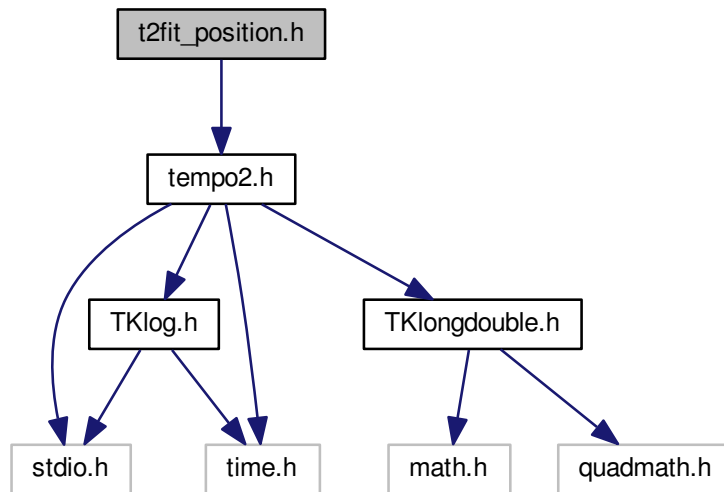
12.145.1.1 `double t2FitFunc_stdPosition (pulsar * psr, int ipsr, double x, int ipos, param_label i, int k)`

12.145.1.2 `void t2UpdateFunc_stdPosition (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

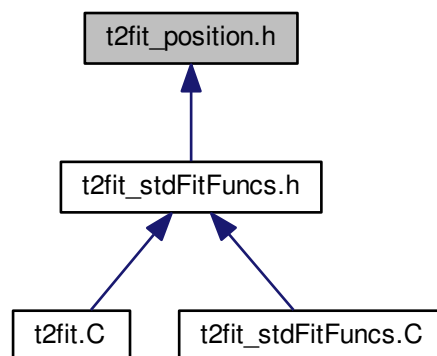
12.146 t2fit_position.h File Reference

```
#include <tempo2.h>
```

Include dependency graph for t2fit_position.h:



This graph shows which files directly or indirectly include this file:



Functions

- double `t2FitFunc_stdPosition` (`pulsar` *psr, int ipsr, double x, int ipos, `param_label` label, int k)
- void `t2UpdateFunc_stdPosition` (`pulsar` *psr, int ipsr, `param_label` label, int k, double val, double err)

12.146.1 Function Documentation

12.146.1.1 double `t2FitFunc_stdPosition` (`pulsar` * psr, int ipsr, double x, int ipos, `param_label` label, int k)

12.146.1.2 void t2UpdateFunc_stdPosition (pulsar *psr, int ipsr, param_label label, int k, double val, double err)

12.147 t2fit_stdFitFuncs.C File Reference

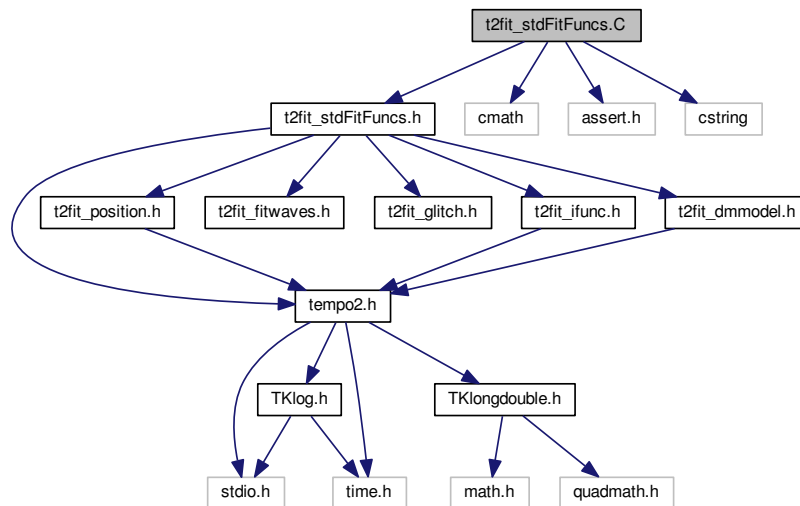
```
#include "t2fit_stdFitFuncs.h"
```

```
#include <cmath>
```

```
#include <assert.h>
```

```
#include <cstring>
```

Include dependency graph for t2fit_stdFitFuncs.C:



Functions

- double t2FitFunc_zero (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_zero (pulsar *psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_stdFreq (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdFreq (pulsar *psr, int ipsr, param_label label, int k, double val, double error)
- double t2FitFunc_binaryModels (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_binaryModels (pulsar *psr, int ipsr, param_label label, int k, double val, double error)
- double t2FitFunc_planet (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_planet (pulsar *psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_stdDm (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_simpleAdd (pulsar *psr, int ipsr, param_label label, int k, double val, double error)
- void t2UpdateFunc_simpleMinus (pulsar *psr, int ipsr, param_label label, int k, double val, double error)
- double t2FitFunc_stdGravWav (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_stdGravWav (pulsar *psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_telPos (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_telPos (pulsar *psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_miscDm (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_miscDm (pulsar *psr, int ipsr, param_label label, int k, double val, double err)
- double t2FitFunc_jump (pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)
- void t2UpdateFunc_jump (pulsar *psr, int ipsr, param_label label, int k, double val, double err)

12.147.1 Function Documentation

12.147.1.1 `double t2FitFunc_binaryModels (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

Binary models - need to select on `psr[ipsr].binaryModel`

12.147.1.2 `double t2FitFunc_jump (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.147.1.3 `double t2FitFunc_miscDm (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.147.1.4 `double t2FitFunc_planet (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.147.1.5 `double t2FitFunc_stdDm (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.147.1.6 `double t2FitFunc_stdFreq (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

The pulse frequency, and derivatives

12.147.1.7 `double t2FitFunc_stdGravWav (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.147.1.8 `double t2FitFunc_telPos (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

12.147.1.9 `double t2FitFunc_zero (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`

The zero offset.

12.147.1.10 `void t2UpdateFunc_binaryModels (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`

12.147.1.11 `void t2UpdateFunc_jump (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

12.147.1.12 `void t2UpdateFunc_miscDm (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

12.147.1.13 `void t2UpdateFunc_planet (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

12.147.1.14 `void t2UpdateFunc_simpleAdd (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`

12.147.1.15 `void t2UpdateFunc_simpleMinus (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`

12.147.1.16 `void t2UpdateFunc_stdFreq (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`

12.147.1.17 `void t2UpdateFunc_stdGravWav (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

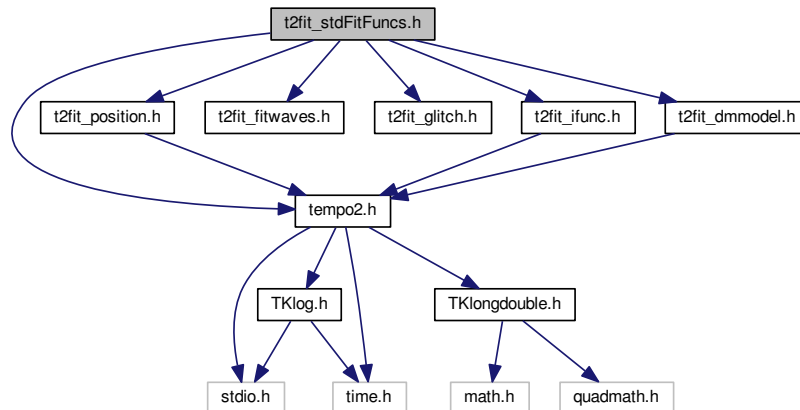
12.147.1.18 `void t2UpdateFunc_telPos (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

12.147.1.19 `void t2UpdateFunc_zero (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

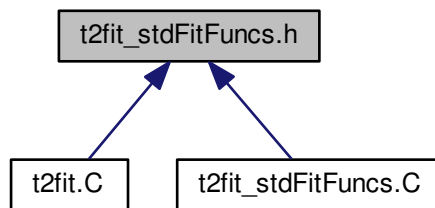
12.148 t2fit_stdFitFuncs.h File Reference

```
#include <tempo2.h>
#include "t2fit_position.h"
#include "t2fit_fitwaves.h"
#include "t2fit_glitch.h"
#include "t2fit_ifunc.h"
#include "t2fit_dmmodel.h"
```

Include dependency graph for t2fit_stdFitFuncs.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [t2UpdateFunc_simpleAdd](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double error)
- void [t2UpdateFunc_simpleMinus](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double error)
- double [t2FitFunc_zero](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_zero](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)
- double [t2FitFunc_stdFreq](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_stdFreq](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)
- double [t2FitFunc_binaryModels](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_binaryModels](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)
- double [t2FitFunc_planet](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_planet](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)
- double [t2FitFunc_stdDm](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- double [t2FitFunc_stdGravWav](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_stdGravWav](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)
- double [t2FitFunc_telPos](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_telPos](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)
- double [t2FitFunc_ifunc](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)

- void `t2UpdateFunc_ifunc` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)
- double `t2FitFunc_miscDm` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)
- void `t2UpdateFunc_miscDm` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)
- double `t2FitFunc_jump` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)
- void `t2UpdateFunc_jump` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1 Function Documentation

12.148.1.1 double `t2FitFunc_binaryModels` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

Binary models - need to select on `psr[iprsr].binaryModel`

12.148.1.2 double `t2FitFunc_ifunc` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.148.1.3 double `t2FitFunc_jump` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.148.1.4 double `t2FitFunc_miscDm` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.148.1.5 double `t2FitFunc_planet` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.148.1.6 double `t2FitFunc_stdDm` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.148.1.7 double `t2FitFunc_stdFreq` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

The pulse frequency, and derivatives

12.148.1.8 double `t2FitFunc_stdGravWav` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.148.1.9 double `t2FitFunc_telPos` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

12.148.1.10 double `t2FitFunc_zero` (`pulsar *psr`, int `iprsr`, double `x`, int `ipos`, `param_label label`, int `k`)

The zero offset.

12.148.1.11 void `t2UpdateFunc_binaryModels` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1.12 void `t2UpdateFunc_ifunc` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1.13 void `t2UpdateFunc_jump` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1.14 void `t2UpdateFunc_miscDm` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1.15 void `t2UpdateFunc_planet` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1.16 void `t2UpdateFunc_simpleAdd` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `error`)

12.148.1.17 void `t2UpdateFunc_simpleMinus` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `error`)

12.148.1.18 void `t2UpdateFunc_stdFreq` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1.19 void `t2UpdateFunc_stdGravWav` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

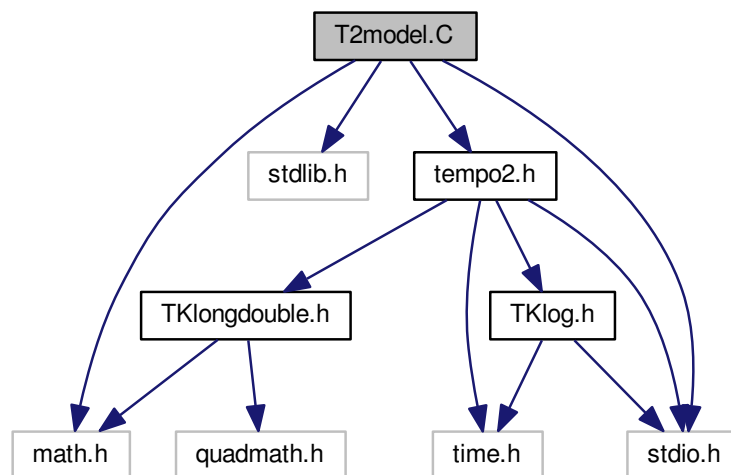
12.148.1.20 void `t2UpdateFunc_telPos` (`pulsar *psr`, int `iprsr`, `param_label label`, int `k`, double `val`, double `err`)

12.148.1.21 void t2UpdateFunc_zero (pulsar *psr, int ipsr, param_label label, int k, double val, double err)

12.149 T2model.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for T2model.C:



Functions

- [longdouble getParameter](#) (pulsar *psr, int p, int k)
- void [calcGR](#) (double mtot, double m2, double x, double ecc, double an, double afac, double f0, double *dr, double *dth, double *er, double *eth, double *xk, double *si, double *gamma, double *pbdot, double *a0, double *b0)
- void [getKeplerian](#) (pulsar *psr, int com, double *pb, [longdouble](#) *t0, double *ecc, double *omz, double *x, double *eps1, double *eps2, [longdouble](#) *t0asc, double *shapmax, double *kom, double *kin)
- void [addKeplerianJumps](#) (pulsar *psr, int ipos, double *torb, double *x, double *ecc, double *omz, double *pb)
- void [getPostKeplerian](#) (pulsar *psr, int com, double an, double *si, double *m2, double *mtot, double *omdot, double *gamma, double *xdot, double *xpbdot, double *pbdot, double *edot, double *pmra, double *pmdec, double *dpara, double *dr, double *dth, double *a0, double *b0, double *xomdot, double *afac, double *eps1dot, double *eps2dot, double *daop)
- void [updateParameters](#) (double edot, double xdot, double eps1dot, double eps2dot, [longdouble](#) tt0, double *ecc, double *x, double *eps1, double *eps2)
- void [deriveKeplerian](#) (double pb, double kom, double *an, double *sin_omega, double *cos_omega)
- void [derivePostKeplerian](#) (double mtot, double m2, double dr, double dth, double ecc, double *m1, double *er, double *eth)
- void [KopeikinTerms](#) (pulsar *psr, int ipos, double ki, double pmra, double sin_omega, double pmdec, double cos_omega, [longdouble](#) tt0, double dpara, double daop, double si, double *x, [longdouble](#) *DK011, [longdouble](#) *DK012, [longdouble](#) *DK021, [longdouble](#) *DK022, [longdouble](#) *DK031, [longdouble](#) *DK032, [longdouble](#) *DK041, [longdouble](#) *DK042, [longdouble](#) *DK013, [longdouble](#) *DK014, [longdouble](#) *DK023, [longdouble](#) *DK024, [longdouble](#) *DK033, [longdouble](#) *DK034, [longdouble](#) *DK043, [longdouble](#) *DK044)

- void `computeU` (double *phase*, double *ecc*, double **u*)
- double `T2model` (*pulsar* **psr*, int *p*, int *ipos*, int *param*, int *arr*)
- void `updateT2` (*pulsar* **psr*, double *val*, double *err*, int *pos*, int *arr*)

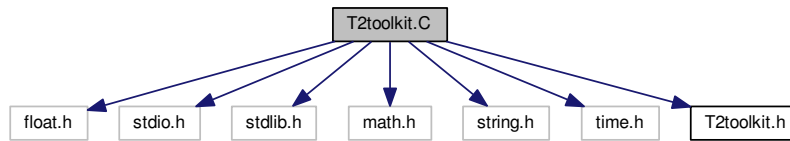
12.149.1 Function Documentation

- 12.149.1.1 void `addKeplerianJumps` (*pulsar* **psr*, int *ipos*, double **torb*, double **x*, double **ecc*, double **omz*, double **pb*)
- 12.149.1.2 void `calcGR` (double *mtot*, double *m2*, double *x*, double *ecc*, double *an*, double *afac*, double *f0*, double **dr*, double **dth*, double **er*, double **eth*, double **xk*, double **si*, double **gamma*, double **pbdot*, double **a0*, double **b0*)
- 12.149.1.3 void `computeU` (double *phase*, double *ecc*, double **u*)
- 12.149.1.4 void `deriveKeplerian` (double *pb*, double *kom*, double **an*, double **sin_omega*, double **cos_omega*)
- 12.149.1.5 void `derivePostKeplerian` (double *mtot*, double *m2*, double *dr*, double *dth*, double *ecc*, double **m1*, double **er*, double **eth*)
- 12.149.1.6 void `getKeplerian` (*pulsar* **psr*, int *com*, double **pb*, longdouble **t0*, double **ecc*, double **omz*, double **x*, double **eps1*, double **eps2*, longdouble **t0asc*, double **shapmax*, double **kom*, double **kin*)
- 12.149.1.7 longdouble `getParameter` (*pulsar* **psr*, int *p*, int *k*)
- 12.149.1.8 void `getPostKeplerian` (*pulsar* **psr*, int *com*, double *an*, double **si*, double **m2*, double **mtot*, double **omdot*, double **gamma*, double **xdot*, double **xpbdot*, double **pbdot*, double **edot*, double **pmra*, double **pmdec*, double **dpara*, double **dr*, double **dth*, double **a0*, double **b0*, double **xomdot*, double **afac*, double **eps1dot*, double **eps2dot*, double **daop*)
- 12.149.1.9 void `KopeikinTerms` (*pulsar* **psr*, int *ipos*, double *ki*, double *pmra*, double *sin_omega*, double *pmdec*, double *cos_omega*, longdouble *tt0*, double *dpara*, double *daop*, double *si*, double **x*, longdouble **DK011*, longdouble **DK012*, longdouble **DK021*, longdouble **DK022*, longdouble **DK031*, longdouble **DK032*, longdouble **DK041*, longdouble **DK042*, longdouble **DK013*, longdouble **DK014*, longdouble **DK023*, longdouble **DK024*, longdouble **DK033*, longdouble **DK034*, longdouble **DK043*, longdouble **DK044*)
- 12.149.1.10 double `T2model` (*pulsar* **psr*, int *p*, int *ipos*, int *param*, int *arr*)
- 12.149.1.11 void `updateParameters` (double *edot*, double *xdot*, double *eps1dot*, double *eps2dot*, longdouble *tt0*, double **ecc*, double **x*, double **eps1*, double **eps2*)
- 12.149.1.12 void `updateT2` (*pulsar* **psr*, double *val*, double *err*, int *pos*, int *arr*)

12.150 T2toolkit.C File Reference

```
#include <float.h>
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <time.h>
#include "T2toolkit.h"
```


Include dependency graph for T2toolkit.C:



Macros

- `#define RAND_N 624`
- `#define RAND_M 397`

Functions

- void `TKconvertFloat1` (double *x, float *ox, int n)
- void `TKconvertFloat2` (double *x, double *y, float *ox, float *oy, int n)
- float `TKfindMin_f` (float *x, int n)
- double `TKfindMin_d` (double *x, int n)
- double `TKsign_d` (double a, double b)
- float `TKfindMedian_f` (float *val, int count)
- double `TKfindMedian_d` (double *val, int count)
- float `TKfindRMS_f` (float *x, int n)
- double `TKfindRMS_d` (double *x, int n)
- float `TKfindRMSweight_d` (double *x, double *e, int n)
- float `TKfindMax_f` (float *x, int n)
- double `TKretMax_d` (double a, double b)
- double `TKretMin_d` (double a, double b)
- float `TKretMax_f` (float a, float b)
- float `TKretMin_f` (float a, float b)
- int `TKretMin_i` (int a, int b)
- double `TKfindMax_d` (double *x, int n)
- double `TKmean_d` (double *x, int n)
- float `TKmean_f` (float *x, int n)
- double `TKvariance_d` (double *x, int n)
- double `TKrange_d` (double *x, int n)
- float `TKrange_f` (float *x, int n)
- void `TKsort_f` (float *val, int nob)
- void `TKsort_d` (double *val, int nob)
- void `TKsort_2f` (float *val, float *val2, int nob)
- void `TKsort_3d` (double *val, double *val2, double *val3, int nob)
- void `TKzeromean_d` (int n, double *y)
- double `TKranDev` (long *seed)
- double `TKgaussDev` (long *seed)
- long `TKsetSeed` ()
- void `init_genrand` (unsigned long s)
- unsigned long `genrand_int32` (void)
- double `genrand_real1` (void)

12.150.1 Macro Definition Documentation

12.150.1.1 `#define RAND_M 397`

12.150.1.2 `#define RAND_N 624`

12.150.2 Function Documentation

12.150.2.1 `unsigned long genrand_int32 (void)`

12.150.2.2 `double genrand_real1 (void)`

12.150.2.3 `void init_genrand (unsigned long s)`

12.150.2.4 `void TKconvertFloat1 (double * x, float * ox, int n)`

12.150.2.5 `void TKconvertFloat2 (double * x, double * y, float * ox, float * oy, int n)`

12.150.2.6 `double TKfindMax_d (double * x, int n)`

12.150.2.7 `float TKfindMax_f (float * x, int n)`

12.150.2.8 `double TKfindMedian_d (double * val, int count)`

12.150.2.9 `float TKfindMedian_f (float * val, int count)`

12.150.2.10 `double TKfindMin_d (double * x, int n)`

12.150.2.11 `float TKfindMin_f (float * x, int n)`

12.150.2.12 `double TKfindRMS_d (double * x, int n)`

12.150.2.13 `float TKfindRMS_f (float * x, int n)`

12.150.2.14 `float TKfindRMSweight_d (double * x, double * e, int n)`

12.150.2.15 `double TKgaussDev (long * seed)`

12.150.2.16 `double TKmean_d (double * x, int n)`

12.150.2.17 `float TKmean_f (float * x, int n)`

12.150.2.18 `double TKranDev (long * seed)`

12.150.2.19 `double TKrange_d (double * x, int n)`

12.150.2.20 `float TKrange_f (float * x, int n)`

12.150.2.21 `double TKretMax_d (double a, double b)`

12.150.2.22 `float TKretMax_f (float a, float b)`

12.150.2.23 `double TKretMin_d (double a, double b)`

12.150.2.24 `float TKretMin_f (float a, float b)`

- 12.150.2.25 `int TKretMin_i (int a, int b)`
- 12.150.2.26 `long TKsetSeed ()`
- 12.150.2.27 `double TKsign_d (double a, double b)`
- 12.150.2.28 `void TKsort_2f (float * val, float * val2, int nobs)`
- 12.150.2.29 `void TKsort_3d (double * val, double * val2, double * val3, int nobs)`
- 12.150.2.30 `void TKsort_d (double * val, int nobs)`
- 12.150.2.31 `void TKsort_f (float * val, int nobs)`
- 12.150.2.32 `double TKvariance_d (double * x, int n)`
- 12.150.2.33 `void TKzeromean_d (int n, double * y)`

12.151 T2toolkit.h File Reference

Set of routines that are commonly used in tempo2 and/or its plugins.

This graph shows which files directly or indirectly include this file:



Functions

- void [TKconvertFloat1](#) (double **x*, float **ox*, int *n*)
- void [TKconvertFloat2](#) (double **x*, double **y*, float **ox*, float **oy*, int *n*)
- float [TKfindMin_f](#) (float **x*, int *n*)
- float [TKfindMedian_f](#) (float **val*, int *count*)
- double [TKfindMedian_d](#) (double **val*, int *count*)
- float [TKfindRMS_f](#) (float **x*, int *n*)
- double [TKfindRMS_d](#) (double **x*, int *n*)
- float [TKfindRMSweight_d](#) (double **x*, double **e*, int *n*)
- float [TKfindMax_f](#) (float **x*, int *n*)
- float [TKmean_f](#) (float **x*, int *n*)
- double [TKmean_d](#) (double **x*, int *n*)
- double [TKvariance_d](#) (double **x*, int *n*)
- double [TKrange_d](#) (double **x*, int *n*)
- float [TKrange_f](#) (float **x*, int *n*)
- double [TKfindMin_d](#) (double **x*, int *n*)
- double [TKfindMax_d](#) (double **x*, int *n*)
- double [TKsign_d](#) (double *a*, double *b*)
- double [TKretMax_d](#) (double *a*, double *b*)
- double [TKretMin_d](#) (double *a*, double *b*)
- float [TKretMax_f](#) (float *a*, float *b*)
- float [TKretMin_f](#) (float *a*, float *b*)
- int [TKretMin_i](#) (int *a*, int *b*)
- void [TKsort_f](#) (float **val*, int *nobs*)
- void [TKsort_d](#) (double **val*, int *nobs*)
- void [TKsort_2f](#) (float **val*, float **val2*, int *nobs*)
- void [TKsort_3d](#) (double **val*, double **val2*, double **val3*, int *nobs*)

- void [TKzeromean_d](#) (int *n*, double **y*)
- double [TKranDev](#) (long **seed*)
- double [TKgaussDev](#) (long **seed*)
- long [TKsetSeed](#) ()
- void [init_genrand](#) (unsigned long *s*)
- unsigned long [genrand_int32](#) (void)
- double [genrand_real1](#) (void)

12.151.1 Detailed Description

Set of routines that are commonly used in tempo2 and/or its plugins.

These routines are mainly stand-alone functions and exist for float and double precision variables

G. Hobbs: v2, 31 Dec 2008. Complete rewrite of the routines

NOTES: Related toolkits include: [TKspectrum.h](#): contains routines for spectral estimation [TKfit.h](#): contains routines for fitting

12.151.2 Function Documentation

- 12.151.2.1 unsigned long [genrand_int32](#) (void)
- 12.151.2.2 double [genrand_real1](#) (void)
- 12.151.2.3 void [init_genrand](#) (unsigned long *s*)
- 12.151.2.4 void [TKconvertFloat1](#) (double * *x*, float * *ox*, int *n*)
- 12.151.2.5 void [TKconvertFloat2](#) (double * *x*, double * *y*, float * *ox*, float * *oy*, int *n*)
- 12.151.2.6 double [TKfindMax_d](#) (double * *x*, int *n*)
- 12.151.2.7 float [TKfindMax_f](#) (float * *x*, int *n*)
- 12.151.2.8 double [TKfindMedian_d](#) (double * *val*, int *count*)
- 12.151.2.9 float [TKfindMedian_f](#) (float * *val*, int *count*)
- 12.151.2.10 double [TKfindMin_d](#) (double * *x*, int *n*)
- 12.151.2.11 float [TKfindMin_f](#) (float * *x*, int *n*)
- 12.151.2.12 double [TKfindRMS_d](#) (double * *x*, int *n*)
- 12.151.2.13 float [TKfindRMS_f](#) (float * *x*, int *n*)
- 12.151.2.14 float [TKfindRMSweight_d](#) (double * *x*, double * *e*, int *n*)
- 12.151.2.15 double [TKgaussDev](#) (long * *seed*)
- 12.151.2.16 double [TKmean_d](#) (double * *x*, int *n*)
- 12.151.2.17 float [TKmean_f](#) (float * *x*, int *n*)
- 12.151.2.18 double [TKranDev](#) (long * *seed*)

12.151.2.19 double TKrange_d (double * *x*, int *n*)

12.151.2.20 float TKrange_f (float * *x*, int *n*)

12.151.2.21 double TKretMax_d (double *a*, double *b*)

12.151.2.22 float TKretMax_f (float *a*, float *b*)

12.151.2.23 double TKretMin_d (double *a*, double *b*)

12.151.2.24 float TKretMin_f (float *a*, float *b*)

12.151.2.25 int TKretMin_i (int *a*, int *b*)

12.151.2.26 long TKsetSeed ()

12.151.2.27 double TKsign_d (double *a*, double *b*)

12.151.2.28 void TKsort_2f (float * *val*, float * *val2*, int *nobs*)

12.151.2.29 void TKsort_3d (double * *val*, double * *val2*, double * *val3*, int *nobs*)

12.151.2.30 void TKsort_d (double * *val*, int *nobs*)

12.151.2.31 void TKsort_f (float * *val*, int *nobs*)

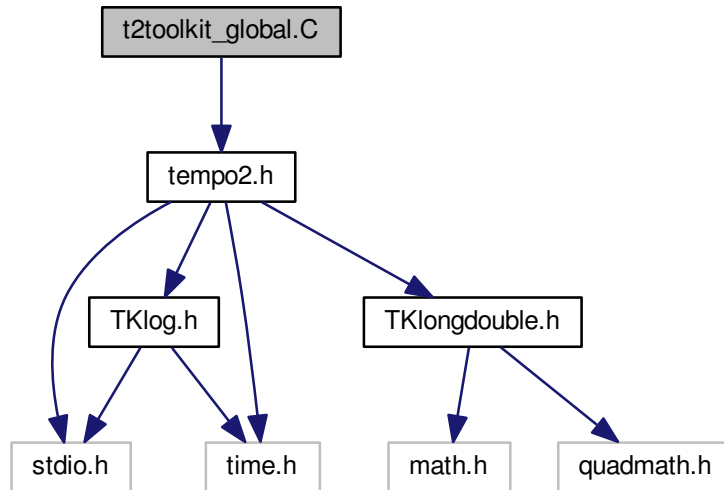
12.151.2.32 double TKvariance_d (double * *x*, int *n*)

12.151.2.33 void TKzeromean_d (int *n*, double * *y*)

12.152 t2toolkit_global.C File Reference

```
#include "tempo2.h"
```

Include dependency graph for t2toolkit_global.C:



Variables

- int `MAX_OBSN` = `MAX_OBSN_VAL`

12.152.1 Variable Documentation

12.152.1.1 int `MAX_OBSN` = `MAX_OBSN_VAL`

size of the arrays of [observations](#) inside each [pulsar](#)

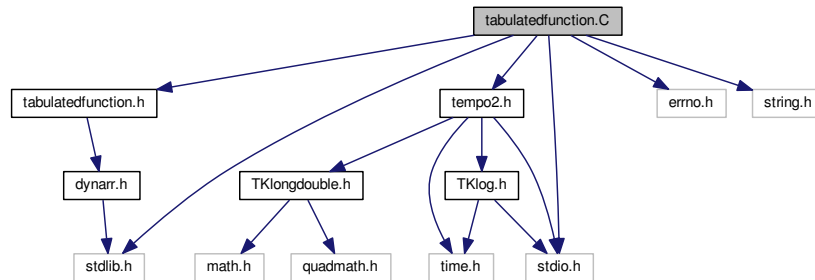
12.153 tabulatedfunction.C File Reference

```

#include "tabulatedfunction.h"
#include "tempo2.h"
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>

```

Include dependency graph for tabulatedfunction.C:



Functions

- void [TabulatedFunction_load](#) ([TabulatedFunction](#) *func, char *fileName)
- double [TabulatedFunction_getValue](#) ([TabulatedFunction](#) *func, double x)
- double [TabulatedFunction_getStartX](#) ([TabulatedFunction](#) *func)
- double [TabulatedFunction_getEndX](#) ([TabulatedFunction](#) *func)

12.153.1 Function Documentation

12.153.1.1 double [TabulatedFunction_getEndX](#) ([TabulatedFunction](#) * *func*)

12.153.1.2 double [TabulatedFunction_getStartX](#) ([TabulatedFunction](#) * *func*)

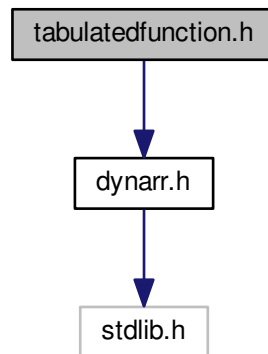
12.153.1.3 double [TabulatedFunction_getValue](#) ([TabulatedFunction](#) * *func*, double *x*)

12.153.1.4 void [TabulatedFunction_load](#) ([TabulatedFunction](#) * *func*, char * *fileName*)

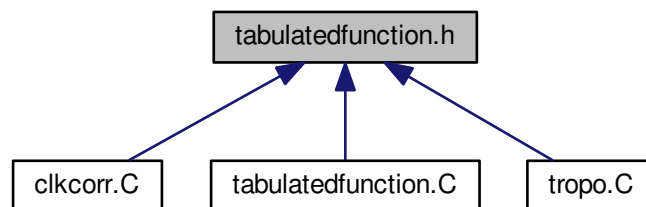
12.154 tabulatedfunction.h File Reference

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

Include dependency graph for tabulatedfunction.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [TabulatedFunctionSample](#)
- struct [TabulatedFunction](#)

Functions

- void [TabulatedFunction_load](#) ([TabulatedFunction](#) *func, char *fileName)
- double [TabulatedFunction_getValue](#) ([TabulatedFunction](#) *func, double x)
- double [TabulatedFunction_getStartX](#) ([TabulatedFunction](#) *func)
- double [TabulatedFunction_getEndX](#) ([TabulatedFunction](#) *func)

12.154.1 Function Documentation

12.154.1.1 `double TabulatedFunction_getEndX (TabulatedFunction * func)`

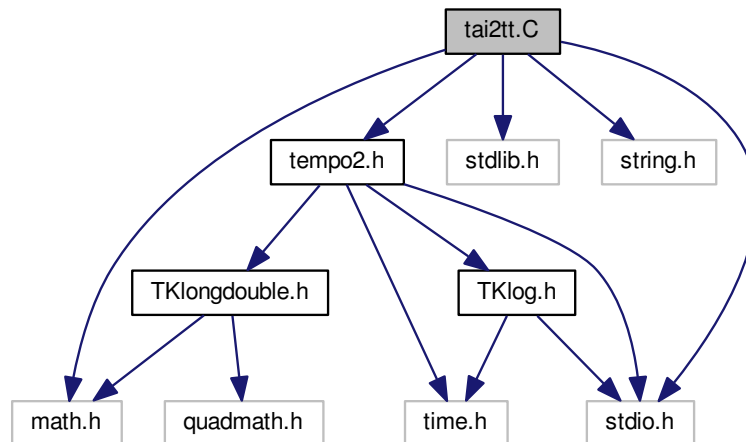
12.154.1.2 `double TabulatedFunction_getStartX (TabulatedFunction * func)`

12.154.1.3 `double TabulatedFunction_getValue (TabulatedFunction * func, double x)`

12.154.1.4 `void TabulatedFunction_load (TabulatedFunction * func, char * fileName)`

12.155 tai2tt.C File Reference

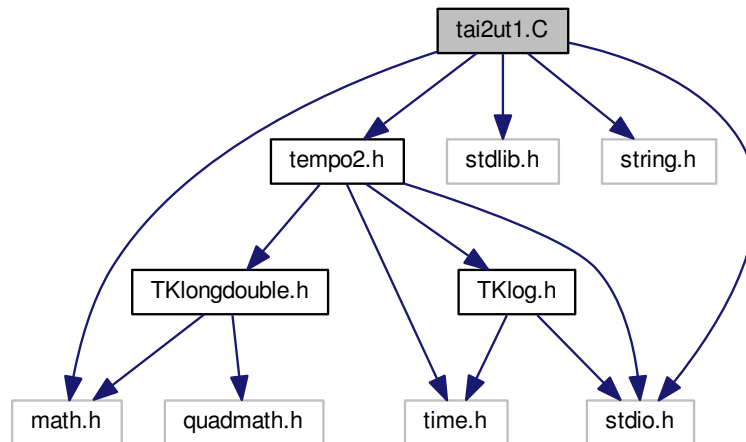
```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"
Include dependency graph for tai2tt.C:
```



12.156 tai2ut1.C File Reference

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"
```

Include dependency graph for tai2ut1.C:



Functions

- double [ut1red](#) (double *mjd*, int *warnings*)
- void [tai2ut1](#) (*pulsar* **psr*, int *npsr*)

12.156.1 Function Documentation

12.156.1.1 void [tai2ut1](#) (*pulsar* * *psr*, int *npsr*)

12.156.1.2 double [ut1red](#) (double *mjd*, int *warnings*)

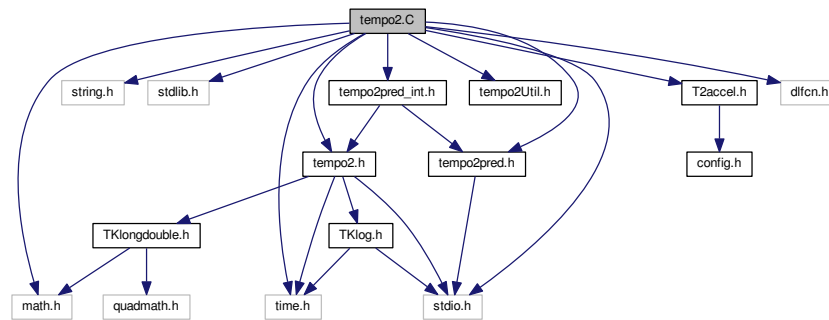
12.157 tempo2.C File Reference

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "tempo2.h"
#include "tempo2Util.h"
#include "tempo2pred.h"
#include "tempo2pred_int.h"
#include "T2accel.h"
#include <dlfcn.h>

```

Include dependency graph for tempo2.C:



Functions

- void [ephemeris_routines](#) ([pulsar](#) *psr, int npsr)
- void [clock_corrections](#) ([pulsar](#) *psr, int npsr)
- void [extra_delays](#) ([pulsar](#) *psr, int npsr)
- int [main](#) (int argc, char *argv[])
- void [thwart_annoying_dynamic_library_stuff](#) (int never_call_me, float or_sink)

12.157.1 Function Documentation

12.157.1.1 void [clock_corrections](#) ([pulsar](#) * *psr*, int *npsr*)

12.157.1.2 void [ephemeris_routines](#) ([pulsar](#) * *psr*, int *npsr*)

12.157.1.3 void [extra_delays](#) ([pulsar](#) * *psr*, int *npsr*)

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

12.157.1.5 void [thwart_annoying_dynamic_library_stuff](#) (int *never_call_me*, float *or_sink*)

12.158 tempo2.h File Reference

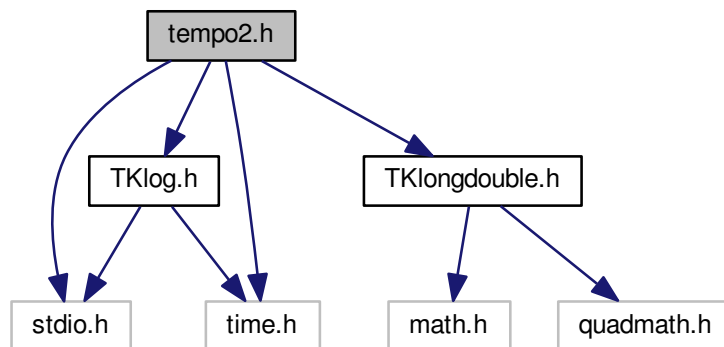
contains the main interface to libtempo2.

```

#include <stdio.h>
#include <time.h>
#include "TKlongdouble.h"
#include "TKlog.h"

```

Include dependency graph for tempo2.h:



Classes

- struct [FitInfo](#)
contains details of the fit
- struct [storePrecision](#)
- struct [parameter](#)
Holds the values for a parameter.
- struct [clock_correction](#)
- struct [observation](#)
A struct containing the details of a single obesrvation.
- struct [pulsar](#)
contains the details for a single pulsar.
- struct [observatory](#)

Macros

- #define [TEMPO2_h_HASH](#) "\$Id: da810cd817da8229f1a155b119a771e9e962a9b7 \$"
- #define [TEMPO2_h_VER](#) "2015.09.0"
- #define [TEMPO2_h_MAJOR_VER](#) 2015.09
- #define [TEMPO2_h_MINOR_VER](#) 0
- #define [TSUN](#) longdouble(4.925490947e-6)
- #define [MAX_FREQ_DERIVATIVES](#) 13
- #define [MAX_DM_DERIVATIVES](#) 10
- #define [MAX_PSR_VAL](#) 40
- #define [MAX_COMPANIONS](#) 4
- #define [NE_SW_DEFAULT](#) 4
- #define [ECLIPTIC_OBLIQUITY_VAL](#) 84381.4059
- #define [MAX_COEFF](#) 5000
- #define [MAX_CLKCORR](#) 5000
- #define [MAX_LEAPSEC](#) 100
- #define [MAX_STRLEN](#) 1000
- #define [MAX_FILELEN](#) 500
- #define [MAX_STOREPRECISION](#) 50

- #define `MAX_OBSN_VAL` 20000
- #define `MAX_SITE` 100
- #define `MAX_PARAMS` 2000
- #define `MAX_JUMPS` 2000
- #define `MAX_WHITE` 100
- #define `MAX_IFUNC` 1000
- #define `MAX_TEL_CLK_OFFS` 500
- #define `MAX_TEL_DX` 500
- #define `MAX_TEL_DY` 500
- #define `MAX_TEL_DZ` 500
- #define `MAX_FIT` 10000
- #define `MAX_T2EFAC` 100
- #define `MAX_T2EQUAD` 100
- #define `MAX_TNEF` 50
- #define `MAX_TNEQ` 50
- #define `MAX_TNGN` 50
- #define `MAX_TNBN` 50 /*maximum number of TNBandNoise parameters allowed*/
- #define `MAX_TNECORR` 50
- #define `MAX_TNDMEv` 10 /*Maximum number of TNDMEvents allowed */
- #define `MAX_TNSQ` 50
- #define `MAX_BPJ_JUMPS` 5
- #define `MAX_TOFFSET` 10
- #define `MAX_QUAD` 150
- #define `MAX_DMx` 512
- #define `MAX_FLAGS` 20
- #define `MAX_FLAG_LEN` 32
- #define `MAX_CLK_CORR` 30
- #define `SECDAY` 86400.0
- #define `SECDAYl` longdouble(86400.0)
- #define `SPEED_LIGHT` 299792458.0
- #define `SOLAR_MASS` 1.98892e30
- #define `SOLAR_RADIUS` 6.96e8
- #define `BIG_G` 6.673e-11
- #define `GM` 1.3271243999e20
- #define `GM_C3` 4.925490947e-6
- #define `GMJ_C3` 4.70255e-9
- #define `GMS_C3` 1.40797e-9
- #define `GMV_C3` 1.2061e-11
- #define `GMU_C3` 2.14539e-10
- #define `GMN_C3` 2.54488e-10
- #define `OBLQ` 23.445833333333333
- #define `AULTSC` 499.00478364
- #define `AU_DIST` 1.49598e11
- #define `DM_CONST` 2.41e-4
- #define `DM_CONST_SI` 7.436e6
- #define `PCM` 3.08568025e16
- #define `MASYR2RADS` 1.53628185e-16
- #define `MAX_MSG` 50
- #define `LEAPSECOND_FILE` "/clock/leap.sec"
- #define `UT1_FILE` "/clock/ut1.dat"
- #define `TDBTDT_FILE` "/ephemeris/TDB.1950.2050"
- #define `IFTEPH_FILE` "/ephemeris/TIMEEPH_short.te405"
- #define `OBSSYS_FILE` "/observatory/newobsys.dat"
- #define `SI_UNITS` 1
- #define `TDB_UNITS` 2

- `#define IF99_TIMEEPH 1`
- `#define FB90_TIMEEPH 2`
- `#define T2C_IAU2000B 1`
- `#define T2C_TEMPO 2`
- `#define HAVE_GWSIM_H`

Typedefs

- `typedef int param_label`
- `typedef int constraint_label`
- `typedef double(* paramDerivFunc) (struct pulsar *, int, double, int, param_label, int)`
a function used to get the derivative of a parameter w.r.t. data.
- `typedef double(* constraintDerivFunc) (struct pulsar *, int, constraint_label, param_label, int, int)`
a function used to get the derivative of a parameter w.r.t. constraint.
- `typedef void(* paramUpdateFunc) (struct pulsar *, int, param_label, int, double, double)`
a function used to update the parameters after a fit.
- `typedef struct FitInfo FitInfo`
contains details of the fit
- `typedef struct storePrecision storePrecision`
- `typedef struct parameter parameter`
Holds the values for a parameter.
- `typedef struct observation observation`
A struct containing the details of a single obesrvation.
- `typedef struct pulsar pulsar`
contains the details for a single pulsar.

Enumerations

- enum `label` {
`param_raj`, `param_decj`, `param_f`, `param_pepoch`,
`param_posepoch`, `param_dmepoch`, `param_dm`, `param_pmra`,
`param_pmdec`, `param_px`, `param_sini`, `param_pb`,
`param_fb`, `param_t0`, `param_a1`, `param_om`,
`param_pmr`, `param_ecc`, `param_edot`, `param_e2dot`,
`param_xpbdot`, `param_pbdot`, `param_a1dot`, `param_a2dot`,
`param_omdot`, `param_om2dot`, `param_orbpx`, `param_tasc`,
`param_eps1`, `param_eps2`, `param_m2`, `param_gamma`,
`param_mtot`, `param_glep`, `param_glph`, `param_glf0`,
`param_glf1`, `param_glf2`, `param_glf0d`, `param_gltd`,
`param_start`, `param_finish`, `param_track`, `param_bp`,
`param_bpp`, `param_tzrmjd`, `param_tzrfreq`, `param_fddc`,
`param_fddi`, `param_fd`, `param_dr`, `param_dtheta`,
`param_tspan`, `param_bpjep`, `param_bpjph`, `param_bpja1`,
`param_bpjec`, `param_bpjom`, `param_bpjpb`, `param_wave_om`,
`param_kom`, `param_kin`, `param_shapmax`, `param_dth`,
`param_a0`, `param_b0`, `param_xomdot`, `param_afac`,
`param_eps1dot`, `param_eps2dot`, `param_tres`, `param_wave_dm`,
`param_waveepoch_dm`, `param_dshk`, `param_ephver`, `param_daop`,
`param_iperharm`, `param_dmassplanet`, `param_waveepoch`, `param_ifunc`,
`param_clk_offs`, `param_dmx`, `param_dmrx1`, `param_dmrx2`,
`param_dmmodel`, `param_gwsingle`, `param_cgw`, `param_quad_om`,
`param_h3`, `param_h4`, `param_nharm`, `param_stig`,
`param_telx`, `param_tely`, `param_telz`, `param_telEpoch`,
`param_quad_ifunc_p`, `param_quad_ifunc_c`, `param_tel_dx`, `param_tel_dy`,
`param_tel_dz`, `param_tel_vx`, `param_tel_vy`, `param_tel_vz`,
`param_tel_x0`, `param_tel_y0`, `param_tel_z0`, `param_gwm_amp`,
`param_gwecc`, `param_gwb_amp`, `param_dm_sin1yr`, `param_dm_cos1yr`,
`param_brake`, `param_stateSwitchT`, `param_df1`, `param_LAST`,
`param_ZERO`, `param_JUMP` }
enumeration for the various parameters that appear in a .par file
- enum `constraint` {
`constraint_dmmodel_mean`, `constraint_dmmodel_dm1`, `constraint_dmmodel_cw_0`, `constraint_dmmodel_cw_1`,
`constraint_dmmodel_cw_2`, `constraint_dmmodel_cw_3`, `constraint_ifunc_0`, `constraint_ifunc_1`,
`constraint_ifunc_2`, `constraint_tel_dx_0`, `constraint_tel_dx_1`, `constraint_tel_dx_2`,
`constraint_tel_dy_0`, `constraint_tel_dy_1`, `constraint_tel_dy_2`, `constraint_tel_dz_0`,
`constraint_tel_dz_1`, `constraint_tel_dz_2`, `constraint_quad_ifunc_p_0`, `constraint_quad_ifunc_p_1`,
`constraint_quad_ifunc_p_2`, `constraint_quad_ifunc_c_0`, `constraint_quad_ifunc_c_1`, `constraint_quad_ifunc_c_2`,
`constraint_dmmodel_cw_year_sin`, `constraint_dmmodel_cw_year_cos`, `constraint_dmmodel_cw_year_xsin`,
`constraint_dmmodel_cw_year_xcos`,
`constraint_dmmodel_cw_year_sin2`, `constraint_dmmodel_cw_year_cos2`, `constraint_dmmodel_cw_px`,
`constraint_ifunc_year_sin`,
`constraint_ifunc_year_cos`, `constraint_ifunc_year_xsin`, `constraint_ifunc_year_xcos`, `constraint_ifunc_year_sin2`,
`constraint_ifunc_year_cos2`, `constraint_qifunc_p_year_sin`, `constraint_qifunc_p_year_cos`, `constraint_qifunc_p_year_xsin`,
`constraint_qifunc_p_year_xcos`, `constraint_qifunc_p_year_sin2`, `constraint_qifunc_p_year_cos2`, `constraint_qifunc_c_year_sin`,
`constraint_qifunc_c_year_cos`, `constraint_qifunc_c_year_xsin`, `constraint_qifunc_c_year_xcos`, `constraint_qifunc_c_year_sin2`,
`constraint_qifunc_c_year_cos2`, `constraint_LAST` }
These represent the possible constraints to the fit that have been implemented.

Functions

- int [id_residual](#) (float xcurs, float ycurs)
- float [setStart](#) (float xcurs, float ycurs, int flag)
- int [zoom_graphics](#) (float xcurs2, float ycurs2, int flag)
- void [getInputs](#) ([pulsar](#) *psr, int argc, char *argv[], char [timFile](#)[[[MAX_FILELEN](#)], char [parFile](#)[[[MAX_FILELEN](#)], int *displayParams, int *npsr, int *nGlobal, int *outRes, int *writeModel, char *outputSO, int *polyco, char *polyco_args, char *polyco_file, int *newpar, int *onlypre, char *dcmFile, char *covarFuncFile, char *newparname)
- void [polyco](#) ([pulsar](#) *psr, int npsr, [longdouble](#) polyco_MJD1, [longdouble](#) polyco_MJD2, int nspan, int ncoeff, [longdouble](#) maxha, char *sitename, [longdouble](#) freq, [longdouble](#) coeff[[MAX_COEFF](#)], int trueDM, char *polyco_file)
- void [readParfile](#) ([pulsar](#) *psr, char [parFile](#)[[[MAX_FILELEN](#)], char [timFile](#)[[[MAX_FILELEN](#)], int npsr)
- void [readParfileGlobal](#) ([pulsar](#) *psr, int npsr, char tpar[[MAX_STRLEN](#)][[MAX_FILELEN](#)], char ttim[[MAX_STRLEN](#)][[MAX_FILELEN](#)])
- int [readSimpleParfile](#) (FILE *fin, [pulsar](#) *p)
- int [setupParameterFileDefaults](#) ([pulsar](#) *p)
- void [displayParameters](#) (int pos, char timeFile[[[MAX_FILELEN](#)], char [parFile](#)[[[MAX_FILELEN](#)], [pulsar](#) *psr, int npsr)
- void [initialise](#) ([pulsar](#) *psr, int noWarnings)
- void [initialiseOne](#) ([pulsar](#) *psr, int noWarnings, int fullSetup)
- void [destroyOne](#) ([pulsar](#) *psr)
- void [recordPrecision](#) ([pulsar](#) *psr, [longdouble](#) prec, const char *routine, const char *comment)
- void [readTimfile](#) ([pulsar](#) *psr, char [timFile](#)[[[MAX_FILELEN](#)], int npsr)
- void [formBats](#) ([pulsar](#) *psr, int npsr)
- void [formBatsAll](#) ([pulsar](#) *psr, int npsr)
- void [updateBatsAll](#) ([pulsar](#) *psr, int npsr)
- void [formResiduals](#) ([pulsar](#) *psr, int npsr, int [removeMean](#))
- int [bootstrap](#) ([pulsar](#) *psr, int p, int npsr)
- void [doFitAll](#) ([pulsar](#) *psr, int npsr, const char *covarFuncFile) [DEPRECATED](#)
- void [doFit](#) ([pulsar](#) *psr, int npsr, int writeModel) [DEPRECATED](#)
- void [doFitDCM](#) ([pulsar](#) *psr, const char *dcmFile, const char *covarFuncFile, int npsr, int writeModel) [DEPRECATED](#)
- void [doFitGlobal](#) ([pulsar](#) *psr, int npsr, double *globalParameter, int nGlobal, int writeModel) [DEPRECATED](#)
- void [getCholeskyMatrix](#) (double **uinv, const char *fname, [pulsar](#) *psr, double *resx, double *resy, double *rese, int np, int nc, int ip)
- double [getParamDeriv](#) ([pulsar](#) *psr, int ipos, double x, int i, int k) [DEPRECATED](#)
- void [textOutput](#) ([pulsar](#) *psr, int npsr, double globalParameter, int nGlobal, int outRes, int newpar, const char *fname)
- void [shapiro_delay](#) ([pulsar](#) *psr, int npsr, int p, int i, double delt, double dt_SSB)
- void [dm_delays](#) ([pulsar](#) *psr, int npsr, int p, int i, double delt, double dt_SSB)
- void [calculate_bclt](#) ([pulsar](#) *psr, int npsr)
- void [secularMotion](#) ([pulsar](#) *psr, int npsr)
- void [autoConstraints](#) ([pulsar](#) *psr, int ipsr, int npsr)
- void [setPlugPath](#) ()
- void [sortToAs](#) ([pulsar](#) *psr)
- void [preProcess](#) ([pulsar](#) *psr, int npsr, int argc, char *argv[])
- void [preProcessSimple](#) ([pulsar](#) *psr)
- void [preProcessSimple1](#) ([pulsar](#) *psr, int tempo1, double thelast)
- void [preProcessSimple2](#) ([pulsar](#) *psr, float startdmmjd, int ndm, float *dmvals, int trimonly)
- void [preProcessSimple3](#) ([pulsar](#) *psr)
- void [useSelectFile](#) (char *fname, [pulsar](#) *psr, int npsr)
- void [processSimultaneous](#) (char *line, [pulsar](#) *psr, int npsr)
- void [processFlag](#) (char *line, [pulsar](#) *psr, int npsr)
- void [logicFlag](#) (char *line, [pulsar](#) *psr, int npsr)

- void [toa2utc](#) ([pulsar](#) *psr, int npsr)
- void [utc2tai](#) ([pulsar](#) *psr, int npsr)
- void [tt2tb](#) ([pulsar](#) *psr, int npsr)
- void [tai2tt](#) ([pulsar](#) *psr, int npsr)
- void [tai2ut1](#) ([pulsar](#) *psr, int npsr)
- void [vectorPulsar](#) ([pulsar](#) *psr, int npsr)
- void [readEphemeris](#) ([pulsar](#) *psr, int npsr, int addEphemNoise)
- void [readOneEphemeris](#) ([pulsar](#) *psr, int npsr, int addEphemNoise, int obsNumber)
- void [readEphemeris_calceph](#) ([pulsar](#) *psr, int npsr)
- void [get_obsCoord](#) ([pulsar](#) *psr, int npsr)
- void [get_OneobsCoord](#) ([pulsar](#) *psr, int npsr, int obs)
- double [calcRMS](#) ([pulsar](#) *psr, int p)
- void [allocateMemory](#) ([pulsar](#) *psr, int realloc)
- void [destroyMemory](#) ([pulsar](#) *psr)
- void [readJBO_bat](#) (char *fname, [pulsar](#) *psr, int p)
- void [readObsFile](#) (double alat[[MAX_SITE](#)], double along[[MAX_SITE](#)], double elev[[MAX_SITE](#)], int icoord[[MAX_SITE](#)], char obsnam[[MAX_SITE](#)][100], char obscode[[MAX_SITE](#)][100], int *nobservatory, int obsnum[[MAX_SITE](#)])
- double [dotproduct](#) (double *v1, double *v2)
- void [vectorsum](#) (double *res, double *v1, double *v2)
- void [vectorscale](#) (double *v, double k)
- void [writeTim](#) (const char *timname, [pulsar](#) *psr, const char *fileFormat)
- int [turn_hms](#) (double turn, char *hms)
- int [turn_dms](#) (double turn, char *dms)
- double [dms_turn](#) (char *line)
- double [hms_turn](#) (char *line)
- double [turn_deg](#) (double turn)
- longdouble [fortran_mod](#) (longdouble a, longdouble p)
- int [fortran_nint](#) (double x)
- long [fortran_nlong](#) (longdouble x)
- void [equ2ecl](#) (double *x)
- void [copyParam](#) ([parameter](#) p1, [parameter](#) *p2)
- void [copyPSR](#) ([pulsar](#) *p, int p1, int p2)
- longdouble [getParameterValue](#) ([pulsar](#) *psr, int param, int arr)
- void [simplePlot](#) ([pulsar](#) *psr, double unitFlag)
- double [solarWindModel](#) ([pulsar](#) psr, int iobs)
- double [MSSmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateMSS](#) ([pulsar](#) *psr, double val, double err, int pos)
- double [BTmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateBT](#) ([pulsar](#) *psr, double val, double err, int pos)
- double [BTJmodel](#) ([pulsar](#) *psr, int p, int obs, int param, int arr)
- void [updateBTJ](#) ([pulsar](#) *psr, double val, double err, int pos, int arr)
- double [BTXmodel](#) ([pulsar](#) *psr, int p, int obs, int param, int arr)
- void [updateBTX](#) ([pulsar](#) *psr, double val, double err, int pos, int arr)
- double [ELL1model](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateELL1](#) ([pulsar](#) *psr, double val, double err, int pos)
- longdouble [DDmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateDD](#) ([pulsar](#) *psr, double val, double err, int pos)
- double [T2model](#) ([pulsar](#) *psr, int p, int obs, int param, int arr)
- void [updateT2](#) ([pulsar](#) *psr, double val, double err, int pos, int arr)
- double [T2_PTAmodel](#) ([pulsar](#) *psr, int p, int obs, int param, int arr)
- void [updateT2_PTA](#) ([pulsar](#) *psr, double val, double err, int pos, int arr)
- double [JVmodel](#) ([pulsar](#) *psr, int p, int obs, int param, int arr)
- void [updateJV](#) ([pulsar](#) *psr, double val, double err, int pos, int arr)
- double [DDKmodel](#) ([pulsar](#) *psr, int p, int obs, int param)

- void [updateDDK](#) ([pulsar](#) *psr, double val, double err, int pos)
- double [DDSmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateDDS](#) ([pulsar](#) *psr, double val, double err, int pos)
- double [DDGRmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateDDGR](#) ([pulsar](#) *psr, double val, double err, int pos)
- double [DDHmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateDDH](#) ([pulsar](#) *psr, double val, double err, int pos)
- double [ELL1Hmodel](#) ([pulsar](#) *psr, int p, int obs, int param)
- void [updateELL1H](#) ([pulsar](#) *psr, double val, double err, int pos)
- void [displayMsg](#) (int type, const char *key, const char *searchStr, const char *variableStr, int noWarnings)
- void [CVSdisplayVersion](#) (const char *file, const char *func, const char *verNum)
- void [transform_units](#) (struct [pulsar](#) *psr, int from, int to)
- void [FITfuncs](#) (double x, double afunc[], int ma, [pulsar](#) *psr, int ipos, int ipsr)
- void [updateParameters](#) ([pulsar](#) *psr, int p, double *val, double *error)
- void [defineClockCorrectionSequence](#) (char *fileList, int dispWarnings)
- void [getClockCorrections](#) ([observation](#) *obs, const char *clockFrom, const char *clockTo, int warnings)
- double [getCorrectionTT](#) ([observation](#) *obs)
- double [getCorrection](#) ([observation](#) *obs, const char *clockFrom, const char *clockTo, int warnings)
- [observatory](#) * [getObservatory](#) (char *code)
- void [lookup_observatory_alias](#) (char *incode, char *outcode)
- void [get_obsCoord_IAU2000B](#) (double [observatory_trs](#)[3], double [zenith_trs](#)[3], [longdouble](#) tt_mjd, [longdouble](#) utc_mjd, double [observatory_crs](#)[3], double [zenith_crs](#)[3], double [observatory_velocity_crs](#)[3])
- void [get_EOP](#) (double mjd, double *xp, double *yp, double *dut1, double *dut1dot, int dispWarnings, char *eopcFile)
- void [compute_tropospheric_delays](#) ([pulsar](#) *psr, int npsr)

Variables

- char [TEMPO2_ENVIRON](#) []
- char [TEMPO2_ERROR](#) []
- char [NEWFIT](#)
- int [MAX_PSR](#)
- int [MAX_OBSN](#)
- double [ECLIPTIC_OBLIQUITY](#)
- int [forceGlobalFit](#)
- int [veryFast](#)
- char [tempo2MachineType](#) [[MAX_FILELEN](#)]
- int [displayCVSversion](#)
- char [dcmFile](#) [[MAX_FILELEN](#)]
- char [covarFuncFile](#) [[MAX_FILELEN](#)]
- char [tempo2_plug_path](#) [32][[MAX_STRLEN](#)]
- int [tempo2_plug_path_len](#)

12.158.1 Detailed Description

contains the main interface to libtempo2.

Note

some parts of this to be moved to an internal interface

12.158.2 Macro Definition Documentation

12.158.2.1 `#define AU_DIST 1.49598e11`

1 AU in m

12.158.2.2 `#define AULTSC 499.00478364`

Number of light seconds in 1 AU

12.158.2.3 `#define BIG_G 6.673e-11`

Gravitational constant

12.158.2.4 `#define DM_CONST 2.41e-4`

12.158.2.5 `#define DM_CONST_SI 7.436e6`

Dispersion constant in SI units

12.158.2.6 `#define ECLIPTIC_OBLIQUITY_VAL 84381.4059`

mean obliquity of ecliptic in arcsec

12.158.2.7 `#define FB90_TIMEEPH 2`

Fairhead & Bretagnon time ephemeris

12.158.2.8 `#define GM 1.3271243999e20`

Gravitational constant * mass sun

12.158.2.9 `#define GM_C3 4.925490947e-6`

GM_{\odot}/c^3 (in seconds)

12.158.2.10 `#define GMJ_C3 4.70255e-9`

GM_{jupiter}/c^3 (in seconds)

12.158.2.11 `#define GMN_C3 2.54488e-10`

GM_{neptune}/c^3 (in seconds)

12.158.2.12 `#define GMS_C3 1.40797e-9`

GM_{saturn}/c^3 (in seconds)

12.158.2.13 `#define GMU_C3 2.14539e-10`

GM_uranus/c³ (in seconds)

12.158.2.14 `#define GMV_C3 1.2061e-11`

GM_venus/c³ (in seconds)

12.158.2.15 `#define HAVE_GWSIM_H`

12.158.2.16 `#define IF99_TIMEEPH 1`

Irwin & Fukushima time ephemeris

12.158.2.17 `#define IFTEPH_FILE "/ephemeris/TIMEEPH_short.te405"`

12.158.2.18 `#define LEAPSECOND_FILE "/clock/leap.sec"`

Path for the file containing dates when leap seconds should be added

12.158.2.19 `#define MASR2RADS 1.53628185e-16`

Converts from mas/yr to rad/s

12.158.2.20 `#define MAX_BPJ_JUMPS 5`

Maximum number of jumps in binary params - for BPJ model

12.158.2.21 `#define MAX_CLK_CORR 30`

Maximum number of steps in the correction to TT

12.158.2.22 `#define MAX_CLKCORR 5000`

Maximum number of lines in time.dat file

12.158.2.23 `#define MAX_COEFF 5000`

Maximum number of coefficients in polyco

12.158.2.24 `#define MAX_COMPANIONS 4`

Maximum number of binary companions

12.158.2.25 `#define MAX_DM_DERIVATIVES 10`

DM0 -> DMn where n=10

12.158.2.26 `#define MAX_DM 512`

Max number of DM steps allowed

12.158.2.27 `#define MAX_FILELEN 500`

Maximum filename length

12.158.2.28 `#define MAX_FIT 10000`

Maximum number of parameters to fit for

12.158.2.29 `#define MAX_FLAG_LEN 32`

Maximum number of characters in each flag

12.158.2.30 `#define MAX_FLAGS 20`

Maximum number of flags in .tim file/observation

12.158.2.31 `#define MAX_FREQ_DERIVATIVES 13`

F0 -> Fn where n=10

12.158.2.32 `#define MAX_IFUNC 1000`

Maximum number of parameters for interpolation function

12.158.2.33 `#define MAX_JUMPS 2000`

Maximum number of phase jumps

12.158.2.34 `#define MAX_LEAPSEC 100`

Maximum number of line in the leap second file

12.158.2.35 `#define MAX_MSG 50`

Maximum number of different warnings

12.158.2.36 `#define MAX_OBSN_VAL 20000`

Maximum number of TOAs

12.158.2.37 `#define MAX_PARAMS 2000`

Maximum number of parameters

12.158.2.38 `#define MAX_PSR_VAL 40`

Maximum number of pulsars

12.158.2.39 `#define MAX_QUAD 150`

Maximum number of frequency channels in quadrupolar function

12.158.2.40 `#define MAX_SITE 100`

Maximum number of observatory sites

12.158.2.41 `#define MAX_STOREPRECISION 50`

How many routines in TEMPO2 store precision information

12.158.2.42 `#define MAX_STRLEN 1000`

Maximum length for strings

12.158.2.43 `#define MAX_T2EFAC 100`

Maximum number of T2EFACs allowed

12.158.2.44 `#define MAX_T2EQUAD 100`

Maximum number of T2EQUADs allowed

12.158.2.45 `#define MAX_TEL_CLK_OFFS 500`

Maximum number of parameters for telescope clock offset

12.158.2.46 `#define MAX_TEL_DX 500`

Maximum number of parameters for interpolation function

12.158.2.47 `#define MAX_TEL_DY 500`

Maximum number of parameters for interpolation function

12.158.2.48 `#define MAX_TEL_DZ 500`

Maximum number of parameters for interpolation function

12.158.2.49 `#define MAX_TNBN 50 /*maximum number of TNBAndNoise parameters allowed*/`

12.158.2.50 `#define MAX_TNDMEv 10 /*Maximum number of TNDMEvents allowed */`

12.158.2.51 `#define MAX_TNECORR 50`

Maximum number of TNECORRss allowed

12.158.2.52 `#define MAX_TNEF 50`

Maximum number of TNEFACs allowed

12.158.2.53 `#define MAX_TNEQ 50`

Maximum number of TNEQUADs allowed

12.158.2.54 `#define MAX_TNGN 50`

maximum number of TNGroupNoise parameters allowed

12.158.2.55 `#define MAX_TNSQ 50`

Maximum number of TNEQUADs allowed

12.158.2.56 `#define MAX_TOFFSET 10`

Number of time jumps allowed in .par file

12.158.2.57 `#define MAX_WHITE 100`

Maximum number of parameters for whitening

12.158.2.58 `#define NE_SW_DEFAULT 4`

Default value for electron density (cm⁻³) at 1AU due to solar wind

12.158.2.59 `#define OBLQ 23.445833333333333`

Obliquity of the ecliptic

12.158.2.60 `#define OBSSYS_FILE "/observatory/newobsys.dat"`

Path for file containing Observatory data (obsys.dat)

12.158.2.61 `#define PCM 3.08568025e16`

one parsec in meters

12.158.2.62 `#define SECDAY 86400.0`

Number of seconds in 1 day

12.158.2.63 `#define SECDAYI longdouble(86400.0)`

Number of seconds in 1 day

12.158.2.64 `#define SI_UNITS 1`

New tempo2 mode

12.158.2.65 `#define SOLAR_MASS 1.98892e30`

Mass of Sun (kg)

12.158.2.66 `#define SOLAR_RADIUS 6.96e8`

Radius of the Sun (in meters)

12.158.2.67 `#define SPEED_LIGHT 299792458.0`

Speed of light (m/s)

12.158.2.68 `#define T2C_IAU2000B 1`

12.158.2.69 `#define T2C_TEMPO 2`

12.158.2.70 `#define TDB_UNITS 2`

original tempo mode

12.158.2.71 `#define TDBTDT_FILE "/ephemeris/TDB.1950.2050"`

Path for file containing TDB-TDT ephemeris

12.158.2.72 `#define TEMPO2_h_HASH "$Id: da810cd817da8229f1a155b119a771e9e962a9b7 $"`

12.158.2.73 `#define TEMPO2_h_MAJOR_VER 2015.09`

12.158.2.74 `#define TEMPO2_h_MINOR_VER 0`

12.158.2.75 `#define TEMPO2_h_VER "2015.09.0"`

12.158.2.76 `#define TSUN longdouble(4.925490947e-6)`

Solar constant for mass calculations.

12.158.2.77 `#define UT1_FILE "/clock/ut1.dat"`

Path for the file containing TAI-UT1

12.158.3 Typedef Documentation

12.158.3.1 `typedef int constraint_label`

for 'strong typing' - type for enum constraint

12.158.3.2 `typedef double(* constraintDerivFunc) (struct pulsar *, int, constraint_label, param_label, int, int)`

a function used to get the derivative of a parameter w.r.t. constraint.

Used to build the derivative matrix for the least squares solvers.

12.158.3.3 `typedef struct FitInfo FitInfo`

contains details of the fit

Holds references to the fit functions, as well as references linking the index in the derivative matrix to the actual parameter fit for.

12.158.3.4 `typedef struct observation observation`

A struct containing the details of a single observation.

12.158.3.5 `typedef int param_label`

for 'strong typing' - type for enum label

12.158.3.6 `typedef double(* paramDerivFunc) (struct pulsar *, int, double, int, param_label, int)`

a function used to get the derivative of a parameter w.r.t. data.

Used to build the derivative matrix for the least squares solvers.

12.158.3.7 `typedef struct parameter parameter`

Holds the values for a parameter.

May include multiple values, for e.g. F0, F1, F2,...

Note

If this structure is modified - must update copyParam in [tempo2Util.C](#)

12.158.3.8 `typedef void(* paramUpdateFunc) (struct pulsar *, int, param_label, int, double, double)`

a function used to update the parameters after a fit.

12.158.3.9 `typedef struct pulsar pulsar`

contains the details for a single pulsar.

Includes an array of [observations](#) and [parameters](#)

12.158.3.10 `typedef struct storePrecision storePrecision`

12.158.4 Enumeration Type Documentation

12.158.4.1 `enum constraint`

These represent the possible constraints to the fit that have been implemented.

Enumerator

- `constraint_dmmodel_mean`
- `constraint_dmmodel_dm1`
- `constraint_dmmodel_cw_0`
- `constraint_dmmodel_cw_1`
- `constraint_dmmodel_cw_2`
- `constraint_dmmodel_cw_3`
- `constraint_ifunc_0`
- `constraint_ifunc_1`
- `constraint_ifunc_2`
- `constraint_tel_dx_0`
- `constraint_tel_dx_1`
- `constraint_tel_dx_2`
- `constraint_tel_dy_0`
- `constraint_tel_dy_1`
- `constraint_tel_dy_2`
- `constraint_tel_dz_0`
- `constraint_tel_dz_1`
- `constraint_tel_dz_2`
- `constraint_quad_ifunc_p_0`
- `constraint_quad_ifunc_p_1`
- `constraint_quad_ifunc_p_2`
- `constraint_quad_ifunc_c_0`
- `constraint_quad_ifunc_c_1`
- `constraint_quad_ifunc_c_2`
- `constraint_dmmodel_cw_year_sin`
- `constraint_dmmodel_cw_year_cos`
- `constraint_dmmodel_cw_year_xsin`
- `constraint_dmmodel_cw_year_xcos`
- `constraint_dmmodel_cw_year_sin2`
- `constraint_dmmodel_cw_year_cos2`
- `constraint_dmmodel_cw_px`
- `constraint_ifunc_year_sin`
- `constraint_ifunc_year_cos`
- `constraint_ifunc_year_xsin`
- `constraint_ifunc_year_xcos`
- `constraint_ifunc_year_sin2`
- `constraint_ifunc_year_cos2`

constraint_qifunc_p_year_sin
constraint_qifunc_p_year_cos
constraint_qifunc_p_year_xsin
constraint_qifunc_p_year_xcos
constraint_qifunc_p_year_sin2
constraint_qifunc_p_year_cos2
constraint_qifunc_c_year_sin
constraint_qifunc_c_year_cos
constraint_qifunc_c_year_xsin
constraint_qifunc_c_year_xcos
constraint_qifunc_c_year_sin2
constraint_qifunc_c_year_cos2
constraint_LAST marker for the last constraint

12.158.4.2 enum label

enumeration for the various parameters that appear in a .par file

The last parameter is param_LAST, but there are enumerations after this for special fits. It is important not to change the order of the elements

Note

when adding a new parameter, initialise it in initialise.c after param_LAST.

Enumerator

param_raj
param_decj
param_f
param_pepoch
param_posepoch
param_dmepoch
param_dm
param_pmra
param_pmdec
param_px
param_sini
param_pb
param_fb
param_t0
param_a1
param_om
param_pmr
param_ecc
param_edot
param_e2dot
param_xpbdot

param_pbdot
param_a1dot
param_a2dot
param_omdot
param_om2dot
param_orbpx
param_tasc
param_eps1
param_eps2
param_m2
param_gamma
param_mtot
param_glep
param_glph
param_glf0
param_glf1
param_glf2
param_glf0d
param_gltd
param_start
param_finish
param_track
param_bp
param_bpp
param_tzrmjd
param_tzrfreq
param_fddc
param_fddi
param_fd
param_dr
param_dtheta
param_tspan
param_bpjep
param_bpjph
param_bpja1
param_bpjec
param_bpjom
param_bpjpb
param_wave_om
param_kom
param_kin
param_shapmax
param_dth
param_a0
param_b0

param_xomdot
param_afac
param_eps1dot
param_eps2dot
param_tres
param_wave_dm
param_waveepoch_dm
param_dshk
param_ephver
param_daop
param_iperharm
param_dmassplanet
param_waveepoch
param_ifunc
param_clk_offs
param_dmx
param_dmrx1
param_dmrx2
param_dmmodel
param_gwsingle
param_cgw
param_quad_om
param_h3
param_h4
param_nharm
param_stig
param_telx
param_tely
param_telz
param_telEpoch
param_quad_ifunc_p
param_quad_ifunc_c
param_tel_dx
param_tel_dy
param_tel_dz
param_tel_vx
param_tel_vy
param_tel_vz
param_tel_x0
param_tel_y0
param_tel_z0
param_gwm_amp
param_gwecc
param_gwb_amp
param_dm_sin1yr

param_dm_cos1yr
param_brake
param_stateSwitchT
param_df1
param_LAST Marker for the last param to be used in for loops
param_ZERO virtual parameter for DC offset
param_JUMP virtual parameter for jumps

12.158.5 Function Documentation

- 12.158.5.1 void allocateMemory (pulsar * *psr*, int *realloc*)
- 12.158.5.2 void autoConstraints (pulsar * *psr*, int *ipsr*, int *npsr*)
- 12.158.5.3 int bootstrap (pulsar * *psr*, int *p*, int *npsr*)
- 12.158.5.4 double BTJmodel (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 12.158.5.5 double BTmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 12.158.5.6 double BTXmodel (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 12.158.5.7 double calcRMS (pulsar * *psr*, int *p*)
- 12.158.5.8 void calculate_bclt (pulsar * *psr*, int *npsr*)
- 12.158.5.9 void compute_tropospheric_delays (pulsar * *psr*, int *npsr*)
- 12.158.5.10 void copyParam (parameter *p1*, parameter * *p2*)
- 12.158.5.11 void copyPSR (pulsar * *p*, int *p1*, int *p2*)
- 12.158.5.12 void CVSdisplayVersion (const char * *file*, const char * *func*, const char * *verNum*)
- 12.158.5.13 double DDGRmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 12.158.5.14 double DDHmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 12.158.5.15 double DDKmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 12.158.5.16 longdouble DDmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 12.158.5.17 double DDSmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 12.158.5.18 void defineClockCorrectionSequence (char * *fileList*, int *dispWarnings*)
- 12.158.5.19 void destroyMemory (pulsar * *psr*)
- 12.158.5.20 void destroyOne (pulsar * *psr*)
- 12.158.5.21 void displayMsg (int *type*, const char * *key*, const char * *searchStr*, const char * *variableStr*, int *noWarnings*)
- 12.158.5.22 void displayParameters (int *pos*, char *timeFile*[][MAX_FILELEN], char *parFile*[][MAX_FILELEN], pulsar * *psr*, int *npsr*)

12.158.5.23 void dm_delays (pulsar * *psr*, int *npsr*, int *p*, int *i*, double *delt*, double *dt_SSB*)

12.158.5.24 double dms_turn (char * *line*)

12.158.5.25 void doFit (pulsar * *psr*, int *npsr*, int *writeModel*)

12.158.5.26 void doFitAll (pulsar * *psr*, int *npsr*, const char * *covarFuncFile*)

Master fitting routine with or without cholesky, global or not.

12.158.5.27 void doFitDCM (pulsar * *psr*, const char * *dcmFile*, const char * *covarFuncFile*, int *npsr*, int *writeModel*)

12.158.5.28 void doFitGlobal (pulsar * *psr*, int *npsr*, double * *globalParameter*, int *nGlobal*, int *writeModel*)

12.158.5.29 double dotproduct (double * *v1*, double * *v2*)

12.158.5.30 double ELL1Hmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)

12.158.5.31 double ELL1model (pulsar * *psr*, int *p*, int *obs*, int *param*)

12.158.5.32 void equ2ecl (double * *x*)

12.158.5.33 void FITfuncs (double *x*, double *afunc*[], int *ma*, pulsar * *psr*, int *ipos*, int *ipr*)

12.158.5.34 void formBats (pulsar * *psr*, int *npsr*)

12.158.5.35 void formBatsAll (pulsar * *psr*, int *npsr*)

12.158.5.36 void formResiduals (pulsar * *psr*, int *npsr*, int *removeMean*)

12.158.5.37 longdouble fortran_mod (longdouble *a*, longdouble *p*)

12.158.5.38 int fortran_nint (double *x*)

12.158.5.39 long fortran_nlong (longdouble *x*)

12.158.5.40 void get_EOP (double *mjd*, double * *xp*, double * *yp*, double * *dut1*, double * *dut1dot*, int *dispWarnings*, char * *eopcFile*)

12.158.5.41 void get_obsCoord (pulsar * *psr*, int *npsr*)

12.158.5.42 void get_obsCoord_IAU2000B (double *observatory_trs*[3], double *zenith_trs*[3], longdouble *tt_mjd*, longdouble *utc_mjd*, double *observatory_crs*[3], double *zenith_crs*[3], double *observatory_velocity_crs*[3])

12.158.5.43 void get_OneobsCoord (pulsar * *psr*, int *npsr*, int *obs*)

12.158.5.44 void getCholeskyMatrix (double ** *uinv*, const char * *fname*, pulsar * *psr*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*)

12.158.5.45 void getClockCorrections (observation * *obs*, const char * *clockFrom*, const char * *clockTo*, int *warnings*)

12.158.5.46 double getCorrection (observation * *obs*, const char * *clockFrom*, const char * *clockTo*, int *warnings*)

12.158.5.47 double getCorrectionTT (observation * *obs*)

12.158.5.48 void getInputs (pulsar * *psr*, int *argc*, char * *argv*[], char *timFile*[][MAX_FILELEN], char *parFile*[][MAX_FILELEN], int * *displayParams*, int * *npsr*, int * *nGlobal*, int * *outRes*, int * *writeModel*, char * *outputSO*, int * *polyco*, char * *polyco_args*, char * *polyco_file*, int * *newpar*, int * *onlypre*, char * *dcmFile*, char * *covarFuncFile*, char * *newparname*)

12.158.5.49 observatory* getObservatory (char * *code*)

12.158.5.50 double getParamDeriv (pulsar * *psr*, int *ipos*, double *x*, int *i*, int *k*)

psr->param[param_f].val[0];

12.158.5.51 longdouble getParameterValue (pulsar * *psr*, int *param*, int *arr*)

12.158.5.52 double hms_turn (char * *line*)

12.158.5.53 int id_residual (float *xcurs*, float *ycurs*)

12.158.5.54 void initialise (pulsar * *psr*, int *noWarnings*)

12.158.5.55 void initialiseOne (pulsar * *psr*, int *noWarnings*, int *fullSetup*)

12.158.5.56 double JVmodel (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)

12.158.5.57 void logicFlag (char * *line*, pulsar * *psr*, int *npsr*)

12.158.5.58 void lookup_observatory_alias (char * *incode*, char * *outcode*)

12.158.5.59 double MSSmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)

RAD/pow(365.25*SECDAY,2.);

12.158.5.60 void polyco (pulsar * *psr*, int *npsr*, longdouble *polyco_MJD1*, longdouble *polyco_MJD2*, int *nspan*, int *ncoeff*, longdouble *maxha*, char * *sitename*, longdouble *freq*, longdouble *coeff*[MAX_COEFF], int *trueDM*, char * *polyco_file*)

12.158.5.61 void preProcess (pulsar * *psr*, int *npsr*, int *argc*, char * *argv*[])

12.158.5.62 void preProcessSimple (pulsar * *psr*)

12.158.5.63 void preProcessSimple1 (pulsar * *psr*, int *tempo1*, double *thelast*)

12.158.5.64 void preProcessSimple2 (pulsar * *psr*, float *startdmmjd*, int *ndm*, float * *dmvals*, int *trimonly*)

12.158.5.65 void preProcessSimple3 (pulsar * *psr*)

12.158.5.66 void processFlag (char * *line*, pulsar * *psr*, int *npsr*)

12.158.5.67 void processSimultaneous (char * *line*, pulsar * *psr*, int *npsr*)

12.158.5.68 void readEphemeris (pulsar * *psr*, int *npsr*, int *addEphemNoise*)

12.158.5.69 void readEphemeris_calceph (pulsar * *psr*, int *npsr*)

12.158.5.70 void readJBO_bat (char * *fname*, pulsar * *psr*, int *p*)

- 12.158.5.71 void readObsFile (double *alat*[MAX_SITE], double *along*[MAX_SITE], double *elev*[MAX_SITE], int *icoord*[MAX_SITE], char *obsnam*[MAX_SITE][100], char *obscode*[MAX_SITE][100], int * *nobservatory*, int *obsnum*[MAX_SITE])
- 12.158.5.72 void readOneEphemeris (pulsar * *psr*, int *npsr*, int *addEphemNoise*, int *obsNumber*)
- 12.158.5.73 void readParfile (pulsar * *psr*, char *parFile*[][MAX_FILELEN], char *timFile*[][MAX_FILELEN], int *npsr*)
- 12.158.5.74 void readParfileGlobal (pulsar * *psr*, int *npsr*, char *tpar*[MAX_STRLEN][MAX_FILELEN], char *ttim*[MAX_STRLEN][MAX_FILELEN])
- 12.158.5.75 int readSimpleParfile (FILE * *fin*, pulsar * *p*)
- 12.158.5.76 void readTimfile (pulsar * *psr*, char *timFile*[][MAX_FILELEN], int *npsr*)
- 12.158.5.77 void recordPrecision (pulsar * *psr*, longdouble *prec*, const char * *routine*, const char * *comment*)
- 12.158.5.78 void secularMotion (pulsar * *psr*, int *npsr*)
- 12.158.5.79 void setPlugPath ()
- 12.158.5.80 float setStart (float *xcurs*, float *ycurs*, int *flag*)
- 12.158.5.81 int setupParameterFileDefaults (pulsar * *p*)
- 12.158.5.82 void shapiro_delay (pulsar * *psr*, int *npsr*, int *p*, int *i*, double *delt*, double *dt_SSB*)
- 12.158.5.83 void simplePlot (pulsar * *psr*, double *unitFlag*)
- 12.158.5.84 double solarWindModel (pulsar *psr*, int *iobs*)
- 12.158.5.85 void sortToAs (pulsar * *psr*)

Sort ToAs for one pulsar.

- 12.158.5.86 double T2_PTAmode1 (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 12.158.5.87 double T2model (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 12.158.5.88 void tai2tt (pulsar * *psr*, int *npsr*)
- 12.158.5.89 void tai2ut1 (pulsar * *psr*, int *npsr*)
- 12.158.5.90 void textOutput (pulsar * *psr*, int *npsr*, double *globalParameter*, int *nGlobal*, int *outRes*, int *newpar*, const char * *fname*)

$\text{sqrt}(\text{psr}[0].\text{covar}[ii][ii]*\text{psr}[0].\text{covar}[jj][jj]));$

- 12.158.5.91 void toa2utc (pulsar * *psr*, int *npsr*)
- 12.158.5.92 void transform_units (struct pulsar * *psr*, int *from*, int *to*)
- 12.158.5.93 void tt2tb (pulsar * *psr*, int *npsr*)
- 12.158.5.94 double turn_deg (double *turn*)

12.158.5.95 int turn_dms (double *turn*, char * *dms*)

12.158.5.96 int turn_hms (double *turn*, char * *hms*)

12.158.5.97 void updateBatsAll (pulsar * *psr*, int *npsr*)

12.158.5.98 void updateBT (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.99 void updateBTJ (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)

12.158.5.100 void updateBTX (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)

12.158.5.101 void updateDD (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.102 void updateDDGR (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.103 void updateDDH (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.104 void updateDDK (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.105 void updateDDS (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.106 void updateELL1 (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.107 void updateELL1H (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.108 void updateJV (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)

12.158.5.109 void updateMSS (pulsar * *psr*, double *val*, double *err*, int *pos*)

12.158.5.110 void updateParameters (pulsar * *psr*, int *p*, double * *val*, double * *error*)

12.158.5.111 void updateT2 (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)

12.158.5.112 void updateT2_PTA (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)

12.158.5.113 void useSelectFile (char * *fname*, pulsar * *psr*, int *npsr*)

12.158.5.114 void utc2tai (pulsar * *psr*, int *npsr*)

12.158.5.115 void vectorPulsar (pulsar * *psr*, int *npsr*)

12.158.5.116 void vectorscale (double * *v*, double *k*)

12.158.5.117 void vectorsum (double * *res*, double * *v1*, double * *v2*)

12.158.5.118 void writeTim (const char * *timname*, pulsar * *psr*, const char * *fileFormat*)

12.158.5.119 int zoom_graphics (float *xcurs2*, float *ycurs2*, int *flag*)

12.158.6 Variable Documentation

12.158.6.1 char covarFuncFile[MAX_FILELEN]

12.158.6.2 char dcmFile[MAX_FILELEN]

12.158.6.3 int displayCVSversion

Display CVS version

12.158.6.4 double ECLIPTIC_OBLIQUITY

12.158.6.5 int forceGlobalFit

Global = 1 if we are forcing a global fit

12.158.6.6 int MAX_OBSN

size of the arrays of [observations](#) inside each [pulsar](#)

12.158.6.7 int MAX_PSR

size of the array of [pulsars](#) used in tempo2

12.158.6.8 char NEWFIT

global boolean used to enable new fit.

Warning

 this will be removed in future.

12.158.6.9 char TEMPO2_ENVIRON[]

TEMPO2 environment variable

12.158.6.10 char TEMPO2_ERROR[]

TEMPO2 error messages

12.158.6.11 char tempo2_plug_path[32][MAX_STRLEN]

paths to search for plugins

12.158.6.12 int tempo2_plug_path_len

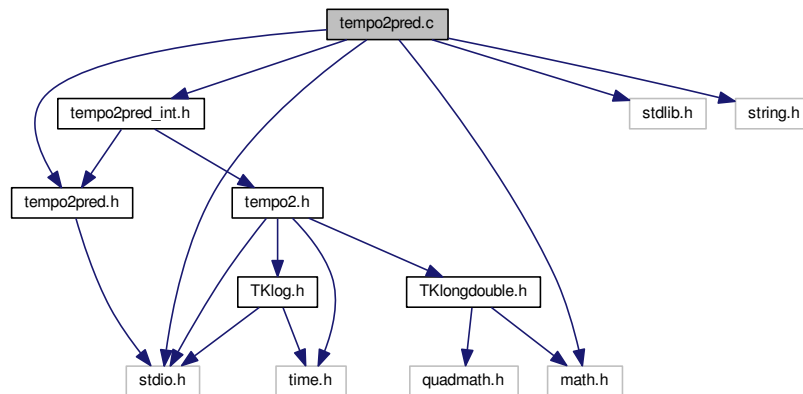
12.158.6.13 char tempo2MachineType[MAX_FILELEN]

12.158.6.14 int veryFast

Global to run the code fast

12.159 tempo2pred.c File Reference

```
#include "tempo2pred.h"
#include "tempo2pred_int.h"
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
Include dependency graph for tempo2pred.c:
```



Functions

- int [T2Predictor_Read](#) (T2Predictor *t2p, char *fname)
- int [T2Predictor_FRead](#) (T2Predictor *t2p, FILE *f)
- void [T2Predictor_Write](#) (const T2Predictor *t2p, char *fname)
- void [T2Predictor_FWrite](#) (const T2Predictor *t2p, FILE *f)
- void [T2Predictor_Init](#) (T2Predictor *t2p)
- void [T2Predictor_Copy](#) (T2Predictor *into_t2p, const T2Predictor *from_t2p)
- int [T2Predictor_Insert](#) (T2Predictor *into_t2p, const T2Predictor *from_t2p)
- void [T2Predictor_Keep](#) (T2Predictor *t2p, unsigned nmjd, const long double *mjd)
- void [T2Predictor_Destroy](#) (T2Predictor *t2p)
- char * [T2Predictor_GetPSRName](#) (T2Predictor *t2p)
- char * [T2Predictor_GetSiteName](#) (T2Predictor *t2p)
- long double [T2Predictor_GetStartMJD](#) (T2Predictor *t2p)
- long double [T2Predictor_GetEndMJD](#) (T2Predictor *t2p)
- long double [T2Predictor_GetStartFreq](#) (T2Predictor *t2p)
- long double [T2Predictor_GetEndFreq](#) (T2Predictor *t2p)
- T2PredictorKind [T2Predictor_Kind](#) (T2Predictor *t2p)
- long double [T2Predictor_GetPhase](#) (const T2Predictor *t2p, long double mjd, long double freq)
- long double [T2Predictor_GetFrequency](#) (const T2Predictor *t2p, long double mjd, long double freq)
- int [T2Predictor_GetPlan](#) (char *filename, long double mjd_start, long double mjd_end, long double step, long double freq, long double *phase0, int *nsegments, long double *pulse_frequencies)
- int [T2Predictor_GetPlan_Ext](#) (char *filename, long double mjd_start, long double mjd_end, long double step, long double freq, char *psrname, char *sitename, long double *phase0, int *nsegments, long double *pulse_frequencies)

Variables

- unsigned `tempo2_verbose` = 1

12.159.1 Function Documentation

- 12.159.1.1 void T2Predictor_Copy (T2Predictor * *into_t2p*, const T2Predictor * *from_t2p*)
- 12.159.1.2 void T2Predictor_Destroy (T2Predictor * *t2p*)
- 12.159.1.3 int T2Predictor_FRead (T2Predictor * *t2p*, FILE * *f*)
- 12.159.1.4 void T2Predictor_FWrite (const T2Predictor * *t2p*, FILE * *f*)
- 12.159.1.5 long double T2Predictor_GetEndFreq (T2Predictor * *t2p*)
- 12.159.1.6 long double T2Predictor_GetEndMJD (T2Predictor * *t2p*)
- 12.159.1.7 long double T2Predictor_GetFrequency (const T2Predictor * *t2p*, long double *mjd*, long double *freq*)
- 12.159.1.8 long double T2Predictor_GetPhase (const T2Predictor * *t2p*, long double *mjd*, long double *freq*)
- 12.159.1.9 int T2Predictor_GetPlan (char * *filename*, long double *mjd_start*, long double *mjd_end*, long double *step*, long double *freq*, long double * *phase0*, int * *nsegments*, long double * *pulse_frequencies*)
- 12.159.1.10 int T2Predictor_GetPlan_Ext (char * *filename*, long double *mjd_start*, long double *mjd_end*, long double *step*, long double *freq*, char * *psrname*, char * *sitename*, long double * *phase0*, int * *nsegments*, long double * *pulse_frequencies*)
- 12.159.1.11 char* T2Predictor_GetPSRName (T2Predictor * *t2p*)
- 12.159.1.12 char* T2Predictor_GetSiteName (T2Predictor * *t2p*)
- 12.159.1.13 long double T2Predictor_GetStartFreq (T2Predictor * *t2p*)
- 12.159.1.14 long double T2Predictor_GetStartMJD (T2Predictor * *t2p*)
- 12.159.1.15 void T2Predictor_Init (T2Predictor * *t2p*)
- 12.159.1.16 int T2Predictor_Insert (T2Predictor * *into_t2p*, const T2Predictor * *from_t2p*)
- 12.159.1.17 void T2Predictor_Keep (T2Predictor * *t2p*, unsigned *nmjd*, const long double * *mjd*)
- 12.159.1.18 T2PredictorKind T2Predictor_Kind (T2Predictor * *t2p*)
- 12.159.1.19 int T2Predictor_Read (T2Predictor * *t2p*, char * *fname*)
- 12.159.1.20 void T2Predictor_Write (const T2Predictor * *t2p*, char * *fname*)

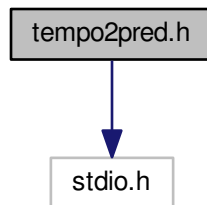
12.159.2 Variable Documentation

- 12.159.2.1 unsigned `tempo2_verbose` = 1

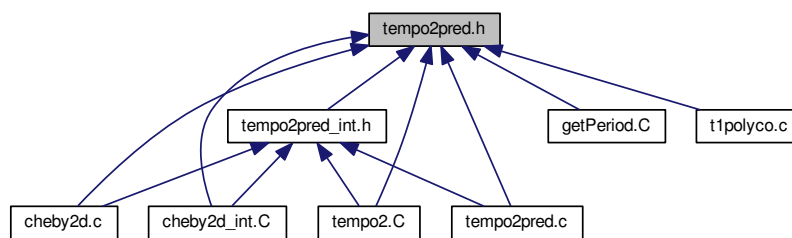
12.160 tempo2pred.h File Reference

```
#include <stdio.h>
```

Include dependency graph for tempo2pred.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [Cheby2D](#)
- struct [ChebyModel](#)
- struct [ChebyModelSet](#)
- struct [T1Polyco](#)
- struct [T1PolycoSet](#)
- struct [T2Predictor](#)

Enumerations

- enum [T2PredictorKind](#) { [NonePredType](#), [Cheby](#), [T1](#) }

Functions

- void [T2Predictor_Init](#) ([T2Predictor](#) *t2p)
- void [T2Predictor_Copy](#) ([T2Predictor](#) *into_t2p, const [T2Predictor](#) *from_t2p)
- int [T2Predictor_Insert](#) ([T2Predictor](#) *into_t2p, const [T2Predictor](#) *from_t2p)
- void [T2Predictor_Keep](#) ([T2Predictor](#) *, unsigned nmjd, const long double *mjd)

- void [T2Predictor_Destroy](#) ([T2Predictor](#) *t2p)
- int [T2Predictor_Read](#) ([T2Predictor](#) *t2p, char *fname)
- int [T2Predictor_FRead](#) ([T2Predictor](#) *t2p, FILE *f)
- void [T2Predictor_Write](#) (const [T2Predictor](#) *t2p, char *fname)
- void [T2Predictor_FWrite](#) (const [T2Predictor](#) *t2p, FILE *f)
- char * [T2Predictor_GetPSRName](#) ([T2Predictor](#) *t2p)
- char * [T2Predictor_GetSiteName](#) ([T2Predictor](#) *t2p)
- long double [T2Predictor_GetStartMJD](#) ([T2Predictor](#) *t2p)
- long double [T2Predictor_GetEndMJD](#) ([T2Predictor](#) *t2p)
- long double [T2Predictor_GetStartFreq](#) ([T2Predictor](#) *t2p)
- long double [T2Predictor_GetEndFreq](#) ([T2Predictor](#) *t2p)
- [T2PredictorKind](#) [T2Predictor_Kind](#) ([T2Predictor](#) *t2p)
- long double [T2Predictor_GetPhase](#) (const [T2Predictor](#) *t2p, long double mjd, long double freq)
- long double [T2Predictor_GetFrequency](#) (const [T2Predictor](#) *t2p, long double mjd, long double freq)
- int [T2Predictor_GetPlan](#) (char *filename, long double mjd_start, long double mjd_end, long double step, long double freq, long double *phase0, int *nsegments, long double *pulse_frequencies)
- int [T2Predictor_GetPlan_Ext](#) (char *filename, long double mjd_start, long double mjd_end, long double step, long double freq, char *psrname, char *sitename, long double *phase0, int *nsegments, long double *pulse_frequencies)

Variables

- int [ChebyModelSet_OutOfRange](#)

12.160.1 Enumeration Type Documentation

12.160.1.1 enum T2PredictorKind

Enumerator

NonePredType

Cheby

T1

12.160.2 Function Documentation

- 12.160.2.1 void [T2Predictor_Copy](#) ([T2Predictor](#) * *into_t2p*, const [T2Predictor](#) * *from_t2p*)
- 12.160.2.2 void [T2Predictor_Destroy](#) ([T2Predictor](#) * *t2p*)
- 12.160.2.3 int [T2Predictor_FRead](#) ([T2Predictor](#) * *t2p*, FILE * *f*)
- 12.160.2.4 void [T2Predictor_FWrite](#) (const [T2Predictor](#) * *t2p*, FILE * *f*)
- 12.160.2.5 long double [T2Predictor_GetEndFreq](#) ([T2Predictor](#) * *t2p*)
- 12.160.2.6 long double [T2Predictor_GetEndMJD](#) ([T2Predictor](#) * *t2p*)
- 12.160.2.7 long double [T2Predictor_GetFrequency](#) (const [T2Predictor](#) * *t2p*, long double *mjd*, long double *freq*)
- 12.160.2.8 long double [T2Predictor_GetPhase](#) (const [T2Predictor](#) * *t2p*, long double *mjd*, long double *freq*)
- 12.160.2.9 int [T2Predictor_GetPlan](#) (char * *filename*, long double *mjd_start*, long double *mjd_end*, long double *step*, long double *freq*, long double * *phase0*, int * *nsegments*, long double * *pulse_frequencies*)

- 12.160.2.10 `int T2Predictor_GetPlan_Ext (char * filename, long double mjd_start, long double mjd_end, long double step, long double freq, char * psrname, char * sitename, long double * phase0, int * nsegments, long double * pulse_frequencies)`
- 12.160.2.11 `char* T2Predictor_GetPSRName (T2Predictor * t2p)`
- 12.160.2.12 `char* T2Predictor_GetSiteName (T2Predictor * t2p)`
- 12.160.2.13 `long double T2Predictor_GetStartFreq (T2Predictor * t2p)`
- 12.160.2.14 `long double T2Predictor_GetStartMJD (T2Predictor * t2p)`
- 12.160.2.15 `void T2Predictor_Init (T2Predictor * t2p)`
- 12.160.2.16 `int T2Predictor_Insert (T2Predictor * into_t2p, const T2Predictor * from_t2p)`
- 12.160.2.17 `void T2Predictor_Keep (T2Predictor * , unsigned nmjd, const long double * mjd)`
- 12.160.2.18 `T2PredictorKind T2Predictor_Kind (T2Predictor * t2p)`
- 12.160.2.19 `int T2Predictor_Read (T2Predictor * t2p, char * fname)`
- 12.160.2.20 `void T2Predictor_Write (const T2Predictor * t2p, char * fname)`

12.160.3 Variable Documentation

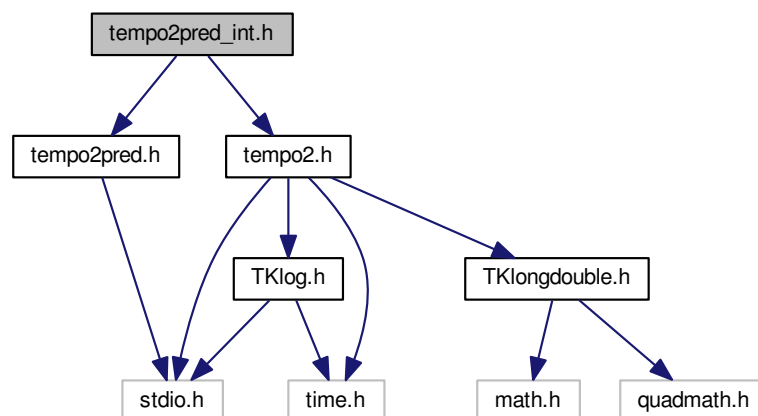
- 12.160.3.1 `int ChebyModelSet_OutOfRange`

12.161 tempo2pred_int.h File Reference

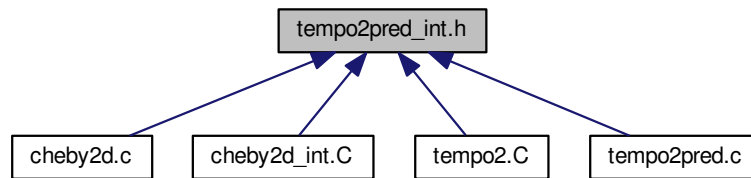
```
#include "tempo2.h"
```

```
#include "tempo2pred.h"
```

Include dependency graph for tempo2pred_int.h:



This graph shows which files directly or indirectly include this file:



Functions

- void [ChebyModel_Construct](#) ([ChebyModel](#) *cm, const [pulsar](#) *psr)
- void [ChebyModel_Test](#) ([ChebyModel](#) *cm, const [pulsar](#) *psr, int nmjd, int nfreq, long double *residualRMS, long double *residualMAV)
- void [ChebyModelSet_Construct](#) ([ChebyModelSet](#) *cms, const [pulsar](#) *psr, const char *sitename, long double mjd_start, long double mjd_end, long double segment_length, long double overlap, long double freq_start, long double freq_end, int nmjdcoeff, int nfreqcoeff)
- void [ChebyModelSet_Test](#) ([ChebyModelSet](#) *cms, const [pulsar](#) *psr, int nmjd, int nfreq, long double *residualRMS, long double *residualMAV)
- void [Cheby2D_Construct](#) ([Cheby2D](#) *cheby, void(*func)(long double *x, long double *y, int nx, int ny, long double *z, void *info), void *info)
- void [Cheby2D_Construct_x_Derivative](#) ([Cheby2D](#) *dcheby, const [Cheby2D](#) *cheby)
- void [Cheby2D_Test](#) ([Cheby2D](#) *cheby, int nx_test, int ny_test, void(*func)(long double *x, long double *y, int nx, int ny, long double *z, void *info), void *info, long double *residualRMS, long double *residualMAV)
- void [ChebyModel_Init](#) ([ChebyModel](#) *cmodel, int nmjdcoeff, int nfreqcoeff)
- void [ChebyModel_Copy](#) ([ChebyModel](#) *cm, [ChebyModel](#) *from)
- void [ChebyModel_Destroy](#) ([ChebyModel](#) *cm)
- long double [ChebyModel_GetPhase](#) (const [ChebyModel](#) *cm, long double mjd, long double freq)
- long double [ChebyModel_GetFrequency](#) (const [ChebyModel](#) *cm, long double mjd, long double freq)
- void [ChebyModel_Write](#) (const [ChebyModel](#) *cm, FILE *f)
- int [ChebyModel_Read](#) ([ChebyModel](#) *cm, FILE *f)
- [ChebyModel](#) * [ChebyModelSet_GetNearest](#) (const [ChebyModelSet](#) *cms, long double mjd)
- long double [ChebyModelSet_GetPhase](#) (const [ChebyModelSet](#) *cms, long double mjd, long double freq)
- long double [ChebyModelSet_GetFrequency](#) (const [ChebyModelSet](#) *cms, long double mjd, long double freq)
- void [ChebyModelSet_Write](#) (const [ChebyModelSet](#) *cms, FILE *f)
- int [ChebyModelSet_Read](#) ([ChebyModelSet](#) *cms, FILE *f)
- void [ChebyModelSet_Init](#) ([ChebyModelSet](#) *cms)
- int [ChebyModelSet_Insert](#) ([ChebyModelSet](#) *cms, const [ChebyModelSet](#) *from)
- void [ChebyModelSet_Keep](#) ([ChebyModelSet](#) *cms, unsigned nmjd, const long double *mjd)
- void [ChebyModelSet_Destroy](#) ([ChebyModelSet](#) *cms)
- long double [T1Polyco_GetPhase](#) (const [T1Polyco](#) *t1p, long double mjd, long double freq)
- long double [T1Polyco_GetFrequency](#) (const [T1Polyco](#) *t1p, long double mjd, long double freq)
- void [T1Polyco_Write](#) (const [T1Polyco](#) *t1p, FILE *f)
- int [T1Polyco_Read](#) ([T1Polyco](#) *t1p, FILE *f)
- [T1Polyco](#) * [T1PolycoSet_GetNearest](#) (long double mjd)
- long double [T1PolycoSet_GetPhase](#) (const [T1PolycoSet](#) *t1ps, long double mjd, long double freq)
- long double [T1PolycoSet_GetFrequency](#) (const [T1PolycoSet](#) *t1ps, long double mjd, long double freq)
- void [T1PolycoSet_Write](#) (const [T1PolycoSet](#) *t1ps, FILE *f)
- int [T1PolycoSet_Read](#) ([T1PolycoSet](#) *t1ps, FILE *f)
- void [T1PolycoSet_Destroy](#) ([T1PolycoSet](#) *t1ps)

12.161.1 Function Documentation

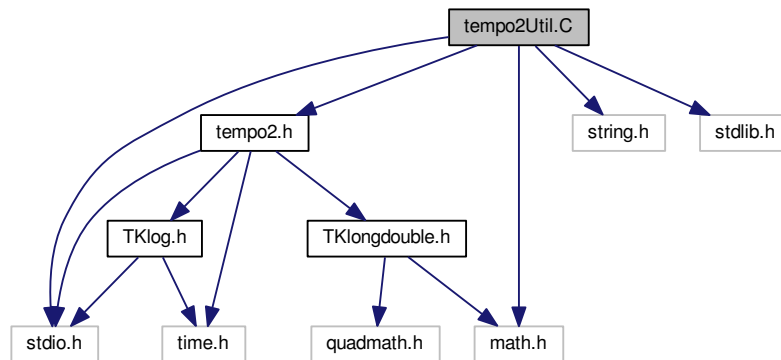
- 12.161.1.1 void Cheby2D_Construct (Cheby2D * *cheby*, void(*)(long double *x, long double *y, int nx, int ny, long double *z, void *info) *func*, void * *info*)
- 12.161.1.2 void Cheby2D_Construct_x_Derivative (Cheby2D * *dcheby*, const Cheby2D * *cheby*)
- 12.161.1.3 void Cheby2D_Test (Cheby2D * *cheby*, int *nx_test*, int *ny_test*, void(*)(long double *x, long double *y, int nx, int ny, long double *z, void *info) *func*, void * *info*, long double * *residualRMS*, long double * *residualMAV*)
- 12.161.1.4 void ChebyModel_Construct (ChebyModel * *cm*, const pulsar * *psr*)
- 12.161.1.5 void ChebyModel_Copy (ChebyModel * *cm*, ChebyModel * *from*)
- 12.161.1.6 void ChebyModel_Destroy (ChebyModel * *cm*)
- 12.161.1.7 long double ChebyModel_GetFrequency (const ChebyModel * *cm*, long double *mjd*, long double *freq*)
- 12.161.1.8 long double ChebyModel_GetPhase (const ChebyModel * *cm*, long double *mjd*, long double *freq*)
- 12.161.1.9 void ChebyModel_Init (ChebyModel * *cmodel*, int *nmjdcoeff*, int *nfreqcoeff*)
- 12.161.1.10 int ChebyModel_Read (ChebyModel * *cm*, FILE * *f*)
- 12.161.1.11 void ChebyModel_Test (ChebyModel * *cm*, const pulsar * *psr*, int *nmjd*, int *nfreq*, long double * *residualRMS*, long double * *residualMAV*)
- 12.161.1.12 void ChebyModel_Write (const ChebyModel * *cm*, FILE * *f*)
- 12.161.1.13 void ChebyModelSet_Construct (ChebyModelSet * *cms*, const pulsar * *psr*, const char * *sitename*, long double *mjd_start*, long double *mjd_end*, long double *segment_length*, long double *overlap*, long double *freq_start*, long double *freq_end*, int *nmjdcoeff*, int *nfreqcoeff*)
- 12.161.1.14 void ChebyModelSet_Destroy (ChebyModelSet * *cms*)
- 12.161.1.15 long double ChebyModelSet_GetFrequency (const ChebyModelSet * *cms*, long double *mjd*, long double *freq*)
- 12.161.1.16 ChebyModel* ChebyModelSet_GetNearest (const ChebyModelSet * *cms*, long double *mjd*)
- 12.161.1.17 long double ChebyModelSet_GetPhase (const ChebyModelSet * *cms*, long double *mjd*, long double *freq*)
- 12.161.1.18 void ChebyModelSet_Init (ChebyModelSet * *cms*)
- 12.161.1.19 int ChebyModelSet_Insert (ChebyModelSet * *cms*, const ChebyModelSet * *from*)
- 12.161.1.20 void ChebyModelSet_Keep (ChebyModelSet * *cms*, unsigned *nmjd*, const long double * *mjd*)
- 12.161.1.21 int ChebyModelSet_Read (ChebyModelSet * *cms*, FILE * *f*)
- 12.161.1.22 void ChebyModelSet_Test (ChebyModelSet * *cms*, const pulsar * *psr*, int *nmjd*, int *nfreq*, long double * *residualRMS*, long double * *residualMAV*)
- 12.161.1.23 void ChebyModelSet_Write (const ChebyModelSet * *cms*, FILE * *f*)
- 12.161.1.24 long double T1Polyco_GetFrequency (const T1Polyco * *t1p*, long double *mjd*, long double *freq*)

- 12.161.1.25 `long double T1Polyco_GetPhase (const T1Polyco * t1p, long double mjd, long double freq)`
- 12.161.1.26 `int T1Polyco_Read (T1Polyco * t1p, FILE * f)`
- 12.161.1.27 `void T1Polyco_Write (const T1Polyco * t1p, FILE * f)`
- 12.161.1.28 `void T1PolycoSet_Destroy (T1PolycoSet * t1ps)`
- 12.161.1.29 `long double T1PolycoSet_GetFrequency (const T1PolycoSet * t1ps, long double mjd, long double freq)`
- 12.161.1.30 `T1Polyco* T1PolycoSet_GetNearest (long double mjd)`
- 12.161.1.31 `long double T1PolycoSet_GetPhase (const T1PolycoSet * t1ps, long double mjd, long double freq)`
- 12.161.1.32 `int T1PolycoSet_Read (T1PolycoSet * t1ps, FILE * f)`
- 12.161.1.33 `void T1PolycoSet_Write (const T1PolycoSet * t1ps, FILE * f)`

12.162 tempo2Util.C File Reference

```
#include <math.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "tempo2.h"
```

Include dependency graph for tempo2Util.C:



Functions

- `double dotproduct (double *v1, double *v2)`
- `void vectorsum (double *res, double *v1, double *v2)`
- `void vectorscale (double *v, double k)`
- `int turn_hms (double turn, char *hms)`
- `int turn_dms (double turn, char *dms)`
- `double turn_deg (double turn)`
- `double hms_turn (char *line)`
- `double dms_turn (char *line)`
- `longdouble fortran_mod (longdouble a, longdouble p)`

- double `fortran_mod` (double *a*, double *p*)
- int `fortran_nint` (double *x*)
- long `fortran_nlong` (longdouble *x*)
- void `equ2ecl` (double **x*)
- void `copyPSR` (pulsar **p*, int *p1*, int *p2*)
- void `copyParam` (parameter *p1*, parameter **p2*)
- void `displayMsg` (int *type*, const char **key*, const char **searchStr*, const char **variableStr*, int *noWarnings*)
- longdouble `getParameterValue` (pulsar **psr*, int *param*, int *arr*)

12.162.1 Function Documentation

12.162.1.1 void `copyParam` (parameter *p1*, parameter * *p2*)

12.162.1.2 void `copyPSR` (pulsar * *p*, int *p1*, int *p2*)

12.162.1.3 void `displayMsg` (int *type*, const char * *key*, const char * *searchStr*, const char * *variableStr*, int *noWarnings*)

12.162.1.4 double `dms_turn` (char * *line*)

12.162.1.5 double `dotproduct` (double * *v1*, double * *v2*)

12.162.1.6 void `equ2ecl` (double * *x*)

12.162.1.7 longdouble `fortran_mod` (longdouble *a*, longdouble *p*)

12.162.1.8 double `fortran_mod` (double *a*, double *p*)

12.162.1.9 int `fortran_nint` (double *x*)

12.162.1.10 long `fortran_nlong` (longdouble *x*)

12.162.1.11 longdouble `getParameterValue` (pulsar * *psr*, int *param*, int *arr*)

12.162.1.12 double `hms_turn` (char * *line*)

12.162.1.13 double `turn_deg` (double *turn*)

12.162.1.14 int `turn_dms` (double *turn*, char * *dms*)

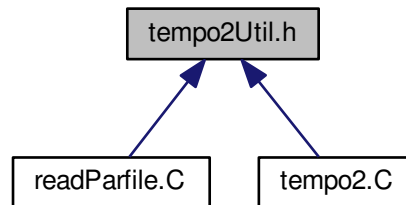
12.162.1.15 int `turn_hms` (double *turn*, char * *hms*)

12.162.1.16 void `vectorscale` (double * *v*, double *k*)

12.162.1.17 void `vectorsum` (double * *res*, double * *v1*, double * *v2*)

12.163 tempo2Util.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- double [turn_deg](#) (double *turn*)
- double [dms_turn](#) (char **line*)
- double [hms_turn](#) (char **line*)

12.163.1 Function Documentation

12.163.1.1 double [dms_turn](#) (char * *line*)

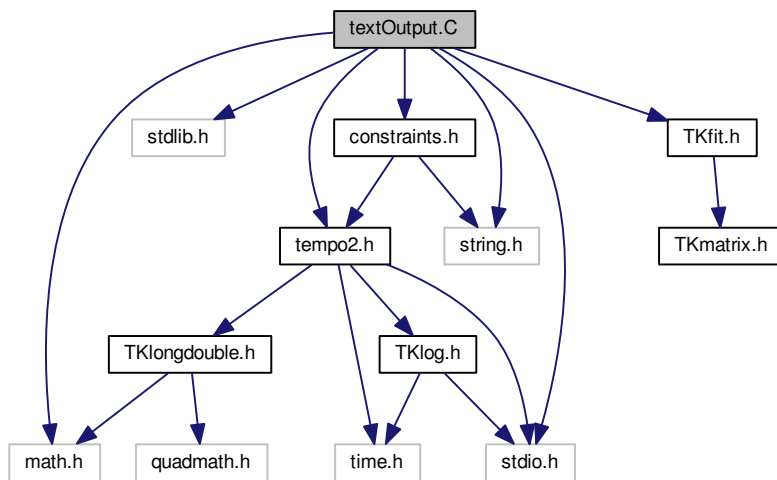
12.163.1.2 double [hms_turn](#) (char * *line*)

12.163.1.3 double [turn_deg](#) (double *turn*)

12.164 textOutput.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "tempo2.h"
#include "constraints.h"
#include "TKfit.h"
```

Include dependency graph for textOutput.C:



Functions

- double [m2](#) ([longdouble](#) mf, [longdouble](#) sini, [longdouble](#) m1)
- void [printGlitch](#) ([pulsar](#) psr)
- double [dglep](#) ([pulsar](#) psr, int gn, double fph)
- void [textOutput](#) ([pulsar](#) *psr, int npsr, double globalParameter, int nGlobal, int outRes, int newpar, const char *fname)
- double [calcRMS](#) ([pulsar](#) *psr, int p)

12.164.1 Function Documentation

12.164.1.1 double [calcRMS](#) ([pulsar](#) * *psr*, int *p*)

12.164.1.2 double [dglep](#) ([pulsar](#) *psr*, int *gn*, double *fph*)

12.164.1.3 double [m2](#) ([longdouble](#) *mf*, [longdouble](#) *sini*, [longdouble](#) *m1*)

12.164.1.4 void [printGlitch](#) ([pulsar](#) *psr*)

12.164.1.5 void [textOutput](#) ([pulsar](#) * *psr*, int *npsr*, double *globalParameter*, int *nGlobal*, int *outRes*, int *newpar*, const char * *fname*)

```
sqrt(psr[0].covar[iii][ii]*psr[0].covar[jjj][jj]);
```

12.165 TKcholesky.h File Reference

Functions

- void [cholesky_readFromCovarianceFunction](#) (double **m, const char *fname, double *resx, double *resy, double *rese, int np, int nc)

- void [cholesky_covarFunc2matrix](#) (double **m, double *covarFunc, int ndays, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_powerlawModel](#) (double **m, double modelAlpha, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_powerlawModel_withBeta](#) (double **m, double modelAlpha, double beta, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)
- int [cholesky_formUinv](#) (double **uinv, double **m, int np)
- void [cholesky_dmModel](#) (double **m, double D, double d, double ref_freq, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_ecm](#) (double **m, char *fileName, double *resx, double *resy, double *rese, int np, int nc)
- void [cholesky_dmModelCovarParam](#) (double **m, double [alpha](#), double a, double b, double *resx, double *resy, double *rese, int np, int nc)

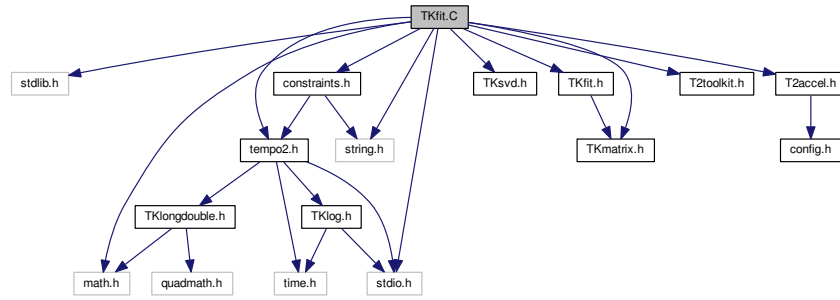
12.165.1 Function Documentation

- 12.165.1.1 void [cholesky_covarFunc2matrix](#) (double ** *m*, double * *covarFunc*, int *ndays*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.165.1.2 void [cholesky_dmModel](#) (double ** *m*, double *D*, double *d*, double *ref_freq*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.165.1.3 void [cholesky_dmModelCovarParam](#) (double ** *m*, double *alpha*, double *a*, double *b*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.165.1.4 void [cholesky_ecm](#) (double ** *m*, char * *fileName*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.165.1.5 int [cholesky_formUinv](#) (double ** *uinv*, double ** *m*, int *np*)
- UINV is a lower triangular matrix. Matrices are row-major order, i.e. uinv[r][c]. returns 0 if ok.
- 12.165.1.6 void [cholesky_powerlawModel](#) (double ** *m*, double *modelAlpha*, double *modelFc*, double *modelA*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.165.1.7 void [cholesky_powerlawModel_withBeta](#) (double ** *m*, double *modelAlpha*, double *beta*, double *modelFc*, double *modelA*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 12.165.1.8 void [cholesky_readFromCovarianceFunction](#) (double ** *m*, const char * *fname*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)

12.166 TKfit.C File Reference

```
#include <stdlib.h>
#include <math.h>
#include <stdio.h>
#include <string.h>
#include "tempo2.h"
#include "constraints.h"
#include "TKsvd.h"
#include "TKmatrix.h"
#include "T2toolkit.h"
#include "T2accel.h"
#include "TKfit.h"
```

Include dependency graph for TKfit.C:



Functions

- void [TKremovePoly_f](#) (float *px, float *py, int n, int m)
- void [TKremovePoly_d](#) (double *x, double *y, int n, int m)
- void [TKfindPoly_d](#) (double *x, double *y, int n, int m, double *p)
- void [TKfitPoly](#) (double x, double *v, int m)
- double [TKleastSquares](#) (double *b, double *white_b, double **designMatrix, double **white_designMatrix, int n, int [nf](#), double tol, char rescale_errors, double *outP, double *e, double **cvm)
- double [TKconstrainedLeastSquares](#) (double *b, double *white_b, double **designMatrix, double **white_designMatrix, double **constraintsMatrix, int n, int [nf](#), int nconstraints, double tol, char rescale_errors, double *outP, double *e, double **cvm)
- double [TKrobustLeastSquares](#) (double *b, double *white_b, double **designMatrix, double **white_designMatrix, int n, int [nf](#), double tol, char rescale_errors, double *outP, double *e, double **cvm, char robust)
- double [TKrobustConstrainedLeastSquares](#) (double *data, double *white_data, double **designMatrix, double **white_designMatrix, double **constraintsMatrix, int ndata, int nparams, int nconstraints, double tol, char rescale_errors, double *outP, double *e, double **Ocv, char robust)
- void [TKleastSquares_single_pulsar](#) (double *x, double *y, int n, double *outP, double *e, int [nf](#), double **cvm, double *chisq, void(*fitFuncs)(double, double[], int, [pulsar](#) *, int, int), [pulsar](#) *psr, double tol, int *ip, char rescale_errors, double **uinv)
- void [TKleastSquares_global_pulsar](#) (double **x, double **y, int *n, double *outP, double *e, int [nf](#), int nglobal, double **cvm, double *chisq, void(*fitFuncs)(double, double[], int, [pulsar](#) *, int, int), [pulsar](#) *psr, double tol, int **ip, char rescale_errors, double **uinv, int npsr)
- void [TKfit_getPulsarDesignMatrix](#) (double *x, double *y, int n, int [nf](#), void(*fitFuncs)(double, double[], int, [pulsar](#) *, int, int), [pulsar](#) *psr, int *ip, double **uinv, int ipsr, double ***OUT_designMatrix, double ***OUT_white_designMatrix, double **OUT_b, double **OUT_wb)
- void [TKleastSquares_svd_psr_dcm](#) (double *x, double *y, double *sig, int n, double *outP, double *e, int [nf](#), double **cvm, double *chisq, void(*fitFuncs)(double, double[], int, [pulsar](#) *, int, int), int weight, [pulsar](#) *psr, double tol, int *ip, double **uinv)
- void [TKleastSquares_svd_psr](#) (double *x, double *y, double *sig, int n, double *p, double *e, int [nf](#), double **cvm, double *chisq, void(*fitFuncs)(double, double[], int, [pulsar](#) *, int, int), int weight, [pulsar](#) *psr, double tol, int *ip)
- void [TKleastSquares_svd_noErr](#) (double *x, double *y, int n, double *p, int [nf](#), void(*fitFuncs)(double, double[], int))
- void [TKleastSquares_svd](#) (double *x, double *y, double *sig, int n, double *p, double *e, int [nf](#), double **cvm, double *chisq, void(*fitFuncs)(double, double[], int), int weight)
- void [TKleastSquares_svd_passN](#) (double *x, double *y, double *sig2, int n, double *p, double *e, int [nf](#), double **cvm, double *chisq, void(*fitFuncs)(double, double[], int, int), int weight)
- [longdouble](#) [TKfindMax_Ld](#) ([longdouble](#) *x, int n)

12.166.1 Function Documentation

12.166.1.1 `double TKconstrainedLeastSquares (double * b, double * white_b, double ** designMatrix, double ** white_designMatrix, double ** constraintsMatrix, int n, int nf, int nconstraints, double tol, char rescale_errors, double * outP, double * e, double ** cvm)`

12.166.1.2 `longdouble TKfindMax_Ld (longdouble * x, int n)`

12.166.1.3 `void TKfindPoly_d (double * x, double * y, int n, int m, double * p)`

12.166.1.4 `void TKfit_getPulsarDesignMatrix (double * x, double * y, int n, int nf, void(*) (double, double[], int, pulsar *, int, int) fitFuncs, pulsar * psr, int * ip, double ** uinv, int ipsr, double *** OUT_designMatrix, double *** OUT_white_designMatrix, double ** OUT_b, double ** OUT_wb)`

12.166.1.5 `void TKfitPoly (double x, double * v, int m)`

12.166.1.6 `double TKleastSquares (double * b, double * white_b, double ** designMatrix, double ** white_designMatrix, int n, int nf, double tol, char rescale_errors, double * outP, double * e, double ** cvm)`

TKleastSquares performs a least squares fit.

double* *b*: Array of Y values. double* *white_b*: Array of whitened Y values. (Uinv.Y) double** *designMatrix*: Fit matrix double** *white_designMatrix*: Whitened fit matrix int *n*: size of "b" int *nf*: number of fit parameters (i.e. columns of *designMatrix*) double *tol*: filter to remove small values of the SVD char *rescale_errors*: boolean to say if resultant errors should be scaled by *chisq* double* *outP*: output fit parameters double* *e*: output error in fit parameters double ***cvm*: *nf***nf* output covariance matrix for fit parameters.

12.166.1.7 `void TKleastSquares_global_pulsar (double ** x, double ** y, int * n, double * outP, double * e, int * nf, int nglobal, double ** cvm, double * chisq, void(*) (double, double[], int, pulsar *, int, int) fitFuncs, pulsar * psr, double tol, int ** ip, char rescale_errors, double *** uinv, int npsr)`

12.166.1.8 `void TKleastSquares_single_pulsar (double * x, double * y, int n, double * outP, double * e, int nf, double ** cvm, double * chisq, void(*) (double, double[], int, pulsar *, int, int) fitFuncs, pulsar * psr, double tol, int * ip, char rescale_errors, double ** uinv)`

12.166.1.9 `void TKleastSquares_svd (double * x, double * y, double * sig, int n, double * p, double * e, int nf, double ** cvm, double * chisq, void(*) (double, double[], int) fitFuncs, int weight)`

12.166.1.10 `void TKleastSquares_svd_noErr (double * x, double * y, int n, double * p, int nf, void(*) (double, double[], int) fitFuncs)`

12.166.1.11 `void TKleastSquares_svd_passN (double * x, double * y, double * sig2, int n, double * p, double * e, int nf, double ** cvm, double * chisq, void(*) (double, double[], int, int) fitFuncs, int weight)`

12.166.1.12 `void TKleastSquares_svd_psr (double * x, double * y, double * sig, int n, double * p, double * e, int nf, double ** cvm, double * chisq, void(*) (double, double[], int, pulsar *, int, int) fitFuncs, int weight, pulsar * psr, double tol, int * ip)`

12.166.1.13 `void TKleastSquares_svd_psr_dcm (double * x, double * y, double * sig, int n, double * outP, double * e, int nf, double ** cvm, double * chisq, void(*) (double, double[], int, pulsar *, int, int) fitFuncs, int weight, pulsar * psr, double tol, int * ip, double ** uinv)`

12.166.1.14 `void TKremovePoly_d (double * x, double * y, int n, int m)`

12.166.1.15 `void TKremovePoly_f (float * px, float * py, int n, int m)`

```
12.166.1.16 double TKrobustConstrainedLeastSquares ( double * data, double * white_data, double ** designMatrix, double
** white_designMatrix, double ** constraintsMatrix, int ndata, int nparams, int nconstraints, double tol, char
rescale_errors, double * outP, double * e, double ** OcvM, char robust )
```

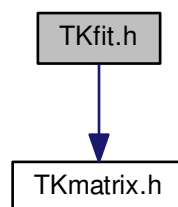
Robust Estimator code by Wang YiDi, Univ. Manchester 2015

```
12.166.1.17 double TKrobustLeastSquares ( double * b, double * white_b, double ** designMatrix, double **
white_designMatrix, int n, int nf, double tol, char rescale_errors, double * outP, double * e, double ** cvm, char
robust )
```

12.167 TKfit.h File Reference

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

Include dependency graph for TKfit.h:



This graph shows which files directly or indirectly include this file:



Functions

- double [TKleastSquares](#) (double *b, double *white_b, double **designMatrix, double **white_designMatrix, int n, int [nf](#), double tol, char rescale_errors, double *outP, double *e, double **CVM)
- double [TKrobustLeastSquares](#) (double *b, double *white_b, double **designMatrix, double **white_↵ designMatrix, int n, int [nf](#), double tol, char rescale_errors, double *outP, double *e, double **cvm, char robust)
- double [TKconstrainedLeastSquares](#) (double *b, double *white_b, double **designMatrix, double **white_↵ designMatrix, double **constraintsMatrix, int n, int [nf](#), int nconstraints, double tol, char rescale_errors, double *outP, double *e, double **cvm)
- double [TKrobustConstrainedLeastSquares](#) (double *b, double *white_b, double **designMatrix, double **white_designMatrix, double **constraintsMatrix, int n, int [nf](#), int nconstraints, double tol, char rescale_↵ _errors, double *outP, double *e, double **cvm, char robust)
- void [TKleastSquares_svd](#) (double *x, double *y, double *sig, int n, double *p, double *e, int [nf](#), double **cvm, double *chisq, void(*fitFuncs)(double, double[], int), int weight)
- void [TKleastSquares_svd_noErr](#) (double *x, double *y, int n, double *p, int [nf](#), void(*fitFuncs)(double, double[], int))
- void [TKremovePoly_f](#) (float *px, float *py, int n, int m)
- void [TKremovePoly_d](#) (double *px, double *py, int n, int m)

- void [TKfindPoly_d](#) (double *px, double *py, int n, int m, double *p)
- void [TKfitPoly](#) (double x, double *v, int m)

12.167.1 Function Documentation

12.167.1.1 double TKconstrainedLeastSquares (double * *b*, double * *white_b*, double ** *designMatrix*, double ** *white_designMatrix*, double ** *constraintsMatrix*, int *n*, int *nf*, int *nconstraints*, double *tol*, char *rescale_errors*, double * *outP*, double * *e*, double ** *cvm*)

12.167.1.2 void TKfindPoly_d (double * *px*, double * *py*, int *n*, int *m*, double * *p*)

12.167.1.3 void TKfitPoly (double *x*, double * *v*, int *m*)

12.167.1.4 double TKleastSquares (double * *b*, double * *white_b*, double ** *designMatrix*, double ** *white_designMatrix*, int *n*, int *nf*, double *tol*, char *rescale_errors*, double * *outP*, double * *e*, double ** *cvm*)

TKleastSquares performs a least squares fit.

double* *b*: Array of Y values. double* *white_b*: Array of whitened Y values. (Uinv.Y) double** *designMatrix*: Fit matrix double** *white_designMatrix*: Whitened fit matrix int *n*: size of "b" int *nf*: number of fit parameters (i.e. columns of *designMatrix*) double *tol*: filter to remove small values of the SVD char *rescale_errors*: boolean to say if resultant errors should be scaled by *chisq* double* *outP*: output fit parameters double* *e*: output error in fit parameters double ***cvm*: *nf***nf* output covariance matrix for fit parameters.

12.167.1.5 void TKleastSquares_svd (double * *x*, double * *y*, double * *sig*, int *n*, double * *p*, double * *e*, int *nf*, double ** *cvm*, double * *chisq*, void(*) (double, double[], int) *fitFuncs*, int *weight*)

12.167.1.6 void TKleastSquares_svd_noErr (double * *x*, double * *y*, int *n*, double * *p*, int *nf*, void(*) (double, double[], int) *fitFuncs*)

12.167.1.7 void TKremovePoly_d (double * *px*, double * *py*, int *n*, int *m*)

12.167.1.8 void TKremovePoly_f (float * *px*, float * *py*, int *n*, int *m*)

12.167.1.9 double TKrobustConstrainedLeastSquares (double * *b*, double * *white_b*, double ** *designMatrix*, double ** *white_designMatrix*, double ** *constraintsMatrix*, int *n*, int *nf*, int *nconstraints*, double *tol*, char *rescale_errors*, double * *outP*, double * *e*, double ** *cvm*, char *robust*)

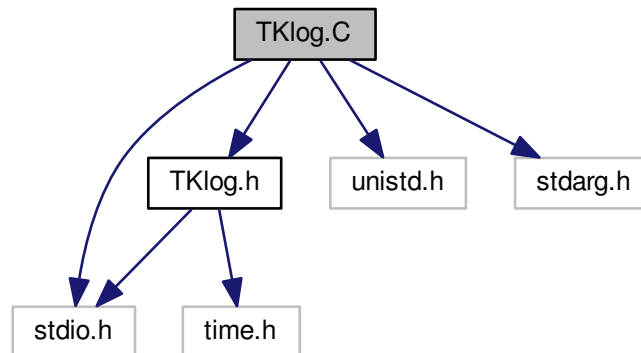
Robust Estimator code by Wang YiDi, Univ. Manchester 2015

12.167.1.10 double TKrobustLeastSquares (double * *b*, double * *white_b*, double ** *designMatrix*, double ** *white_designMatrix*, int *n*, int *nf*, double *tol*, char *rescale_errors*, double * *outP*, double * *e*, double ** *cvm*, char *robust*)

12.168 TKlog.C File Reference

```
#include "TKlog.h"
#include <unistd.h>
#include <stdarg.h>
#include <stdio.h>
```

Include dependency graph for TKlog.C:



Functions

- int [logerr_check](#) ()
- void [_TKchklog](#) (FILE *out, const char *fmt,...)

Variables

- int [debugFlag](#) = 0
- unsigned [TK_errorCount](#) = 0
- unsigned [TK_warnCount](#) = 0
- int [writeResiduals](#) =0
- int [tcheck](#) = 0
- clock_t [timer_clk](#) = 0
- char [TK_errorlog](#) [TK_MAX_ERRORS][TK_MAX_ERROR_LEN]
- char [TK_warnlog](#) [TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

12.168.1 Function Documentation

12.168.1.1 void [_TKchklog](#) (FILE * *out*, const char * *fmt*, ...)

12.168.1.2 int [logerr_check](#) ()

12.168.2 Variable Documentation

12.168.2.1 int [debugFlag](#) = 0

12.168.2.2 int [tcheck](#) = 0

12.168.2.3 clock_t [timer_clk](#) = 0

12.168.2.4 unsigned [TK_errorCount](#) = 0

12.168.2.5 `char TK_errorlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]`

12.168.2.6 `unsigned TK_warnCount = 0`

12.168.2.7 `char TK_warnlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]`

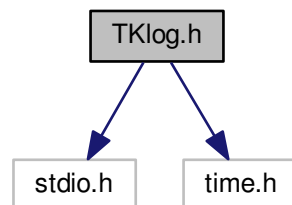
12.168.2.8 `int writeResiduals = 0`

12.169 TKlog.h File Reference

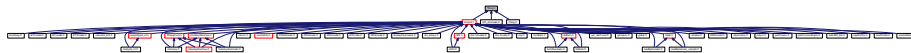
```
#include <stdio.h>
```

```
#include <time.h>
```

Include dependency graph for TKlog.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define TK_MAX_ERRORS 16`
- `#define TK_MAX_ERROR_LEN 128`
- `#define LOG_OUTFILE stdout`
- `#define RESETCOLOR "\033[0m"`
- `#define WARNCOLOR RESETCOLOR "\033[0;35m"`
- `#define BOLDCOLOR RESETCOLOR "\033[1m"`
- `#define ERRORCOLOR RESETCOLOR "\033[1;31m"`
- `#define WHERESTR "[%s:%d] "`
- `#define WHEREARG __FILE__, __LINE__`
- `#define ENDL "\n"`
- `#define WHEREERR ERRORCOLOR "***ERROR***\n [%s:%d] " RESETCOLOR`
- `#define WHEREWARN BOLDCOLOR "[%s:%d] " WARNCOLOR "Warning: " RESETCOLOR`
- `#define ENDERR "\n***!!!!***"`
- `#define WHERECHK "[%s:%d] T=%.2f s: "`
- `#define _LOG(_fmt, ...) _TKchklog(LOG_OUTFILE, _fmt, ## __VA_ARGS__)`
- `#define logmsg(_fmt, ...) _LOG(WHERESTR _fmt ENDL, WHEREARG, ## __VA_ARGS__)`
- `#define logdbg(_fmt, ...) if(debugFlag)logmsg(_fmt, ## __VA_ARGS__)`

- `#define logerr(_fmt, ...) do{TK_STORE_ERROR(_fmt,##__VA_ARGS__); _LOG(WHEREERR _fmt ENDE↵
RR ENDL, WHEREARG,##__VA_ARGS__);while(0)`
- `#define logwarn(_fmt, ...) do{TK_STORE_WARNING(_fmt,##__VA_ARGS__); _LOG(WHEREWARN _fmt
ENDL, WHEREARG,##__VA_ARGS__);while(0)`
- `#define logtchk(_fmt, ...) if(tcheck) _LOG(WHERETCHK _fmt ENDL, WHEREARG,(clock()-timer_↵
clk)/(float)CLOCKS_PER_SEC,##__VA_ARGS__)`
- `#define TK_STORE_ERROR(_fmt, ...) if(TK_errorCount < TK_MAX_ERRORS)sprintf(TK_errorlog[TK_↵
errorCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_errorCount`
- `#define TK_STORE_WARNING(_fmt, ...) if(TK_warnCount < TK_MAX_ERRORS)sprintf(TK_warnlog[T↵
K_warnCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__); ++TK_warnCount`
- `#define DEPRECATED`

Functions

- `int logerr_check ()`
- `void _TKchklog (FILE *, const char *,...)`

Variables

- `int debugFlag`
- `int writeResiduals`
- `int tcheck`
- `clock_t timer_clk`
- `unsigned TK_errorCount`
- `unsigned TK_warnCount`
- `char TK_errorlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]`
- `char TK_warnlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]`

12.169.1 Macro Definition Documentation

12.169.1.1 `#define _LOG(_fmt, ...) _TKchklog(LOG_OUTFILE,_fmt,##__VA_ARGS__)`

12.169.1.2 `#define BOLDCOLOR RESETCOLOR "\033[1m"`

12.169.1.3 `#define DEPRECATED`

12.169.1.4 `#define ENDERR "\n***!!!!***"`

12.169.1.5 `#define ENDL "\n"`

12.169.1.6 `#define ERRORCOLOR RESETCOLOR "\033[1;31m"`

12.169.1.7 `#define LOG_OUTFILE stdout`

12.169.1.8 `#define logdbg(_fmt, ...) if(debugFlag)logmsg(_fmt,##__VA_ARGS__)`

12.169.1.9 `#define logerr(_fmt, ...) do{TK_STORE_ERROR(_fmt,##__VA_ARGS__); _LOG(WHEREERR _fmt
ENDERR ENDL, WHEREARG,##__VA_ARGS__);while(0)`

12.169.1.10 `#define logmsg(_fmt, ...) _LOG(WHERESTR _fmt ENDL, WHEREARG,##__VA_ARGS__)`

12.169.1.11 `#define logtchk(_fmt, ...) if(tcheck) _LOG(WHERETCHK _fmt ENDL,
WHEREARG,(clock()-timer_clk)/(float)CLOCKS_PER_SEC,##__VA_ARGS__)`

```

12.169.1.12 #define logwarn( _fmt, ... ) do{TK_STORE_WARNING(_fmt,##__VA_ARGS__); _LOG(WHEREWARN _fmt
            ENDL, WHEREARG,##__VA_ARGS__);while(0)

12.169.1.13 #define RESETCOLOR "\033[0m"

12.169.1.14 #define TK_MAX_ERROR_LEN 128

12.169.1.15 #define TK_MAX_ERRORS 16

12.169.1.16 #define TK_STORE_ERROR( _fmt, ... ) if(TK_errorCount < TK_MAX_ERROR↵
            S)snprintf(TK_errorlog[TK_errorCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__);
            ++TK_errorCount

12.169.1.17 #define TK_STORE_WARNING( _fmt, ... ) if(TK_warnCount < TK_MAX_ERROR↵
            S)snprintf(TK_warnlog[TK_warnCount],TK_MAX_ERROR_LEN, _fmt,##__VA_ARGS__);
            ++TK_warnCount

12.169.1.18 #define WARNCOLOR RESETCOLOR "\033[0;35m"

12.169.1.19 #define WHEREARG __FILE__, __LINE__

12.169.1.20 #define WHEREERR ERRORCOLOR "***ERROR***\n [%s:%d] " RESETCOLOR

12.169.1.21 #define WHERESTR "[%s:%d] "

12.169.1.22 #define WHERECHK "[%s:%d] T=%.2f s: "

12.169.1.23 #define WHEREWARN BOLDCOLOR "[%s:%d] " WARNCOLOR "Warning: " RESETCOLOR

```

12.169.2 Function Documentation

```

12.169.2.1 void _TKchklog ( FILE *, const char *, ... )

12.169.2.2 int logerr_check ( )

```

12.169.3 Variable Documentation

```

12.169.3.1 int debugFlag

12.169.3.2 int tcheck

12.169.3.3 clock_t timer_clk

12.169.3.4 unsigned TK_errorCount

12.169.3.5 char TK_errorlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

12.169.3.6 unsigned TK_warnCount

12.169.3.7 char TK_warnlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

12.169.3.8 int writeResiduals

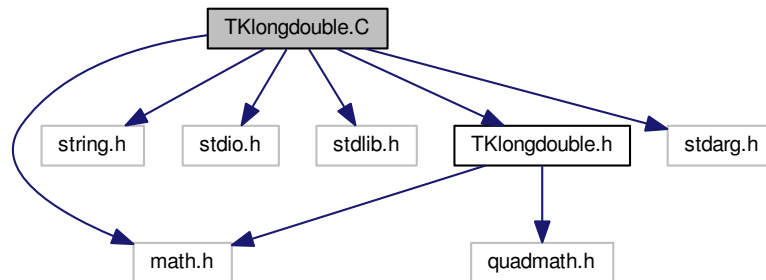
```

12.170 TKlongdouble.C File Reference

```
#include <math.h>
```

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "TKlongdouble.h"
#include <stdarg.h>
```

Include dependency graph for TKlongdouble.C:



Macros

- `#define BUFSIZE 4096`

Functions

- `std::string print_longdouble` (const `longdouble` &ld)
- `longdouble parse_longdouble` (const char *str)
- `int ld_vsprintf` (char *buf, const char *__format, va_list args)
- `int ld_fprintf` (FILE *__stream, const char *__format,...)
- `int ld_printf` (const char *__format,...)
- `int ld_sprintf` (char *buf, const char *__format,...)

12.170.1 Macro Definition Documentation

12.170.1.1 `#define BUFSIZE 4096`

12.170.2 Function Documentation

12.170.2.1 `int ld_fprintf (FILE * __stream, const char * __format, ...)`

12.170.2.2 `int ld_printf (const char * __format, ...)`

12.170.2.3 `int ld_sprintf (char * buf, const char * __format, ...)`

12.170.2.4 `int ld_vsprintf (char * buf, const char * __format, va_list args)`

12.170.2.5 `longdouble parse_longdouble (const char * str)`

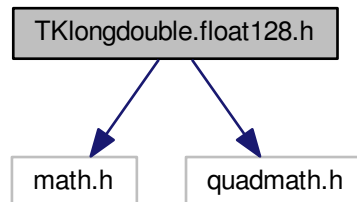
12.170.2.6 `std::string print_longdouble (const longdouble & ld)`

12.171 TKlongdouble.float128.h File Reference

```
#include <math.h>
```

```
#include <quadmath.h>
```

Include dependency graph for TKlongdouble.float128.h:



Macros

- `#define USE_BUILTIN_LONGDOUBLE`
- `#define LONGDOUBLE_IS_FLOAT128`
- `#define LONGDOUBLE_ONE 1.0Q`
- `#define longdouble(a) a##Q`
- `#define FMT_LD "Q"`
- `#define LD_PI M_PiQ`
- `#define cosl cosq`
- `#define sinl sinq`
- `#define floorl floorq`
- `#define fabsl fabsq`

Typedefs

- `typedef __float128 longdouble`

Functions

- `longdouble parse_longdouble` (const char *str)
- `int ld_printf` (const char *__format,...)
- `int ld_fprintf` (FILE *__stream, const char *__format,...)
- `int ld_sprintf` (char *__str, const char *__format,...)

12.171.1 Macro Definition Documentation

12.171.1.1 `#define cosl cosq`

12.171.1.2 `#define fabsl fabsq`

12.171.1.3 `#define floorl floorq`

12.171.1.4 `#define FMT_LD "Q"`

12.171.1.5 `#define LD_PI M_Pi`

12.171.1.6 `#define longdouble(a) a##Q`

12.171.1.7 `#define LONGDOUBLE_IS_FLOAT128`

12.171.1.8 `#define LONGDOUBLE_ONE 1.0Q`

12.171.1.9 `#define sinl sinq`

12.171.1.10 `#define USE_BUILTIN_LONGDOUBLE`

12.171.2 Typedef Documentation

12.171.2.1 `typedef __float128 longdouble`

12.171.3 Function Documentation

12.171.3.1 `int ld_fprintf (FILE * __stream, const char * __format, ...)`

12.171.3.2 `int ld_printf (const char * __format, ...)`

12.171.3.3 `int ld_sprintf (char * __str, const char * __format, ...)`

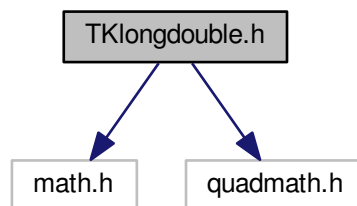
12.171.3.4 `longdouble parse_longdouble (const char * str)`

12.172 TKlongdouble.h File Reference

```
#include <math.h>
```

```
#include <quadmath.h>
```

Include dependency graph for TKlongdouble.h:



This graph shows which files directly or indirectly include this file:



Macros

- `#define USE_BUILTIN_LONGDOUBLE`
- `#define LONGDOUBLE_IS_FLOAT128`
- `#define LONGDOUBLE_ONE 1.0Q`
- `#define longdouble(a) a##Q`
- `#define FMT_LD "Q"`
- `#define LD_PI M_PiQ`
- `#define cosl cosq`
- `#define sinl sinq`
- `#define floorl floorq`
- `#define fabsl fabsq`

Typedefs

- `typedef __float128 longdouble`

Functions

- `longdouble parse_longdouble` (const char *str)
- `int ld_printf` (const char *__format,...)
- `int ld_fprintf` (FILE *__stream, const char *__format,...)
- `int ld_sprintf` (char *__str, const char *__format,...)

12.172.1 Macro Definition Documentation

12.172.1.1 `#define cosl cosq`

12.172.1.2 `#define fabsl fabsq`

12.172.1.3 `#define floorl floorq`

12.172.1.4 `#define FMT_LD "Q"`

12.172.1.5 `#define LD_PI M_PiQ`

12.172.1.6 `#define longdouble(a) a##Q`

12.172.1.7 `#define LONGDOUBLE_IS_FLOAT128`

12.172.1.8 `#define LONGDOUBLE_ONE 1.0Q`

12.172.1.9 `#define sinl sinq`

12.172.1.10 `#define USE_BUILTIN_LONGDOUBLE`

12.172.2 Typedef Documentation

12.172.2.1 `typedef __float128 longdouble`

12.172.3 Function Documentation

12.172.3.1 `int ld_fprintf (FILE * __stream, const char * __format, ...)`

12.172.3.2 `int ld_printf (const char * __format, ...)`

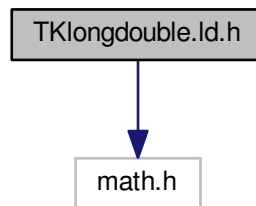
12.172.3.3 `int ld_sprintf (char * __str, const char * __format, ...)`

12.172.3.4 `longdouble parse_longdouble (const char * str)`

12.173 TKlongdouble.Id.h File Reference

```
#include <math.h>
```

Include dependency graph for TKlongdouble.Id.h:



Macros

- `#define USE_BUILTIN_LONGDOUBLE`
- `#define longdouble(a) a##L`
- `#define LD_PI M_PI`
- `#define LONGDOUBLE_IS_IEEE754`
- `#define LONGDOUBLE_ONE 1.0L`
- `#define ld_printf printf`
- `#define ld_fprintf fprintf`
- `#define ld_sprintf sprintf`

Typedefs

- `typedef long double longdouble`

Functions

- `longdouble parse_longdouble (const char *str)`

12.173.1 Macro Definition Documentation

12.173.1.1 `#define ld_fprintf fprintf`

12.173.1.2 `#define LD_PI M_PI`

12.173.1.3 `#define ld_printf printf`

12.173.1.4 `#define Id_sprintf sprintf`

12.173.1.5 `#define longdouble(a) a##L`

12.173.1.6 `#define LONGDOUBLE_IS_IEEE754`

12.173.1.7 `#define LONGDOUBLE_ONE 1.0L`

12.173.1.8 `#define USE_BUILTIN_LONGDOUBLE`

12.173.2 Typedef Documentation

12.173.2.1 `typedef long double longdouble`

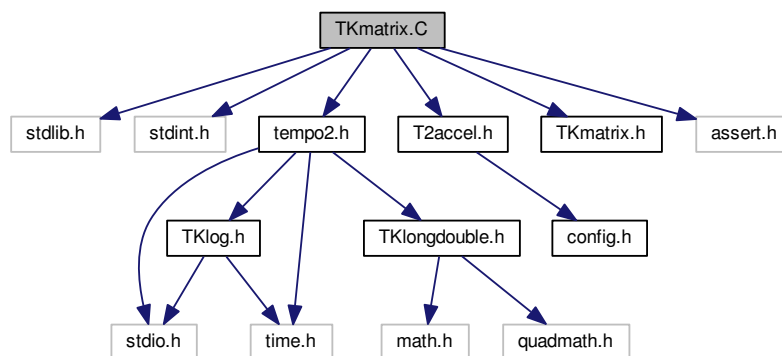
12.173.3 Function Documentation

12.173.3.1 `longdouble parse_longdouble (const char * str)`

12.174 TKmatrix.C File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include "tempo2.h"
#include "T2accel.h"
#include "TKmatrix.h"
#include <assert.h>
```

Include dependency graph for TKmatrix.C:



Functions

- void [TKmultMatrix](#) (double **idcm, double **u, int ndata, int ndata2, int npol, double **uout)
- void [TKmultMatrix_sq](#) (double **idcm, double **u, int ndata, int npol, double **uout)
- void [TKmultMatrixVec](#) (double **idcm, double *b, int ndata, int ndata2, double *bout)
- void [TKmultMatrixVec_sq](#) (double **idcm, double *b, int ndata, double *bout)
- [longdouble ** malloc_2dLL](#) (int rows, int cols)
- void [free_2dLL](#) (longdouble **m)
- double ** [malloc_uinv](#) (int n)
- int [get_blas_rows](#) (double **uinv)

- int [get_blas_cols](#) (double **uinv)
- double ** [malloc_blas](#) (int rows, int cols)
- void [free_blas](#) (double **m)
- void [free_uinv](#) (double **uinv)
- float ** [malloc_2df](#) (int rows, int cols)
- void [free_2df](#) (float **m)

12.174.1 Function Documentation

12.174.1.1 void [free_2df](#) (float ** *m*)

12.174.1.2 void [free_2dLL](#) (longdouble ** *m*)

12.174.1.3 void [free_blas](#) (double ** *m*)

12.174.1.4 void [free_uinv](#) (double ** *uinv*)

12.174.1.5 int [get_blas_cols](#) (double ** *uinv*)

12.174.1.6 int [get_blas_rows](#) (double ** *uinv*)

12.174.1.7 float** [malloc_2df](#) (int *rows*, int *cols*)

12.174.1.8 longdouble** [malloc_2dLL](#) (int *rows*, int *cols*)

12.174.1.9 double** [malloc_blas](#) (int *rows*, int *cols*)

Allocate uinv in a "BLAS/LAPACK" compatible way store the dimensions of the array in two secret elements before the main memory allocation. Useful for checks. WARNING: assumes that sizeof(int) <= sizeof(double)

12.174.1.10 double** [malloc_uinv](#) (int *n*)

Allocate uinv in a "BLAS/LAPACK" compatible way

12.174.1.11 void [TKmultMatrix](#) (double ** *idcm*, double ** *u*, int *ndata*, int *ndata2*, int *npol*, double ** *uout*)

12.174.1.12 void [TKmultMatrix_sq](#) (double ** *idcm*, double ** *u*, int *ndata*, int *npol*, double ** *uout*)

12.174.1.13 void [TKmultMatrixVec](#) (double ** *idcm*, double * *b*, int *ndata*, int *ndata2*, double * *bout*)

12.174.1.14 void [TKmultMatrixVec_sq](#) (double ** *idcm*, double * *b*, int *ndata*, double * *bout*)

12.175 TKmatrix.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [TKmultMatrix_sq](#) (double **idcm, double **u, int ndata, int npol, double **uout)
- void [TKmultMatrixVec_sq](#) (double **idcm, double *b, int ndata, double *bout)
- void [TKmultMatrix](#) (double **idcm, double **u, int ndata, int ndata2, int npol, double **uout)
- void [TKmultMatrixVec](#) (double **idcm, double *b, int ndata, int ndata2, double *bout)
- double ** [malloc_uinv](#) (int n)
- double ** [malloc_blas](#) (int n, int m)
- void [free_blas](#) (double **matrix)
- void [free_uinv](#) (double **uinv)
- int [get_blas_rows](#) (double **uinv)
- int [get_blas_cols](#) (double **uinv)
- float ** [malloc_2df](#) (int rows, int cols)
- void [free_2df](#) (float **uinv)

12.175.1 Function Documentation

12.175.1.1 void [free_2df](#) (float ** *uinv*)

12.175.1.2 void [free_blas](#) (double ** *matrix*)

12.175.1.3 void [free_uinv](#) (double ** *uinv*)

12.175.1.4 int [get_blas_cols](#) (double ** *uinv*)

12.175.1.5 int [get_blas_rows](#) (double ** *uinv*)

12.175.1.6 float** [malloc_2df](#) (int *rows*, int *cols*)

12.175.1.7 double** [malloc_blas](#) (int *rows*, int *cols*)

Allocate uinv in a "BLAS/LAPACK" compatible way store the dimensions of the array in two secret elements before the main memory allocation. Useful for checks. WARNING: assumes that sizeof(int) <= sizeof(double)

12.175.1.8 double** [malloc_uinv](#) (int *n*)

Allocate uinv in a "BLAS/LAPACK" compatible way

12.175.1.9 void [TKmultMatrix](#) (double ** *idcm*, double ** *u*, int *ndata*, int *ndata2*, int *npol*, double ** *uout*)

12.175.1.10 void [TKmultMatrix_sq](#) (double ** *idcm*, double ** *u*, int *ndata*, int *npol*, double ** *uout*)

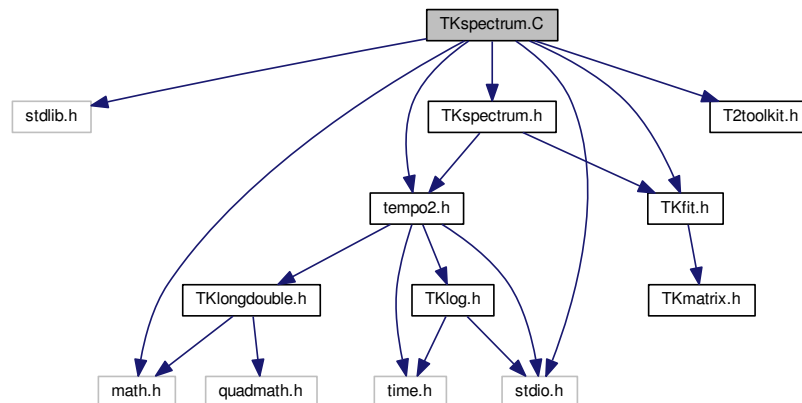
12.175.1.11 void [TKmultMatrixVec](#) (double ** *idcm*, double * *b*, int *ndata*, int *ndata2*, double * *bout*)

12.175.1.12 void [TKmultMatrixVec_sq](#) (double ** *idcm*, double * *b*, int *ndata*, double * *bout*)

12.176 TKspectrum.C File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "tempo2.h"
#include "T2toolkit.h"
#include "TKfit.h"
#include "TKspectrum.h"
```

Include dependency graph for TKspectrum.C:



Macros

- `#define ABS(x) ((x) < 0 ? -(x) : (x))`
- `#define MAX(x, y) ((x) > (y) ? (x) : (y))`
- `#define MIN(x, y) ((x) < (y) ? (x) : (y))`

Functions

- `double TKspectrum (double *x, double *y, double *e, int n, int averageTime, int smoothWidth, int smoothType, int fitSpline, int preWhite, int specType, double ofac, double hifac, int specOut, double *outX, double *outY, int *nout, int calcWhite, int output, double *outY_re, double *outY_im)`
- `void TKfirstDifference (double *x, double *y, int n)`
- `void TK_fitSinusoids (double *x, double *y, double *sig, int n, double *outX, double *outY, int *outN)`
- `void sineFunc (double x, double *v, int ma)`
- `void TK_weightLS (double *x, double *y, double *sig, int n, double *outX, double *outY, int *outN, double *outY_re, double *outY_im)`
- `void TK_dft (double *x, double *y, int n, double *outX, double *outY, int *outN, double *outY_re, double *outY_im)`
- `void TKaveragePts (double *x, double *y, int n, int width, double *meanX, double *meanY, int *nMean)`
- `void TKsortit (double *x, double *y, int n)`
- `void TKboxcar (double *x, double *y, int n, double *ox, double *oy, int *on, int width)`
- `void TKhann (double *x, double *y, int n, double *ox, double *oy, int *on, int width)`
- `void TKcmonot (int n, double x[], double y[], double yd[][4])`
- `void TKspline_interpolate (int n, double *x, double *y, double yd[][4], double *interpX, double *interpY, int nInterp)`
- `void TKlomb_d (double *x, double *y, int n, double ofac, double hifac, double *ox, double *oy, int *outN, double *var)`
- `int TK_fft (short int dir, long n, double *x, double *y)`
- `void TKcalcSigmaz (pulsar psr, int weights, double *ret_tau, double *ret_szbias, double *ret_e1, double *ret_e2, int *ret_nval, double mintau)`
- `void fit4 (int *nfit, double *p4, double *cov4, int ndostats, double *chidf, double *avewt)`
- `void mat20 (double sam[21][21], double a[21][21], int n, double *determ, int *nbad)`
- `void getprtj (int n)`
- `void getweights (int n, double *wt)`

- void `readin` (`pulsar` psr)
- void `indexx8` (int n, double *arrin, int *indx)
- void `TKinterpolateSplineSmoothFixedXPts` (double *inX, double *inY, int inN, double *interpX, double *interpY, int nInterp)
- int `calcSpectraErr` (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY, double *specE, int nfit)
- int `calcSpectra` (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY, int nfit)
- int `calcSpectra_ri` (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY_R, double *specY_I, int nfit, `pulsar` *psr)
- int `calcSpectra_ri_T` (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY_R, double *specY_I, int nfit, double T, char fitfuncMode, `pulsar` *psr)
- void `fitMeanSineFunc` (double x, double *v, int nfit, `pulsar` *psr, int ival, int ipsr)
- void `fitMeanSineFunc_IFUNC` (double x, double *v, int nfit, `pulsar` *psr, int ival, int ipsr)
- void `fitCosSineFunc` (double x, double *v, int nfit, `pulsar` *psr, int ival, int ipsr)

Variables

- double `GLOBAL_OMEGA` = 0
- int `npt`
- int `nusewt`
- int `nxunits`
- int `ntunits`
- int `nformat`
- int `nwriteres`
- int `nbintype`
- int `npt1last`
- int `npt2last`
- int `ncubic`
- int `ncubics`
- int `ntau`
- int `linfile`
- int `indx` [90000]
- int `ndim`
- double `data` [90000]
- double `utjd` [90000]
- double `taumin`
- double `sigmai` [90000]
- double `permax`
- double `root2`
- double `utjd1`
- double `utjd2`
- double `tmin`
- double `tmax`
- double `xmin`
- double `xmax`
- double `utjdlast`
- double `tausec`
- double `taumax`
- double `tauday`
- double `prtl` [5]
- double `utmean`
- double `secyear`
- double `taulog`
- double `addvar`

- double [tauear](#)
- double [tauensure](#)
- double [tdiffmin](#)
- double [utfirst](#)
- double [utlast](#)
- double [globalOmega](#)
- bool [verbose_calc_spectra](#) =false

12.176.1 Macro Definition Documentation

12.176.1.1 `#define ABS(x) ((x) < 0 ? -(x) : (x))`

12.176.1.2 `#define MAX(x, y) ((x) > (y) ? (x) : (y))`

12.176.1.3 `#define MIN(x, y) ((x) < (y) ? (x) : (y))`

12.176.2 Function Documentation

12.176.2.1 `int calcSpectra (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY, int nfit)`

12.176.2.2 `int calcSpectra_ri (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY_R, double * specY_I, int nfit, pulsar * psr)`

12.176.2.3 `int calcSpectra_ri_T (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY_R, double * specY_I, int nfit, double T, char fitfuncMode, pulsar * psr)`

12.176.2.4 `int calcSpectraErr (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY, double * specE, int nfit)`

12.176.2.5 `void fit4 (int * nfit, double * p4, double * cov4, int ndostats, double * chidf, double * avewt)`

12.176.2.6 `void fitCosSineFunc (double x, double * v, int nfit, pulsar * psr, int ival, int ipsr)`

12.176.2.7 `void fitMeanSineFunc (double x, double * v, int nfit, pulsar * psr, int ival, int ipsr)`

12.176.2.8 `void fitMeanSineFunc_IFUNC (double x, double * v, int nfit, pulsar * psr, int ival, int ipsr)`

12.176.2.9 `void getprtj (int n)`

12.176.2.10 `void getweights (int n, double * wt)`

12.176.2.11 `void indexx8 (int n, double * arrin, int * indx)`

12.176.2.12 `void mat20 (double sam[21][21], double a[21][21], int n, double * determ, int * nbad)`

12.176.2.13 `void readin (pulsar psr)`

12.176.2.14 `void sineFunc (double x, double * v, int ma)`

12.176.2.15 `void TK_dft (double * x, double * y, int n, double * outX, double * outY, int * outN, double * outY_re, double * outY_im)`

12.176.2.16 `int TK_fft (short int dir, long n, double * x, double * y)`

12.176.2.17 `void TK_fitSinusoids (double * x, double * y, double * sig, int n, double * outX, double * outY, int * outN)`

- 12.176.2.18 void TK_weightLS (double * *x*, double * *y*, double * *sig*, int *n*, double * *outX*, double * *outY*, int * *outN*, double * *outY_re*, double * *outY_im*)
- 12.176.2.19 void TKaveragePts (double * *x*, double * *y*, int *n*, int *width*, double * *meanX*, double * *meanY*, int * *nMean*)
- 12.176.2.20 void TKboxcar (double * *x*, double * *y*, int *n*, double * *ox*, double * *oy*, int * *on*, int *width*)
- 12.176.2.21 void TKcalcSigmaz (pulsar *psr*, int *weights*, double * *ret_tau*, double * *ret_szbias*, double * *ret_e1*, double * *ret_e2*, int * *ret_nval*, double *mintau*)
- 12.176.2.22 void TKcmonot (int *n*, double *x*[], double *y*[], double *yd*[][4])
- 12.176.2.23 void TKfirstDifference (double * *x*, double * *y*, int *n*)
- 12.176.2.24 void TKhann (double * *x*, double * *y*, int *n*, double * *ox*, double * *oy*, int * *on*, int *width*)
- 12.176.2.25 void TKinterpolateSplineSmoothFixedXPTs (double * *inX*, double * *inY*, int *inN*, double * *interpX*, double * *interpY*, int *nInterp*)
- 12.176.2.26 void TKlomb_d (double * *x*, double * *y*, int *n*, double *ofac*, double *hifac*, double * *ox*, double * *oy*, int * *outN*, double * *var*)
- 12.176.2.27 void TKsortit (double * *x*, double * *y*, int *n*)
- 12.176.2.28 double TKspectrum (double * *x*, double * *y*, double * *e*, int *n*, int *averageTime*, int *smoothWidth*, int *smoothType*, int *fitSpline*, int *preWhite*, int *specType*, double *ofac*, double *hifac*, int *specOut*, double * *outX*, double * *outY*, int * *nout*, int *calcWhite*, int *output*, double * *outY_re*, double * *outY_im*)

(double)nSmooth/(double)nSmooth;

(double)nSpec;///(double)nSmooth/(double)nSmooth;

(double)nSpec;///(double)nSmooth/(double)nSmooth;

- 12.176.2.29 void TKspline_interpolate (int *n*, double * *x*, double * *y*, double *yd*[][4], double * *interpX*, double * *interpY*, int *nInterp*)

12.176.3 Variable Documentation

12.176.3.1 double addvar

12.176.3.2 double data[90000]

12.176.3.3 double GLOBAL_OMEGA = 0

12.176.3.4 double globalOmega

12.176.3.5 int indx[90000]

12.176.3.6 int linfile

12.176.3.7 int nbintype

12.176.3.8 int ncubic

12.176.3.9 int ncubics

12.176.3.10 int ndim

12.176.3.11 int nformat

12.176.3.12 int npt

12.176.3.13 int npt1last

12.176.3.14 int npt2last

12.176.3.15 int ntau

12.176.3.16 int ntunits

12.176.3.17 int nusewt

12.176.3.18 int nwriteres

12.176.3.19 int nxunits

12.176.3.20 double permax

12.176.3.21 double prtl[5]

12.176.3.22 double root2

12.176.3.23 double secyear

12.176.3.24 double sigmai[90000]

12.176.3.25 double tauday

12.176.3.26 double tauensure

12.176.3.27 double tauilog

12.176.3.28 double taumax

12.176.3.29 double taumin

12.176.3.30 double tausec

12.176.3.31 double tauyear

12.176.3.32 double tdiffmin

12.176.3.33 double tmax

12.176.3.34 double tmin

12.176.3.35 double utfirst

12.176.3.36 double utjd[90000]

12.176.3.37 double utjd1

12.176.3.38 double utjd2

12.176.3.39 double utjdlast

12.176.3.40 double utlast

12.176.3.41 double utmean

12.176.3.42 bool verbose_calc_spectra =false

12.176.3.43 double xmax

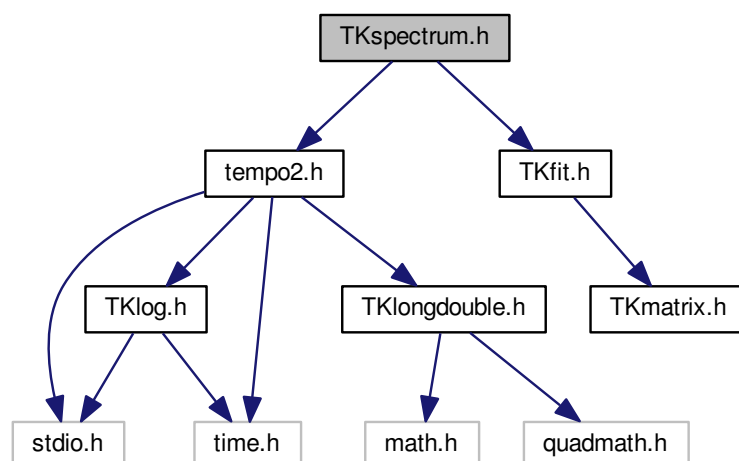
12.176.3.44 double xmin

12.177 TKspectrum.h File Reference

```
#include "tempo2.h"
```

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

Include dependency graph for TKspectrum.h:



This graph shows which files directly or indirectly include this file:



Classes

- struct [complexVal](#)

Macros

- #define [ABS](#)(x) ((x) < 0 ? -(x) : (x))

- `#define MAX(x, y) ((x) > (y) ? (x) : (y))`
- `#define MIN(x, y) ((x) < (y) ? (x) : (y))`

Typedefs

- typedef struct [complexVal](#) [complexVal](#)

Functions

- void [readin](#) ([pulsar](#) psr)
- void [getprtj](#) (int n)
- void [indexx8](#) (int n, double *arrin, int *indx)
- void [getweights](#) (int n, double *wt)
- void [fit4](#) (int *nfit, double *p4, double *cov4, int ndostats, double *chidf, double *avewt)
- void [mat20](#) (double sam[21][21], double a[21][21], int n, double *determ, int *nbad)
- void [sineFunc](#) (double x, double *v, int ma)
- void [TKsortit](#) (double *x, double *y, int n)
- void [TKaveragePts](#) (double *x, double *y, int n, int width, double *meanX, double *meanY, int *nMean)
- void [TKcmonot](#) (int n, double x[], double y[], double yd[][4])
- void [TKspline_interpolate](#) (int n, double *x, double *y, double yd[][4], double *interpX, double *interpY, int nInterp)
- void [TKinterpolateSplineSmoothFixedXPts](#) (double *inX, double *inY, int inN, double *interpX, double *interpY, int nInterp)
- void [TKhann](#) (double *x, double *y, int n, double *ox, double *oy, int *on, int width)
- void [TKfirstDifference](#) (double *x, double *y, int n)
- void [TK_fitSine](#) (double *x, double *y, double *e, int n, int wErr, double *outX, double *outY, int *outN)
- void [TKlomb_d](#) (double *x, double *y, int n, double ofac, double hifac, double *ox, double *oy, int *outN, double *var)
- int [TK_fft](#) (short int dir, long n, double *x, double *y)
- void [TK_dft](#) (double *x, double *y, int n, double *outX, double *outY, int *outN, double *outY_re, double *outY_im)
- void [TK_weightLS](#) (double *x, double *y, double *sig, int n, double *outX, double *outY, int *outN, double *outY_re, double *outY_im)
- void [TK_fitSinusoids](#) (double *x, double *y, double *sig, int n, double *outX, double *outY, int *outN)
- void [fitMeanSineFunc](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival, int ipsr)
- void [fitCosSineFunc](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival, int ipsr)
- int [calcSpectraErr](#) (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY, double *specE, int nfit)
- double [TKspectrum](#) (double *x, double *y, double *e, int n, int averageTime, int [smoothWidth](#), int smoothType, int fitSpline, int preWhite, int specType, double ofac, double hifac, int specOut, double *outX, double *outY, int *nout, int calcWhite, int [output](#), double *outY_re, double *outY_im)
- void [TKboxcar](#) (double *x, double *y, int n, double *ox, double *oy, int *on, int width)
- void [TKcalcSigmaz](#) ([pulsar](#) psr, int weights, double *ret_tau, double *ret_szbias, double *ret_e1, double *ret_e2, int *ret_nval, double mintau)
- int [calcSpectra](#) (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY, int nfit)
- int [calcSpectra_ri](#) (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY_R, double *specY_I, int nfit, [pulsar](#) *psr)
- int [calcSpectra_ri_T](#) (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY_R, double *specY_I, int nfit, double T, char useCM, [pulsar](#) *psr)
- void [fitMeanSineFunc_IFUNC](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival, int ipsr)
- void [fitCosSineFunc](#) (double x, double *v, int nfit, [pulsar](#) *psr, int ival)

Variables

- double [GLOBAL_OMEGA](#)
- bool [verbose_calc_spectra](#)

12.177.1 Macro Definition Documentation

12.177.1.1 `#define ABS(x) ((x) < 0 ? -(x) : (x))`

12.177.1.2 `#define MAX(x, y) ((x) > (y) ? (x) : (y))`

12.177.1.3 `#define MIN(x, y) ((x) < (y) ? (x) : (y))`

12.177.2 Typedef Documentation

12.177.2.1 `typedef struct complexVal complexVal`

12.177.3 Function Documentation

12.177.3.1 `int calcSpectra (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY, int nfit)`

12.177.3.2 `int calcSpectra_ri (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY_R, double * specY_I, int nfit, pulsar * psr)`

12.177.3.3 `int calcSpectra_ri_T (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY_R, double * specY_I, int nfit, double T, char useCM, pulsar * psr)`

12.177.3.4 `int calcSpectraErr (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY, double * specE, int nfit)`

12.177.3.5 `void fit4 (int * nfit, double * p4, double * cov4, int ndostats, double * chidf, double * avewt)`

12.177.3.6 `void fitCosSineFunc (double x, double * v, int nfit, pulsar * psr, int ival, int ipsr)`

12.177.3.7 `void fitCosSineFunc (double x, double * v, int nfit, pulsar * psr, int ival)`

12.177.3.8 `void fitMeanSineFunc (double x, double * v, int nfit, pulsar * psr, int ival, int ipsr)`

12.177.3.9 `void fitMeanSineFunc_IFUNC (double x, double * v, int nfit, pulsar * psr, int ival, int ipsr)`

12.177.3.10 `void getprtj (int n)`

12.177.3.11 `void getweights (int n, double * wt)`

12.177.3.12 `void indexx8 (int n, double * arrin, int * indx)`

12.177.3.13 `void mat20 (double sam[21][21], double a[21][21], int n, double * determ, int * nbad)`

12.177.3.14 `void readin (pulsar psr)`

12.177.3.15 `void sineFunc (double x, double * v, int ma)`

12.177.3.16 `void TK_dft (double * x, double * y, int n, double * outX, double * outY, int * outN, double * outY_re, double * outY_im)`

```

12.177.3.17  int TK_fft ( short int dir, long n, double * x, double * y )

12.177.3.18  void TK_fitSine ( double * x, double * y, double * e, int n, int wErr, double * outX, double * outY, int * outN )

12.177.3.19  void TK_fitSinusoids ( double * x, double * y, double * sig, int n, double * outX, double * outY, int * outN )

12.177.3.20  void TK_weightLS ( double * x, double * y, double * sig, int n, double * outX, double * outY, int * outN, double
* outY_re, double * outY_im )

12.177.3.21  void TKaveragePts ( double * x, double * y, int n, int width, double * meanX, double * meanY, int * nMean )

12.177.3.22  void TKboxcar ( double * x, double * y, int n, double * ox, double * oy, int * on, int width )

12.177.3.23  void TKcalcSigmaz ( pulsar psr, int weights, double * ret_tau, double * ret_szbias, double * ret_e1, double *
ret_e2, int * ret_nval, double mintau )

12.177.3.24  void TKcmonot ( int n, double x[], double y[], double yd[][4] )

12.177.3.25  void TKfirstDifference ( double * x, double * y, int n )

12.177.3.26  void TKhann ( double * x, double * y, int n, double * ox, double * oy, int * on, int width )

12.177.3.27  void TKinterpolateSplineSmoothFixedXPTs ( double * inX, double * inY, int inN, double * interpX, double *
interpY, int nInterp )

12.177.3.28  void TKlomb_d ( double * x, double * y, int n, double ofac, double hifac, double * ox, double * oy, int * outN,
double * var )

12.177.3.29  void TKsortit ( double * x, double * y, int n )

12.177.3.30  double TKspectrum ( double * x, double * y, double * e, int n, int averageTime, int smoothWidth, int
smoothType, int fitSpline, int preWhite, int specType, double ofac, double hifac, int specOut, double * outX,
double * outY, int * nout, int calcWhite, int output, double * outY_re, double * outY_im )

```

```
(double)nSmooth/(double)nSmooth;
```

```
(double)nSpec;///(double)nSmooth/(double)nSmooth;
```

```
(double)nSpec;///(double)nSmooth/(double)nSmooth;
```

```

12.177.3.31  void TKspline_interpolate ( int n, double * x, double * y, double yd[][4], double * interpX, double * interpY, int
nInterp )

```

12.177.4 Variable Documentation

```
12.177.4.1  double GLOBAL_OMEGA
```

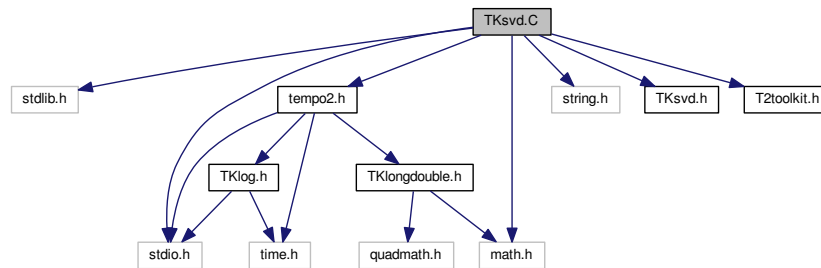
```
12.177.4.2  bool verbose_calc_spectra
```

12.178 TKsvd.C File Reference

```
#include <stdlib.h>
```



```
#include <math.h>
#include <stdio.h>
#include <string.h>
#include "tempo2.h"
#include "TKsvd.h"
#include "T2toolkit.h"
Include dependency graph for TKsvd.C:
```



Functions

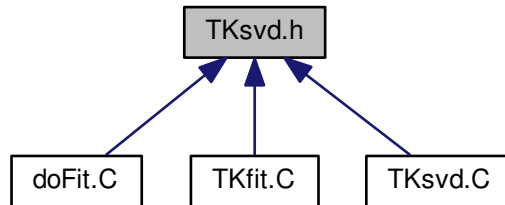
- void [TKsingularValueDecomposition_Isq](#) (longdouble **designMatrix, int n, int nf, longdouble **v, longdouble *w, longdouble **u)
- void [TKbidiagonal](#) (longdouble **a, longdouble *an, int ndata, int nfit, longdouble **v, longdouble *w, longdouble **u, longdouble *rv1)
- void [TKbacksubstitution_svd](#) (longdouble **V, longdouble *w, longdouble **U, longdouble *b, longdouble *x, int n, int nf)
- longdouble [TKpythag](#) (longdouble a, longdouble b)

12.178.1 Function Documentation

- 12.178.1.1 void [TKbacksubstitution_svd](#) (longdouble ** V, longdouble * w, longdouble ** U, longdouble * b, longdouble * x, int n, int nf)
- 12.178.1.2 void [TKbidiagonal](#) (longdouble ** a, longdouble * an, int ndata, int nfit, longdouble ** v, longdouble * w, longdouble ** u, longdouble * rv1)
- 12.178.1.3 longdouble [TKpythag](#) (longdouble a, longdouble b)
- 12.178.1.4 void [TKsingularValueDecomposition_Isq](#) (longdouble ** designMatrix, int n, int nf, longdouble ** v, longdouble * w, longdouble ** u)

12.179 TKsvd.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- void [TKsingularValueDecomposition_Isq](#) (longdouble **designMatrix, int n, int nf, longdouble **v, longdouble *w, longdouble **u)
- void [TKbacksubstitution_svd](#) (longdouble **V, longdouble *w, longdouble **U, longdouble *b, longdouble *x, int n, int nf)
- longdouble [TKpythag](#) (longdouble a, longdouble b)
- void [TKbidiagonal](#) (longdouble **a, longdouble *anorm, int ndata, int nfit, longdouble **v, longdouble *w, longdouble **u, longdouble *rv1)

12.179.1 Function Documentation

12.179.1.1 void [TKbacksubstitution_svd](#) (longdouble ** V, longdouble * w, longdouble ** U, longdouble * b, longdouble * x, int n, int nf)

12.179.1.2 void [TKbidiagonal](#) (longdouble ** a, longdouble * anorm, int ndata, int nfit, longdouble ** v, longdouble * w, longdouble ** u, longdouble * rv1)

12.179.1.3 longdouble [TKpythag](#) (longdouble a, longdouble b)

12.179.1.4 void [TKsingularValueDecomposition_Isq](#) (longdouble ** designMatrix, int n, int nf, longdouble ** v, longdouble * w, longdouble ** u)

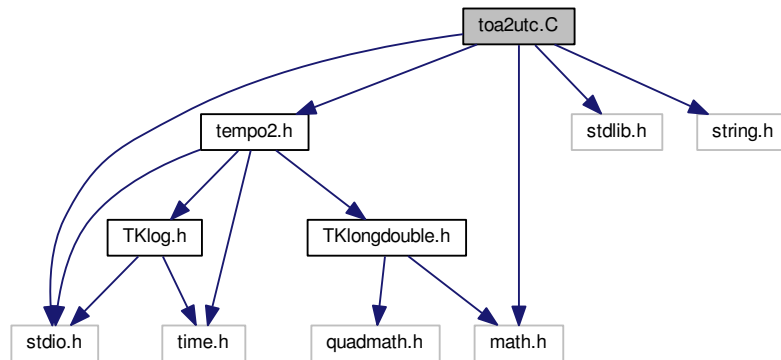
12.180 toa2utc.C File Reference

```

#include <math.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"

```

Include dependency graph for toa2utc.C:



Macros

- #define [USE_NEW_CLK_CORR](#)

Functions

- double [toa2utc_nist](#) (double toa, char *clockFile)
- double [linearInterpolate](#) (double x1, double y1, double x2, double y2, double x3)
- double [convertTOA](#) (double mjd, char *clks)
- void [toa2utc](#) (pulsar *psr, int npsr)

12.180.1 Macro Definition Documentation

12.180.1.1 #define USE_NEW_CLK_CORR

12.180.2 Function Documentation

12.180.2.1 double convertTOA (double mjd, char * clks)

12.180.2.2 double linearInterpolate (double x1, double y1, double x2, double y2, double x3)

12.180.2.3 void toa2utc (pulsar * psr, int npsr)

12.180.2.4 double toa2utc_nist (double toa, char * clockFile)

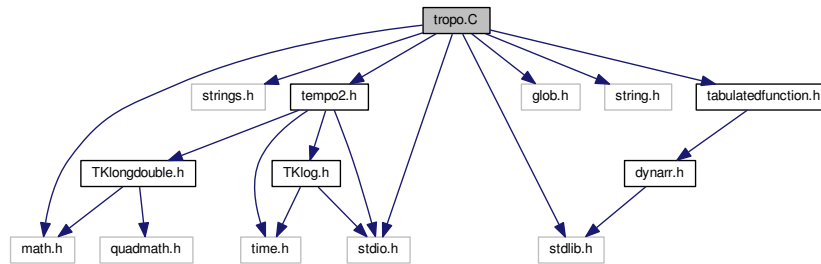
12.181 tropo.C File Reference

```

#include <math.h>
#include <strings.h>
#include "tempo2.h"
#include <stdio.h>
#include <stdlib.h>
#include <glob.h>
#include <string.h>
#include "tabulatedfunction.h"

```

Include dependency graph for tropo.C:



Classes

- struct [MeteorologyFunction](#)

Functions

- double [NMF_hydrostatic](#) (double utc_mjd, double site_latitude, double site_height, double source_elevation)
- double [NMF_wet](#) (double site_latitude, double source_elevation)
- void [MeteorologyFunction_load](#) ([MeteorologyFunction](#) *func, char *fileName)
- double [MeteorologyFunction_getValue](#) ([MeteorologyFunction](#) *func, double mjd)
- double [MeteorologyFunction_getStartMJD](#) ([MeteorologyFunction](#) *func)
- double [MeteorologyFunction_getEndMJD](#) ([MeteorologyFunction](#) *func)
- void [initialize_meteorology_table](#) (int dispWarnings, const char *path, const char *extension, [DynamicArray](#) *tables, const char *description)
- void [initialize_meteorology_tables](#) (int dispWarnings)
- double [getMeteorologicalValue](#) ([DynamicArray](#) *tables, char *siteName, double mjd, int warnings)
- double [getZenithWetDelay](#) (char *siteName, double mjd, int warnings)
- double [getSurfaceAtmosphericPressure](#) (char *siteName, double mjd, int warnings)
- void [compute_tropospheric_delays](#) ([pulsar](#) *psr, int npsr)

Variables

- [DynamicArray](#) zenithWetDelayTables
- [DynamicArray](#) surfaceAtmosphericPressureTables

12.181.1 Function Documentation

12.181.1.1 void [compute_tropospheric_delays](#) ([pulsar](#) * *psr*, int *npsr*)

12.181.1.2 double [getMeteorologicalValue](#) ([DynamicArray](#) * *tables*, char * *siteName*, double *mjd*, int *warnings*)

12.181.1.3 double [getSurfaceAtmosphericPressure](#) (char * *siteName*, double *mjd*, int *warnings*)

12.181.1.4 double [getZenithWetDelay](#) (char * *siteName*, double *mjd*, int *warnings*)

12.181.1.5 void [initialize_meteorology_table](#) (int *dispWarnings*, const char * *path*, const char * *extension*, [DynamicArray](#) * *tables*, const char * *description*)

12.181.1.6 void initialize_meteorology_tables (int *dispWarnings*)

12.181.1.7 double MeteorologyFunction_getEndMJD (MeteorologyFunction * *func*)

12.181.1.8 double MeteorologyFunction_getStartMJD (MeteorologyFunction * *func*)

12.181.1.9 double MeteorologyFunction_getValue (MeteorologyFunction * *func*, double *mjd*)

12.181.1.10 void MeteorologyFunction_load (MeteorologyFunction * *func*, char * *fileName*)

12.181.1.11 double NMF_hydrostatic (double *utc_mjd*, double *site_latitude*, double *site_height*, double *source_elevation*)

12.181.1.12 double NMF_wet (double *site_latitude*, double *source_elevation*)

12.181.2 Variable Documentation

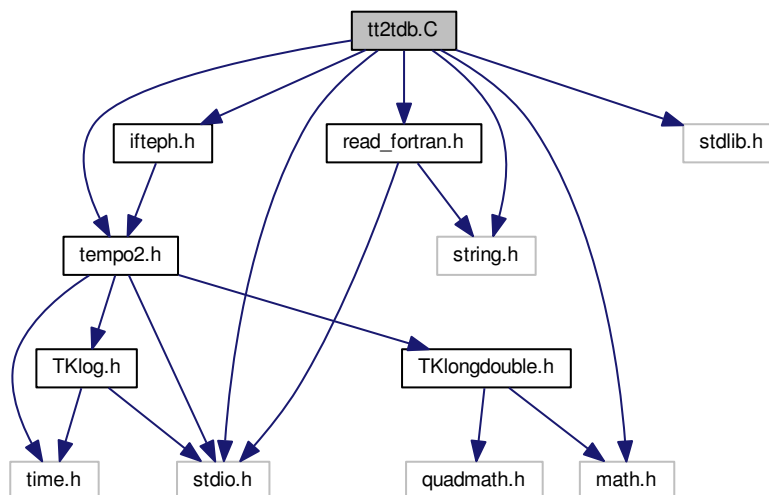
12.181.2.1 DynamicArray surfaceAtmosphericPressureTables

12.181.2.2 DynamicArray zenithWetDelayTables

12.182 tt2tdb.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include "tempo2.h"
#include "read_fortran.h"
#include "ifteph.h"
```

Include dependency graph for tt2tdb.C:



Functions

- void [init_ifte](#) ()

- double `IF_deltaT` (`longdouble` `mjd_tt`)
- double `FB_deltaT` (`longdouble` `mjd_tt`)
- void `tt2tb` (`pulsar` `*psr`, int `npsr`)

12.182.1 Function Documentation

12.182.1.1 double `FB_deltaT` (`longdouble` `mjd_tt`)

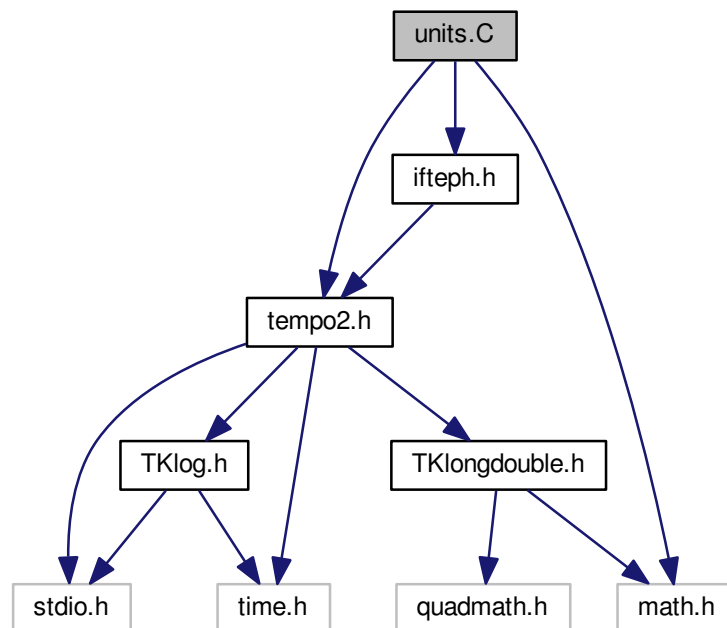
12.182.1.2 double `IF_deltaT` (`longdouble` `mjd_tt`)

12.182.1.3 void `init_ifte` ()

12.182.1.4 void `tt2tb` (`pulsar` `*psr`, int `npsr`)

12.183 units.C File Reference

```
#include "tempo2.h"
#include "ifteph.h"
#include <math.h>
Include dependency graph for units.C:
```



Functions

- void `scale_param` (struct `parameter` `*p`, int `arr`, `longdouble` `f`)
- void `xform_mjd` (struct `parameter` `*p`, int `arr`, `longdouble` `f`)
- void `transform_units` (struct `pulsar` `*psr`, int `from`, int `to`)

12.183.1 Function Documentation

12.183.1.1 void scale_param (struct parameter * *p*, int *arr*, longdouble *f*)

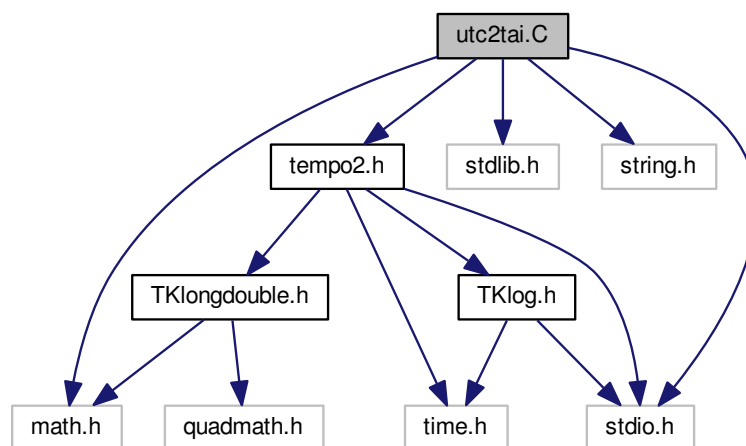
12.183.1.2 void transform_units (struct pulsar * *psr*, int *from*, int *to*)

12.183.1.3 void xform_mjd (struct parameter * *p*, int *arr*, longdouble *f*)

12.184 utc2tai.C File Reference

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include "tempo2.h"
```

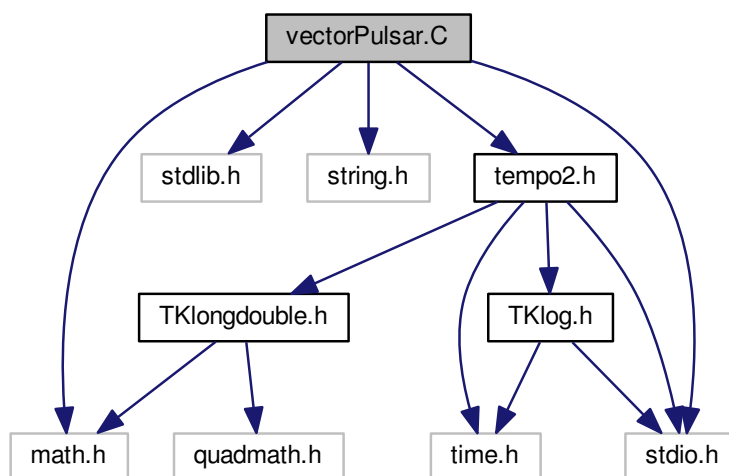
Include dependency graph for utc2tai.C:



12.185 vectorPulsar.C File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include "tempo2.h"
```

Include dependency graph for vectorPulsar.C:



Functions

- void `vectorPulsar` (`pulsar *psr`, int `npsr`)

12.185.1 Function Documentation

12.185.1.1 void `vectorPulsar` (`pulsar * psr`, int `npsr`)

Index

`_DARWIN_USE_64_BIT_INODE`

`config.h`, [100](#)

`_LOG`

`TKlog.h`, [342](#)

`_TKchklog`

`TKlog.C`, [340](#)

`TKlog.h`, [343](#)

`_itt`

`mjk_plug.C`, [220](#)

`ABS`

`TKspectrum.C`, [354](#)

`TKspectrum.h`, [359](#)

`ACCEL_LSQ`

`T2accel.h`, [268](#)

`ACCEL_MULTMATRIX`

`T2accel.h`, [268](#)

`ACCEL_UINV`

`T2accel.h`, [269](#)

`aSize`

parameter, [52](#)

`AU_DIST`

`tempo2.h`, [307](#)

`AULTSC`

`tempo2.h`, [307](#)

`accel_lsq_qr`

`T2accel.C`, [267](#)

`T2accel.h`, [269](#)

`accel_multMatrix`

`T2accel.C`, [267](#)

`T2accel.h`, [269](#)

`accel_multMatrixVec`

`T2accel.C`, [267](#)

`T2accel.h`, [269](#)

`accel_uinv`

`T2accel.C`, [267](#)

`T2accel.h`, [269](#)

`across_g`

`gwSrc`, [36](#)

`gwgeneralSrc`, [34](#)

`across_im_g`

`gwSrc`, [36](#)

`gwgeneralSrc`, [34](#)

`actual`

sample, [73](#)

`add_pulseNumber_plug.C`

`tempoOutput`, [144](#)

`addCovar`

`cholesky.C`, [89](#)

`addKeplerianJumps`

`T2model.C`, [288](#)

`addRed_plug.C`

`graphicalInterface`, [145](#)

help, [145](#)

`plugVersionCheck`, [145](#)

`addTNGlobalEQ`

pulsar, [60](#)

`addedNoise`

observation, [44](#)

`addvar`

`sigmaz_plug.C`, [235](#)

`TKspectrum.C`, [355](#)

`age.C`, [79](#)

`calc_age`, [79](#)

`derivs`, [79](#)

`rk4rms`, [79](#)

`aliases`

`ObservatoryAliasList`, [51](#)

`allParTim`

`calcDMe_plug.C`, [157](#)

`allocateMemory`

`initialise.C`, [137](#)

`tempo2.h`, [318](#)

`alpha`

`fixData_plug.C`, [181](#)

`amod`

`sw_delay.C`, [262](#)

`analyticChol_plug.C`

`graphicalInterface`, [146](#)

help, [146](#)

`plugVersionCheck`, [146](#)

`ang`

`get_obsCoord.C`, [122](#)

`angle_plug.C`

`graphicalInterface`, [147](#)

help, [147](#)

`psrangle`, [147](#)

`aplus_g`

`gwSrc`, [36](#)

`gwgeneralSrc`, [34](#)

`aplus_im_g`

`gwSrc`, [36](#)

`gwgeneralSrc`, [34](#)

`applet_plug.C`

`fortranMod`, [148](#)

`graphicalInterface`, [148](#)

help, [148](#)

`nint_derived`, [148](#)

`parseLine`, [148](#)

- plugVersionCheck, [149](#)
- rnd8, [148](#)
- ascii
 - calcDMe_plug.C, [157](#)
- asl_g
 - gwgeneralSrc, [34](#)
- asl_im_g
 - gwgeneralSrc, [34](#)
- ast_g
 - gwgeneralSrc, [34](#)
- ast_im_g
 - gwgeneralSrc, [34](#)
- atimfake
 - polyco.C, [247](#)
- au
 - jpl_eph_data, [39](#)
- auto_constraints
 - pulsar, [60](#)
- autoConstraints
 - constraints.C, [102](#)
 - tempo2.h, [318](#)
- autoDM_plug.C
 - graphicalInterface, [149](#)
 - help, [149](#)
 - plugVersionCheck, [150](#)
- autoSpectralFit_plug.C
 - graphicalInterface, [150](#)
 - help, [150](#)
 - MAX_FREQ, [150](#)
- autosetDMCM
 - constraints.C, [102](#)
 - constraints.h, [104](#)
- average
 - checkWhite_plug.C, [160](#)
- averageData_plug.C
 - graphicalInterface, [151](#)
 - help, [151](#)
 - MAX_TIMES, [151](#)
 - plugVersionCheck, [152](#)
- AverageEpochWidth
 - pulsar, [60](#)
- AverageFlag
 - pulsar, [60](#)
- averagePts
 - plk_plug.C, [226](#)
- AverageResiduals
 - pulsar, [60](#)
- averageResiduals
 - formResiduals.C, [121](#)
- averagebat
 - observation, [44](#)
- averageerr
 - observation, [44](#)
- averageres
 - observation, [44](#)
- avx_g
 - gwgeneralSrc, [34](#)
- avx_im_g
 - gwgeneralSrc, [34](#)
- avx_g
 - gwgeneralSrc, [34](#)
- avx_im_g
 - gwgeneralSrc, [34](#)
- BIG_G
 - GWevolve_plug.C, [203](#)
 - tempo2.h, [307](#)
- BOLDCOLOR
 - TKlog.h, [342](#)
- BTJmodel
 - BTJmodel.C, [81](#)
 - tempo2.h, [318](#)
- BTJmodel.C, [80](#)
 - BTJmodel, [81](#)
 - updateBTJ, [81](#)
- BTXmodel
 - BTXmodel.C, [83](#)
 - tempo2.h, [318](#)
- BTXmodel.C, [82](#)
 - BTXmodel, [83](#)
 - updateBTX, [83](#)
- BTmodel
 - BTmodel.C, [82](#)
 - tempo2.h, [318](#)
- BTmodel.C, [81](#)
 - BTmodel, [82](#)
 - updateBT, [82](#)
- BUFSIZE
 - TKlongdouble.C, [344](#)
- badness
 - ClockCorrectionFunction, [29](#)
- bary_plug.C
 - GRS80_A, [152](#)
 - GRS80_F, [152](#)
 - graphicalInterface, [153](#)
 - help, [153](#)
 - ITRF_to_GRS80, [153](#)
 - plugVersionCheck, [153](#)
- basic_plug.C
 - callFit, [153](#)
 - graphicalInterface, [154](#)
 - help, [154](#)
 - plot_ppdot, [154](#)
 - plugVersionCheck, [154](#)
- bat
 - observation, [45](#)
- batCorr
 - observation, [45](#)
- bbat
 - observation, [45](#)
- bin_dmCount
 - calcDMe_plug.C, [157](#)
- bin_dmCount_inc
 - calcDMe_plug.C, [157](#)
- bin_fitCount
 - calcDMe_plug.C, [157](#)
- bin_fitCount_inc

- calcDMe_plug.C, 157
- binObs
 - calcDMe_plug.C, 157
- binResiduals
 - plk_plug.C, 226
- binSizeDays
 - calcDMe_plug.C, 157
- binStart
 - calcDMe_plug.C, 157
- binary_frequency
 - T1Polyco, 74
- binary_phase
 - T1Polyco, 74
- binaryModel
 - pulsar, 60
- bootStrap
 - pulsar, 60
- bootstrap
 - bootstrap.C, 80
 - tempo2.h, 318
- bootstrap.C, 80
 - bootstrap, 80
 - MAX_ITER, 80
 - random, 80
- buf
 - IFTEphemeris, 38
- c_fileptr
 - read_fortran.h, 251
- c_fileptr2
 - read_fortran2.h, 252
- CONSTRAINTfuncs
 - constraints.C, 102
 - constraints.h, 104
- CVSdisplayVersion
 - global.C, 126
 - tempo2.h, 318
- cache
 - jpl_eph_data, 39
- calc_age
 - age.C, 79
- calcAmp
 - GWevolve_plug.C, 204
- calcDMe_plug.C
 - allParTim, 157
 - ascii, 157
 - bin_dmCount, 157
 - bin_dmCount_inc, 157
 - bin_fitCount, 157
 - bin_fitCount_inc, 157
 - binObs, 157
 - binSizeDays, 157
 - binStart, 157
 - callFit, 156
 - dcmFile, 157
 - ddm, 157
 - ddmCount, 157
 - ddmErr, 157
 - ddmMJD, 157
 - describe, 156
 - display, 156
 - dm0, 157
 - dm0_err, 157
 - dmCount, 157
 - dmObs, 157
 - doDisplay, 157
 - f0_0, 157
 - f0_0_err, 157
 - f0fit, 157
 - findFirst, 156
 - findMean, 156
 - findSessions, 156
 - finish_sessions, 157
 - fitCount, 157
 - fitObs, 157
 - freq1f, 157
 - freq2f, 158
 - freqArray, 158
 - freqOffset, 158
 - get_binObs, 156
 - gotOut, 158
 - gr, 158
 - graphicalInterface, 156
 - handleFreqPoints, 156
 - hardcopy, 158
 - header, 158
 - help, 156
 - impCount, 158
 - impObs, 158
 - init, 156
 - interpolateSplineSmooth, 156
 - interpolateWeightedSmooth, 156
 - lastUsedSession, 158
 - mean, 158
 - meanMJD, 158
 - meanMJDval, 158
 - meanVal, 158
 - nSessions, 158
 - nf, 158
 - outDM, 158
 - outFileName, 158
 - outInterpCount, 158
 - outSmoothCount, 158
 - outX, 158
 - outY, 158
 - output, 156
 - parFile, 158
 - plugVersionCheck, 158
 - rawOut, 158
 - resetDMandF0, 156
 - sessionSeparation, 158
 - setAllDeleted, 156
 - setFitParams, 157
 - smoothWidth, 158
 - splineOut, 158
 - start_sessions, 159
 - timFile, 159

- title, 159
- valID, 159
- xlab, 159
- ylab, 159
- calcEfacEquad
 - efacEquad_plug.C, 175
- calcEfacEquad2
 - efacEquad_plug.C, 175
- calcGR
 - T2model.C, 288
- calcRMS
 - tempo2.h, 318
 - textOutput.C, 334
- calcRotN
 - sw_delay.C, 262
- calcShapiro
 - pulsar, 60
- calcSigmaz
 - sigmaz_plug.C, 234
- calcSpectra
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- calcSpectra_plugin
 - detectGWB_plug.C, 169
 - detectGWBnew_plug.C, 170
- calcSpectra_ri
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- calcSpectra_ri_T
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- calcSpectraErr
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- calcSpline
 - sigmaz_plug.C, 234
- calcStat
 - checkWhite_plug.C, 160
- calcYr
 - plotMany_plug.C, 229
- calculate_bclt
 - calculate_bclt.C, 84
 - tempo2.h, 318
- calculate_bclt, 83
 - calculate_bclt, 84
- calculateAngularFactors
 - gwmStats_plug.C, 209
- calculateCholeskyCovarFunc
 - planet_plug.C, 223
 - spectralModel_plug.C, 241
- calculated
 - gwmStats_plug.C, 209
- calculateDailyCovariance
 - planet_plug.C, 223
 - spectralModel_plug.C, 241
- calculateGWCholesky
 - icLimit_plug.C, 215
- calculateGWlim
 - sigmaz_plug.C, 234
- calculateResidualGW
 - GWsim.C, 129
 - GWsim.h, 132
- calculateResidualgeneralGW
 - GWsim.C, 129
 - GWsim.h, 132
- calculateSpectra
 - planet_plug.C, 223
 - spectralModel_plug.C, 241
- calculateSpectrum
 - cholSpectra_plug.C, 161
- calculateStatistic
 - icLimit_plug.C, 215
- calculateWeighting
 - icLimit_plug.C, 215
- callFit
 - basic_plug.C, 153
 - calcDMe_plug.C, 156
 - delays_plug.C, 166
 - fake_plug.C, 177
 - plk_plug.C, 226
 - plotMany_plug.C, 229
 - splk_plug.C, 245
- cgw_angpol
 - pulsar, 60
- cgw_cosinc
 - pulsar, 60
- cgw_h0
 - pulsar, 60
- cgw_mc
 - pulsar, 60
- changeFit
 - glitch_plug.C, 188
- changeFitParameters
 - plk_plug.C, 226
- changeParameters
 - plk_plug.C, 226
- chebpc
 - polyco.C, 247
- Cheby
 - tempo2pred.h, 327
- cheby
 - ChebyModel, 26
 - T2Predictor, 76
- Cheby2D, 25
 - coeff, 25
 - nx, 25
 - ny, 25
- Cheby2D_Construct
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
- Cheby2D_Construct_x_Derivative
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
- Cheby2D_Copy
 - cheby2d.c, 86
- Cheby2D_Destroy

- cheby2d.c, 86
- Cheby2D_Evaluate
 - cheby2d.c, 86
- Cheby2D_Init
 - cheby2d.c, 86
- Cheby2D_Test
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
- cheby2d.c, 84
 - Cheby2D_Construct, 86
 - Cheby2D_Construct_x_Derivative, 86
 - Cheby2D_Copy, 86
 - Cheby2D_Destroy, 86
 - Cheby2D_Evaluate, 86
 - Cheby2D_Init, 86
 - Cheby2D_Test, 86
 - ChebyModel_Copy, 86
 - ChebyModel_Destroy, 86
 - ChebyModel_GetFrequency, 86
 - ChebyModel_GetPhase, 86
 - ChebyModel_Init, 86
 - ChebyModel_Read, 86
 - ChebyModel_Write, 86
 - ChebyModelSet_Destroy, 86
 - ChebyModelSet_GetFrequency, 86
 - ChebyModelSet_GetNearest, 86
 - ChebyModelSet_GetNearestIndex, 86
 - ChebyModelSet_GetPhase, 86
 - ChebyModelSet_Init, 86
 - ChebyModelSet_Insert, 86
 - ChebyModelSet_Keep, 86
 - ChebyModelSet_OutOfRange, 87
 - ChebyModelSet_Read, 86
 - ChebyModelSet_Write, 87
 - M_PII, 86
 - testCheby2D, 87
 - testFunc, 87
- cheby2d_int.C, 87
 - ChebyModel_Construct, 88
 - ChebyModel_Test, 88
 - chebyModelFunc, 88
 - ChebyModelSet_Construct, 88
 - ChebyModelSet_Test, 88
- ChebyModel, 25
 - cheby, 26
 - dispersion_constant, 26
 - freq_end, 26
 - freq_start, 26
 - frequency_cheby, 26
 - mjd_end, 26
 - mjd_start, 26
 - psrname, 26
 - sitename, 26
- ChebyModel_Construct
 - cheby2d_int.C, 88
 - tempo2pred_int.h, 330
- ChebyModel_Copy
 - cheby2d.c, 86
- tempo2pred_int.h, 330
 - ChebyModel_Destroy
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModel_GetFrequency
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModel_GetPhase
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModel_Init
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModel_Read
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModel_Test
 - cheby2d_int.C, 88
 - tempo2pred_int.h, 330
 - ChebyModel_Write
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - chebyModelFunc
 - cheby2d_int.C, 88
 - ChebyModelInfo, 27
 - compute_dispersion_constant, 27
 - model, 27
 - psr, 27
 - ChebyModelSet, 27
 - nsegments, 28
 - segments, 28
 - ChebyModelSet_Construct
 - cheby2d_int.C, 88
 - tempo2pred_int.h, 330
 - ChebyModelSet_Destroy
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModelSet_GetFrequency
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModelSet_GetNearest
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModelSet_GetNearestIndex
 - cheby2d.c, 86
 - ChebyModelSet_GetPhase
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModelSet_Init
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModelSet_Insert
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModelSet_Keep
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
 - ChebyModelSet_OutOfRange

- cheby2d.c, 87
- tempo2pred.h, 328
- ChebyModelSet_Read
 - cheby2d.c, 86
 - tempo2pred_int.h, 330
- ChebyModelSet_Test
 - cheby2d_int.C, 88
 - tempo2pred_int.h, 330
- ChebyModelSet_Write
 - cheby2d.c, 87
 - tempo2pred_int.h, 330
- check_barycentered
 - photons_plug.C, 221
- checkAllSet
 - readParfile.C, 256
- checkLine
 - readParfile.C, 256
- checkMenu
 - glitch_plug.C, 188
 - plk_plug.C, 226
 - spectrum_plug.C, 244
- checkMenu3
 - plk_plug.C, 227
- checkReal
 - GWwhiteLimit_plug.C, 213
- checkSecondComparison
 - compareDsets_plug.C, 164
- checkWhite_plug.C
 - average, 160
 - calcStat, 160
 - corr2pt, 160
 - graphicalInterface, 160
 - help, 160
 - lombScargle, 160
 - MAX_POLY, 160
 - plotHistogram, 160
 - plotResiduals, 160
 - plugVersionCheck, 160
 - shuffle, 160
 - shufflePoints, 160
- cholSpectra_plug.C
 - calculateSpectrum, 161
 - graphicalInterface, 161
 - help, 161
 - OMEGA0, 161
 - plugVersionCheck, 161
 - toffset, 161
- choldc
 - interpolate_plug.C, 217
- cholesky.C, 88
 - addCovar, 89
 - cholesky_covarFunc2matrix, 89
 - cholesky_dmModel, 89
 - cholesky_dmModelCovarParam, 89
 - cholesky_ecm, 89
 - cholesky_formUinv, 89
 - cholesky_powerlawModel, 89
 - cholesky_powerlawModel_withBeta, 90
- cholesky_readFromCovarianceFunction, 90
- cholesky_readT2CholModel, 90
- cholesky_readT2CholModel_R, 90
- cholesky_readT2Model1, 90
- cholesky_readT2Model2, 90
- getCholeskyDiagonals, 90
- getCholeskyMatrix, 90
- LINE_LENGTH, 89
- cholesky.h, 90
 - cholesky_covarFunc2matrix, 91
 - cholesky_dmModel, 91
 - cholesky_dmModelCovarParam, 91
 - cholesky_ecm, 91
 - cholesky_formUinv, 91
 - cholesky_powerlawModel, 91
 - cholesky_powerlawModel_withBeta, 91
 - cholesky_readFromCovarianceFunction, 91
- cholesky_covarFunc2matrix
 - cholesky.C, 89
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_dmModel
 - cholesky.C, 89
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_dmModelCovarParam
 - cholesky.C, 89
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_ecm
 - cholesky.C, 89
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_formUinv
 - cholesky.C, 89
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_powerlawModel
 - cholesky.C, 89
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_powerlawModel_withBeta
 - cholesky.C, 90
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_readFromCovarianceFunction
 - cholesky.C, 90
 - cholesky.h, 91
 - TKcholesky.h, 335
- cholesky_readT2CholModel
 - cholesky.C, 90
- cholesky_readT2CholModel_R
 - cholesky.C, 90
- cholesky_readT2Model1
 - cholesky.C, 90
- cholesky_readT2Model2
 - cholesky.C, 90
- choleskyAutomatic.C, 91

- T2get_covFunc_automatic, 92
- choleskyRoutines.C, 92
 - T2calculateCholesky, 93
 - T2calculateCovarFunc, 93
 - T2calculateDailyCovariance, 93
 - T2calculateSpectra, 94
 - T2cholDecomposition, 94
 - T2cubicFit, 94
 - T2findSmoothCurve, 94
 - T2fitSpectra, 94
 - T2getHighFreqRes, 94
 - T2getWhiteNoiseLevel, 94
 - T2getWhiteRes, 94
 - T2guess_vals, 94
 - T2interpolate, 94
 - T2obtainTimingResiduals, 94
 - T2writeCovarFuncModel, 94
- choleskyRoutines.h, 94
 - EXPSMOOTH, 97
 - FCALPHA, 97
 - FCFINAL, 97
 - NFIT, 97
 - T2calculateCholesky, 96
 - T2calculateCovarFunc, 96
 - T2calculateDailyCovariance, 96
 - T2calculateSpectra, 96
 - T2cholDecomposition, 96
 - T2cubicFit, 96
 - T2findSmoothCurve, 96
 - T2fitSpectra, 96
 - T2get_covFunc_automatic, 96
 - T2getHighFreqRes, 96
 - T2getWhiteNoiseLevel, 96
 - T2getWhiteRes, 96
 - T2guess_vals, 96
 - T2interpolate, 97
 - T2obtainTimingResiduals, 97
 - T2writeCovarFuncModel, 97
 - UPW, 97
 - WNLEVEL, 97
- cholmode
 - clock_plug.C, 162
 - plk_plug.C, 228
- clk_offsE
 - pulsar, 60
- clk_offsT
 - pulsar, 60
- clk_offsV
 - pulsar, 60
- clkOffsN
 - pulsar, 60
- clkcorr.C, 97
 - ClockCorrectionFunction_getCorrection, 98
 - ClockCorrectionFunction_getEndMJD, 98
 - ClockCorrectionFunction_getStartMJD, 98
 - ClockCorrectionFunction_load, 98
 - clockCorrectionFunctions, 99
 - ClockCorrectionSequence_getEndMJD, 98
- ClockCorrectionSequence_getStartMJD, 98
 - clockCorrectionSequences, 99
 - defineClockCorrectionSequence, 98
 - getClockCorrectionSequence, 98
 - getClockCorrections, 98
 - getCorrection, 98
 - getCorrectionTT, 98
 - initialize_ClockCorrections, 98
 - makeClockCorrectionSequence, 98
- clock
 - pulsar, 60
- clock_correction, 28
 - correction, 29
 - corrects_to, 29
- clock_corrections
 - global.C, 126
 - tempo2.C, 299
- clock_corrections_fermi
 - fermi_plug.C, 178
- clock_name
 - observatory, 50
- clock_plug.C
 - cholmode, 162
 - covarFuncFile, 162
 - graphicalInterface, 162
 - help, 162
 - mjd2year, 162
 - plugVersionCheck, 163
 - slaCalyd, 162
 - slaClyd, 162
- clockCorr
 - observation, 45
- ClockCorrectionFunction, 29
 - badness, 29
 - clockFrom, 29
 - clockTo, 29
 - table, 29
- ClockCorrectionFunction_getCorrection
 - clkcorr.C, 98
- ClockCorrectionFunction_getEndMJD
 - clkcorr.C, 98
- ClockCorrectionFunction_getStartMJD
 - clkcorr.C, 98
- ClockCorrectionFunction_load
 - clkcorr.C, 98
- clockCorrectionFunctions
 - clkcorr.C, 99
- ClockCorrectionSequence_getEndMJD
 - clkcorr.C, 98
- ClockCorrectionSequence_getStartMJD
 - clkcorr.C, 98
- clockCorrectionSequences
 - clkcorr.C, 99
- clockFrom
 - ClockCorrectionFunction, 29
- clockFromOverride
 - pulsar, 60
- clockTo

- ClockCorrectionFunction, 29
- close_file
 - read_fortran.h, 251
- close_file2
 - read_fortran2.h, 252
- code
 - observatory, 50
 - ObservatoryAliasList, 51
- coeff
 - Cheby2D, 25
 - T1Polyco, 74
- comment
 - storePrecision, 73
- compareBackends_plug.C
 - graphicalInterface, 163
 - help, 163
 - plugVersionCheck, 163
 - runPlugin, 163
- compareDatasets
 - compareDsets_plug.C, 164
- compareDsets_plug.C
 - checkSecondComparison, 164
 - compareDatasets, 164
 - findOverlap, 165
 - graphicalInterface, 165
 - help, 165
 - idPoint, 165
 - idPoint2, 165
 - plugVersionCheck, 165
- compareObs
 - sortToAs.C, 260
- complexVal, 30
 - imag, 30
 - real, 30
 - TKspectrum.h, 359
- compute_dispersion_constant
 - ChebyModelInfo, 27
- compute_tropospheric_delays
 - tempo2.h, 318
 - tropo.C, 364
- computeConstraintWeights
 - constraints.C, 102
 - constraints.h, 104
- computeU
 - T2-PTAmodel.C, 265
 - T2model.C, 288
- config.h, 99
 - _DARWIN_USE_64_BIT_INODE, 100
 - F77_FUNC, 100
 - F77_FUNC_, 100
 - HAVE_BLAS, 100
 - HAVE_DLERROR, 100
 - HAVE_DLFCN_H, 100
 - HAVE_FFTW3, 100
 - HAVE_INTTYPES_H, 100
 - HAVE_LAPACK, 100
 - HAVE_LIBDL, 100
 - HAVE_LIBDLLOADER, 100
 - HAVE_LIBM, 100
 - HAVE_MEMORY_H, 100
 - HAVE_PGPLOT, 100
 - HAVE_PTHREAD, 100
 - HAVE_STDINT_H, 100
 - HAVE_STDLIB_H, 100
 - HAVE_STRING_H, 100
 - HAVE_STRINGS_H, 100
 - HAVE_SYS_STAT_H, 100
 - HAVE_SYS_TYPES_H, 100
 - HAVE_UNISTD_H, 100
 - LT_OBJDIR, 100
 - PACKAGE, 100
 - PACKAGE_BUGREPORT, 100
 - PACKAGE_NAME, 101
 - PACKAGE_STRING, 101
 - PACKAGE_TARNAME, 101
 - PACKAGE_URL, 101
 - PACKAGE_VERSION, 101
 - STDC_HEADERS, 101
 - TEMPO2_ARCH, 101
 - VERSION, 101
 - X_DISPLAY_MISSING, 101
- consFunc_dmmodel_cw
 - constraints.C, 102
 - constraints.h, 104
- consFunc_dmmodel_cw_year
 - constraints.C, 102
 - constraints.h, 104
- consFunc_dmmodel_dm1
 - constraints.C, 102
 - constraints.h, 104
- consFunc_dmmodel_mean
 - constraints.C, 102
 - constraints.h, 104
- consFunc_ifunc
 - constraints.C, 102
 - constraints.h, 104
- consFunc_ifunc_year
 - constraints.C, 102
 - constraints.h, 104
- consFunc_qifunc_c_year
 - constraints.C, 102
 - constraints.h, 104
- consFunc_qifunc_p_year
 - constraints.C, 102
 - constraints.h, 104
- consFunc_quad_ifunc_c
 - constraints.C, 102
 - constraints.h, 104
- consFunc_quad_ifunc_p
 - constraints.C, 102
 - constraints.h, 104
- consFunc_tel_dx
 - constraints.C, 102
 - constraints.h, 104
- consFunc_tel_dy
 - constraints.C, 102

- constraints.h, [104](#)
- consFunc_tel_dz
 - constraints.C, [102](#)
 - constraints.h, [104](#)
- const2
 - GWevolve_plug.C, [204](#)
- constA0
 - GWevolve_plug.C, [204](#)
- constraint
 - tempo2.h, [314](#)
- constraint_LAST
 - tempo2.h, [315](#)
- constraint_dmmodel_cw_0
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_1
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_2
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_3
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_px
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_year_cos
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_year_cos2
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_year_sin
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_year_sin2
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_year_xcos
 - tempo2.h, [314](#)
- constraint_dmmodel_cw_year_xsin
 - tempo2.h, [314](#)
- constraint_dmmodel_dm1
 - tempo2.h, [314](#)
- constraint_dmmodel_mean
 - tempo2.h, [314](#)
- constraint_ifunc_0
 - tempo2.h, [314](#)
- constraint_ifunc_1
 - tempo2.h, [314](#)
- constraint_ifunc_2
 - tempo2.h, [314](#)
- constraint_ifunc_year_cos
 - tempo2.h, [314](#)
- constraint_ifunc_year_cos2
 - tempo2.h, [314](#)
- constraint_ifunc_year_sin
 - tempo2.h, [314](#)
- constraint_ifunc_year_sin2
 - tempo2.h, [314](#)
- constraint_ifunc_year_xcos
 - tempo2.h, [314](#)
- constraint_ifunc_year_xsin
 - tempo2.h, [314](#)
- constraint_label
 - tempo2.h, [313](#)
- constraint_qifunc_c_year_cos
 - tempo2.h, [315](#)
- constraint_qifunc_c_year_cos2
 - tempo2.h, [315](#)
- constraint_qifunc_c_year_sin
 - tempo2.h, [315](#)
- constraint_qifunc_c_year_sin2
 - tempo2.h, [315](#)
- constraint_qifunc_c_year_xcos
 - tempo2.h, [315](#)
- constraint_qifunc_c_year_xsin
 - tempo2.h, [315](#)
- constraint_qifunc_p_year_cos
 - tempo2.h, [315](#)
- constraint_qifunc_p_year_cos2
 - tempo2.h, [315](#)
- constraint_qifunc_p_year_sin
 - tempo2.h, [314](#)
- constraint_qifunc_p_year_sin2
 - tempo2.h, [315](#)
- constraint_qifunc_p_year_xcos
 - tempo2.h, [315](#)
- constraint_qifunc_p_year_xsin
 - tempo2.h, [315](#)
- constraint_quad_ifunc_c_0
 - tempo2.h, [314](#)
- constraint_quad_ifunc_c_1
 - tempo2.h, [314](#)
- constraint_quad_ifunc_c_2
 - tempo2.h, [314](#)
- constraint_quad_ifunc_p_0
 - tempo2.h, [314](#)
- constraint_quad_ifunc_p_1
 - tempo2.h, [314](#)
- constraint_quad_ifunc_p_2
 - tempo2.h, [314](#)
- constraint_tel_dx_0
 - tempo2.h, [314](#)
- constraint_tel_dx_1
 - tempo2.h, [314](#)
- constraint_tel_dx_2
 - tempo2.h, [314](#)
- constraint_tel_dy_0
 - tempo2.h, [314](#)
- constraint_tel_dy_1
 - tempo2.h, [314](#)
- constraint_tel_dy_2
 - tempo2.h, [314](#)
- constraint_tel_dz_0
 - tempo2.h, [314](#)
- constraint_tel_dz_1
 - tempo2.h, [314](#)
- constraint_tel_dz_2
 - tempo2.h, [314](#)
- constraintCounters
 - FitInfo, [32](#)
- constraintDerivFunc
 - tempo2.h, [313](#)

- constraintDerivs
 - FitInfo, 32
- constraintIndex
 - FitInfo, 32
- constraints
 - pulsar, 60
- constraints.C, 101
 - autoConstraints, 102
 - autosetDMCM, 102
 - CONSTRAINTfuncs, 102
 - computeConstraintWeights, 102
 - consFunc_dmmodel_cw, 102
 - consFunc_dmmodel_cw_year, 102
 - consFunc_dmmodel_dm1, 102
 - consFunc_dmmodel_mean, 102
 - consFunc_ifunc, 102
 - consFunc_ifunc_year, 102
 - consFunc_qifunc_c_year, 102
 - consFunc_qifunc_p_year, 102
 - consFunc_quad_ifunc_c, 102
 - consFunc_quad_ifunc_p, 102
 - consFunc_tel_dx, 102
 - consFunc_tel_dy, 102
 - consFunc_tel_dz, 102
 - get_constraint_name, 102
 - getConstraintDeriv, 103
 - matrixDMConstraintWeights, 103
 - standardConstraintFunctions, 103
- constraints.h, 103
 - autosetDMCM, 104
 - CONSTRAINTfuncs, 104
 - computeConstraintWeights, 104
 - consFunc_dmmodel_cw, 104
 - consFunc_dmmodel_cw_year, 104
 - consFunc_dmmodel_dm1, 104
 - consFunc_dmmodel_mean, 104
 - consFunc_ifunc, 104
 - consFunc_ifunc_year, 104
 - consFunc_qifunc_c_year, 104
 - consFunc_qifunc_p_year, 104
 - consFunc_quad_ifunc_c, 104
 - consFunc_quad_ifunc_p, 104
 - consFunc_tel_dx, 104
 - consFunc_tel_dy, 104
 - consFunc_tel_dz, 104
 - get_constraint_name, 104
 - standardConstraintFunctions, 104
- convert_gravWaveBackground_fit
 - sigmaz_plug.C, 234
- convert_gravWaveBackground_noFit
 - sigmaz_plug.C, 235
- convertEcliptic
 - sw_delay.C, 262
- convertTOA
 - toa2utc.C, 363
- convertUnits
 - readEphemeris_calceph.C, 254
- convertXY_celestial
 - GWanisobkgrd_plug.C, 195
 - GWbkgrd_plug.C, 197
 - GWbkgrdfromfile_plug.C, 198
 - GWdipolebkgrd_plug.C, 201
 - GWgeneralanisobkgrd_plug.C, 206
 - GWgeneralbkgrd_plug.C, 207
 - GWsingle_plug.C, 212
- copyPSR
 - tempo2.h, 318
 - tempo2Util.C, 332
- copyParam
 - tempo2.h, 318
 - tempo2Util.C, 332
- corr2pt
 - checkWhite_plug.C, 160
- correctTroposphere
 - pulsar, 60
- correction
 - clock_correction, 29
- correctionTT_TB
 - observation, 45
- correctionTT_Teph
 - observation, 45
- correctionUT1
 - observation, 45
- correctionsTT
 - observation, 45
- corrects_to
 - clock_correction, 29
- cosineFunc
 - detectGWB_plug.C, 169
- cosl
 - TKlongdouble.float128.h, 345
 - TKlongdouble.h, 347
- covar
 - pulsar, 61
- covarFuncFile
 - clock_plug.C, 162
 - global.C, 126
 - plk_plug.C, 228
 - spectrum_plug.C, 244
 - spk_plug.C, 246
 - tempo2.h, 322
- covarFuncFile2
 - plotMany_plug.C, 230
- cpgpt
 - fermi_plug.C, 178
- createGWcovarianceFunction
 - icLimit_plug.C, 215
- createNewArrivalTimes
 - delays_plug.C, 166
- cumulativeHistogram
 - GWwhiteLimit_plug.C, 213
- cumulativeHistogram2
 - GWwhiteLimit_plug.C, 213
- curr_cache_loc
 - jpl_eph_data, 39
- DDGRmodel

- DDGRmodel.C, 105
- tempo2.h, 318
- DDGRmodel.C, 104
 - DDGRmodel, 105
 - mass2dd, 105
 - updateDDGR, 105
- DDHmodel
 - DDHmodel.C, 106
 - tempo2.h, 318
- DDHmodel.C, 105
 - DDHmodel, 106
 - updateDDH, 106
- DDKmodel
 - DDKmodel.C, 107
 - tempo2.h, 318
- DDKmodel.C, 106
 - DDKmodel, 107
 - updateDDK, 107
- DDSmodel
 - DDSmodel.C, 109
 - tempo2.h, 318
- DDSmodel.C, 108
 - DDSmodel, 109
 - updateDDS, 109
- DDmodel
 - DDmodel.C, 108
 - tempo2.h, 318
- DDmodel.C, 107
 - DDmodel, 108
 - updateDD, 108
- DEPRECATED
 - TKlog.h, 342
- DLL_FUNC
 - jpleph.h, 140
- DM_CONST
 - tempo2.h, 307
- DM_CONST_SI
 - tempo2.h, 307
- dadt
 - GWsim.C, 129
 - GWsim.h, 132
- data
 - DynamicArray, 30
 - sigmaz_plug.C, 235
 - TKspectrum.C, 355
- date2mjd
 - readJBO_bat.C, 255
- date_string
 - T1Polyco, 74
- dayGap
 - fixData_plug.C, 181
- dcmFile
 - calcDMe_plug.C, 157
 - global.C, 126
 - plk_plug.C, 228
 - spectrum_plug.C, 244
 - spk_plug.C, 246
 - tempo2.h, 322
- ddm
 - calcDMe_plug.C, 157
- ddmCount
 - calcDMe_plug.C, 157
- ddmErr
 - calcDMe_plug.C, 157
- ddmMJD
 - calcDMe_plug.C, 157
- debugFlag
 - TKlog.C, 340
 - TKlog.h, 343
- decjStrPost
 - pulsar, 61
- decjStrPre
 - pulsar, 61
- decsim
 - pulsar, 61
- dedt
 - GWsim.C, 129
 - GWsim.h, 132
- defineClockCorrectionSequence
 - clkcorr.C, 98
 - tempo2.h, 318
- defineGlitchVal
 - glitch_plug.C, 188
- delayCorr
 - observation, 45
- delays_plug.C
 - callFit, 166
 - createNewArrivalTimes, 166
 - deletePoint, 166
 - doPlot, 166
 - findMax, 166
 - findMean, 166
 - findMin, 166
 - fortranMod, 166
 - graphicalInterface, 166
 - help, 166
 - idPoint, 166
 - MAX_HIGHLIGHT, 166
 - plugVersionCheck, 166
- deleteFileName
 - pulsar, 61
- deletePoint
 - delays_plug.C, 166
 - plk_plug.C, 227
 - spk_plug.C, 245
- deleted
 - observation, 45
- deriveKeplerian
 - T2model.C, 288
- derivePostKeplerian
 - T2model.C, 288
- derivs
 - age.C, 79
- describe
 - calcDMe_plug.C, 156
- designmatrix_plug.C

- graphicalInterface, 167
- help, 167
- ProcessTempo2Objects, 167
- tempo2_GetNumberOfParameters, 167
- WriteDesignMatrix, 167
- destroyMemory
 - initialise.C, 137
 - tempo2.h, 318
- destroyOne
 - initialise.C, 137
 - tempo2.h, 318
- detectGWB_plug.C
 - calcSpectra_plugin, 169
 - cosineFunc, 169
 - formCholeskyMatrixPlugin, 169
 - GLOBAL_COSVAL, 169
 - GLOBAL_MEANSUB, 169
 - getSpectrum, 169
 - graphicalInterface, 169
 - hdfunc, 169
 - hdfunc_cosineSub, 169
 - hdfunc_meanSub, 169
 - hdfunc_offs, 169
 - hdfunc_removeCosine, 169
 - help, 169
 - OMEGA0, 169
 - psrangle, 169
 - write_debug_files, 169
- detectGWBnew_plug.C
 - calcSpectra_plugin, 170
 - fitMeanSineFunc, 170
 - fitPolyFunc, 170
 - formCholeskyMatrixPlugin, 170
 - getSpectrum, 170
 - graphicalInterface, 171
 - hdfunc, 171
 - help, 171
 - notim, 171
 - OMEGA0, 171
 - offsetToCM, 171
 - psrangle, 171
 - write_debug_files, 171
 - write_python_files, 171
- detectSource
 - GWsens_plug.C, 210
- determine1dStructureFunction
 - fixData_plug.C, 181
- dgemm
 - doFit.C, 113
- dgemm_
 - doFit.C, 113
- dgemm_ctof
 - doFit.C, 113
- dgemm_ftoc
 - doFit.C, 113
- dgemv
 - doFit.C, 113
- dgemv_
 - doFit.C, 113
- dgemv_ctof
 - doFit.C, 113
- dgemv_ftoc
 - doFit.C, 113
- dgesvd
 - doFit.C, 113
- dgesvd_
 - doFit.C, 113
- dgesvd_ctof
 - doFit.C, 113
- dgesvd_ftoc
 - doFit.C, 113
- dglep
 - textOutput.C, 334
- dilateFreq
 - pulsar, 61
- dispParameter
 - publish_plug.C, 232
- dispersion_constant
 - ChebyModel, 26
- display
 - calcDMe_plug.C, 156
- displayCVSversion
 - global.C, 126
 - tempo2.h, 322
- displayMsg
 - tempo2.h, 318
 - tempo2Util.C, 332
- displayParameters
 - displayParameters.C, 110
 - tempo2.h, 318
- displayParameters.C, 109
 - displayParameters, 110
- displayStatistics
 - plk_plug.C, 227
- dist_bin
 - gwSrc, 36
 - gwgeneralSrc, 34
- dm
 - T1Polyco, 74
- dm0
 - calcDMe_plug.C, 157
- dm0_err
 - calcDMe_plug.C, 157
- dm_delays
 - dm_delays.C, 111
 - tempo2.h, 318
- dm_delays.C, 110
 - dm_delays, 111
 - solarWindModel, 111
- dm_plug.C
 - doPlot, 172
 - graphicalInterface, 172
 - help, 172
 - MAX_TIMES, 172
 - mjd2year, 172
 - plugVersionCheck, 172

- selectData, 172
- slaCalyd, 172
- slaClyd, 172
- dmCount
 - calcDMe_plug.C, 157
- dmObs
 - calcDMe_plug.C, 157
- dmOffset
 - pulsar, 61
- dmmodel_fitFunc_plug.C
 - getFitLabels, 173
 - pluginFitFunc, 173
 - updateDMvals, 173
- dmoffsCM
 - pulsar, 61
- dmoffsCM_error
 - pulsar, 61
- dmoffsCM_mjd
 - pulsar, 61
- dmoffsCM_weight
 - pulsar, 61
- dmoffsCMnum
 - pulsar, 61
- dmoffsDM
 - pulsar, 61
- dmoffsDM_error
 - pulsar, 61
- dmoffsDM_mjd
 - pulsar, 61
- dmoffsDM_weight
 - pulsar, 61
- dmoffsDMnum
 - pulsar, 61
- dms_turn
 - tempo2.h, 319
 - tempo2Util.C, 332
 - tempo2Util.h, 333
- doDisplay
 - calcDMe_plug.C, 157
- doFit
 - doFit.C, 113
 - tempo2.h, 319
- doFit.C, 111
 - dgemm, 113
 - dgemm_, 113
 - dgemm_ctof, 113
 - dgemm_ftoc, 113
 - dgemv, 113
 - dgemv_, 113
 - dgemv_ctof, 113
 - dgemv_ftoc, 113
 - dgesvd, 113
 - dgesvd_, 113
 - dgesvd_ctof, 113
 - dgesvd_ftoc, 113
 - doFit, 113
 - doFitAll, 113
 - doFitDCM, 113
 - doFitOLD, 113
 - dpotrf, 113
 - dpotrf_, 114
 - dpotrf_ctof, 114
 - dpotrf_ftoc, 114
 - dpotri, 114
 - dpotri_, 114
 - dpotri_ctof, 114
 - dpotri_ftoc, 114
 - FITfuncs, 114
 - getConstraintDeriv, 114
 - getNglobal, 114
 - getNparams, 114
 - getParamDeriv, 114
 - getTempoNestMaxLike, 114
 - globalFITfuncs, 114
 - othpl, 114
 - updateGlobalParameters, 114
 - updateParameters, 114
- doFitAll
 - doFit.C, 113
 - tempo2.h, 319
- doFitDCM
 - doFit.C, 113
 - tempo2.h, 319
- doFitGlobal
 - tempo2.h, 319
- doFitOLD
 - doFit.C, 113
- doGenPlot
 - GWbkgrdfromfile_plug.C, 198
- doPlot
 - delays_plug.C, 166
 - dm_plug.C, 172
 - GWanisobkgrd_plug.C, 195
 - GWbkgrd_plug.C, 197
 - GWbkgrdfromfile_plug.C, 198
 - GWdipolebkgrd_plug.C, 201
 - GWgeneralanisobkgrd_plug.C, 206
 - GWgeneralbkgrd_plug.C, 207
 - GWsingle_plug.C, 212
 - glitch_plug.C, 188
 - plk_plug.C, 227
 - plotMany_plug.C, 229
 - simulDM_plug.C, 239
 - splk_plug.C, 245
- doPlugin
 - GWsens_plug.C, 210
 - planet_plug.C, 223
 - simRedNoise_plug.C, 238
 - spectralModel_plug.C, 241
 - spectrum_plug.C, 244
- doPlugin1
 - fixData_plug.C, 181
- doPlugin2
 - fixData_plug.C, 181
- doPlugin3
 - fixData_plug.C, 181

- doSummary
 - fixData_plug.C, 181
- documentation/DEVELOPER_GUIDE.md, 111
- documentation/USER_GUIDE.md, 111
- documentation/developers.md, 111
- documentation/directories.md, 111
- doplot
 - sigmaz_plug.C, 235
- doplugin
 - simulDM_plug.C, 239
- doppler
 - T1Polyco, 74
- dotProduct
 - GWsim.C, 129
 - GWsim.h, 132
- dotproduct
 - tempo2.h, 319
 - tempo2Util.C, 332
- dpotrf
 - doFit.C, 113
- dpotrf_
 - doFit.C, 114
- dpotrf_ctof
 - doFit.C, 114
- dpotrf_ftoc
 - doFit.C, 114
- dpotri
 - doFit.C, 114
- dpotri_
 - doFit.C, 114
- dpotri_ctof
 - doFit.C, 114
- dpotri_ftoc
 - doFit.C, 114
- draw_grid
 - GWanisobkgrd_plug.C, 195
 - GWbkgrd_plug.C, 197
 - GWbkgrdfromfile_plug.C, 198
 - GWdipolebkgrd_plug.C, 201
 - GWgeneralanisobkgrd_plug.C, 206
 - GWgeneralbkgrd_plug.C, 207
 - GWsingle_plug.C, 212
- drawAxisSel
 - plk_plug.C, 227
- drawMenu
 - glitch_plug.C, 188
 - plk_plug.C, 227
 - spectrum_plug.C, 244
- drawMenu3
 - plk_plug.C, 227
- drawMenu3_2
 - plk_plug.C, 227
- drawOption
 - plk_plug.C, 227
 - spectrum_plug.C, 244
- dtdt
 - GWsim.C, 129
 - GWsim.h, 132
- dut1
 - EOPSample, 31
- dxsav
 - GWevolve_plug.C, 204
- DynamicArray, 30
 - data, 30
 - elem_size, 30
 - nallocated, 30
 - nelem, 30
- DynamicArray_free
 - dynarr.C, 115
 - dynarr.h, 116
- DynamicArray_init
 - dynarr.C, 115
 - dynarr.h, 116
- DynamicArray_push_back
 - dynarr.C, 115
 - dynarr.h, 116
- DynamicArray_resize
 - dynarr.C, 115
 - dynarr.h, 116
- dynarr.C, 114
 - DynamicArray_free, 115
 - DynamicArray_init, 115
 - DynamicArray_push_back, 115
 - DynamicArray_resize, 115
- dynarr.h, 116
 - DynamicArray_free, 116
 - DynamicArray_init, 116
 - DynamicArray_push_back, 116
 - DynamicArray_resize, 116
- e
 - sample, 73
- ECLIPTIC_OBLIQUITY
 - global.C, 126
 - tempo2.h, 323
- ECLIPTIC_OBLIQUITY_VAL
 - tempo2.h, 307
- ELL1Hmodel
 - ELL1Hmodel.C, 117
 - tempo2.h, 319
- ELL1Hmodel.C, 117
 - ELL1Hmodel, 117
 - updateELL1H, 117
- ELL1model
 - ELL1model.C, 118
 - tempo2.h, 319
- ELL1model.C, 117
 - ELL1model, 118
 - updateELL1, 118
- ENDERR
 - TKlog.h, 342
- ENDL
 - TKlog.h, 342
- EOPSample, 30
 - dut1, 31
 - mjd, 31
 - xp, 31

- yp, [31](#)
- EPS
 - efacEquad_plug.C, [175](#)
- EPS1
 - efacEquad_plug.C, [175](#)
- EPS2
 - efacEquad_plug.C, [175](#)
- ERRCON
 - GWevolve_plug.C, [203](#)
- ERRORCOLOR
 - TKlog.h, [342](#)
- EXPSMOOTH
 - choleskyRoutines.h, [97](#)
 - global.C, [126](#)
- earth_ssb
 - observation, [45](#)
- earthMoonBary_earth
 - observation, [45](#)
- earthMoonBary_ssb
 - observation, [45](#)
- eccRes
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- eccResWithEnergy
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- eclCoord
 - pulsar, [61](#)
- efac
 - observation, [46](#)
- efacEquad_plug.C
 - calcEfacedEquad, [175](#)
 - calcEfacedEquad2, [175](#)
 - EPS, [175](#)
 - EPS1, [175](#)
 - EPS2, [175](#)
 - erff, [175](#)
 - FMAX, [175](#)
 - FPMIN, [175](#)
 - FREE_ARG, [175](#)
 - free_ivector, [175](#)
 - gammln, [175](#)
 - gammp, [175](#)
 - gaussFunc, [175](#)
 - gcf, [175](#)
 - graphicalInterface, [175](#)
 - gser, [175](#)
 - help, [175](#)
 - ITMAX, [175](#)
 - ivector, [176](#)
 - ksone, [176](#)
 - kstwo, [176](#)
 - M, [175](#)
 - NR_END, [175](#)
 - NSTACK, [175](#)
 - nerror, [176](#)
 - plugVersionCheck, [176](#)
 - probs, [176](#)
 - SWAP, [175](#)
 - sort, [176](#)
- einsteinRate
 - observation, [46](#)
- elem_size
 - DynamicArray, [30](#)
- elsun2
 - sw_delay.C, [262](#)
- emrat
 - jpl_eph_data, [39](#)
- endJD
 - IFTEphemeris, [38](#)
- eop.C, [118](#)
 - get_EOP, [119](#)
 - load_EOP, [119](#)
- eopc04_file
 - pulsar, [61](#)
- ephem_end
 - jpl_eph_data, [39](#)
- ephem_start
 - jpl_eph_data, [39](#)
- ephem_step
 - jpl_eph_data, [39](#)
- ephemeris
 - pulsar, [62](#)
- ephemeris_routines
 - global.C, [126](#)
 - tempo2.C, [299](#)
- ephemeris_routines_fermi
 - fermi_plug.C, [178](#)
- ephemeris_version
 - jpl_eph_data, [39](#)
- ephver
 - IFTEphemeris, [38](#)
- epsilon
 - lm_control_struct, [40](#)
- equ2ecl
 - tempo2.h, [319](#)
 - tempo2Util.C, [332](#)
- equad
 - observation, [46](#)
- erff
 - efacEquad_plug.C, [175](#)
- err
 - parameter, [52](#)
- extra_delays
 - global.C, [126](#)
 - tempo2.C, [299](#)
- extra_delays_fermi
 - fermi_plug.C, [178](#)
- f
 - IFTEphemeris, [38](#)
 - lmcurve_data_struct, [41](#)
- f0_0
 - calcDMe_plug.C, [157](#)
- f0_0_err
 - calcDMe_plug.C, [157](#)
- f0fit

- calcDMe_plug.C, 157
- F77_FUNC
 - config.h, 100
- F77_FUNC_
 - config.h, 100
- F77_dgels
 - T2accel.C, 266, 267
- F77_dgemm
 - T2accel.C, 266, 267
- F77_dgemv
 - T2accel.C, 266, 267
- F77_dpotf2
 - T2accel.C, 267
- F77_dtptri
 - T2accel.C, 267
- F77_dtrmm
 - T2accel.C, 267
- FALSE
 - jpleph.c, 139
- FB90_TIMEEPH
 - tempo2.h, 307
- FB_deltaT
 - tt2tdb.C, 366
- FCALPHA
 - choleskyRoutines.h, 97
 - global.C, 126
- FCFINAL
 - choleskyRoutines.h, 97
 - global.C, 126
- FITWAVES_harmonicStep
 - plk_plug.C, 228
- FITWAVES_n
 - plk_plug.C, 228
- FITWAVES_omega
 - plk_plug.C, 228
- FITWAVES_par
 - plk_plug.C, 228
- FITfuncs
 - doFit.C, 114
 - tempo2.h, 319
- FMAX
 - efacEquad_plug.C, 175
 - GWevolve_plug.C, 203
- FMT_LD
 - TKlongdouble.float128.h, 345
 - TKlongdouble.h, 347
- FPMIN
 - efacEquad_plug.C, 175
- FREE_ARG
 - efacEquad_plug.C, 175
 - GWevolve_plug.C, 203
 - glast_plug.C, 185
 - interpolate_plug.C, 217
- fabsl
 - TKlongdouble.float128.h, 345
 - TKlongdouble.h, 347
- fake_plug.C
 - callFit, 177
- graphicalInterface, 177
- plugVersionCheck, 177
- fang
 - observatory.C, 143
- Fe
 - GWsim.C, 129
 - GWsim.h, 132
- fermi_plug.C
 - clock_corrections_fermi, 178
 - cpgpt, 178
 - ephemeris_routines_fermi, 178
 - extra_delays_fermi, 178
 - formBatsAll_fermi, 178
 - graphicalInterface, 178
 - HTest, 178
 - inner_product, 178
 - met2mjd, 178
 - mjd2met, 178
 - outer_product, 178
 - plugVersionCheck, 178
 - SECDAY, 178
- fileName
 - TabulatedFunction, 77
- fileOutput2
 - planet_plug.C, 223
 - spectralModel_plug.C, 241
- fileOutput3
 - planet_plug.C, 223
 - spectralModel_plug.C, 241
- filterStr
 - pulsar, 62
- find_event_hdu
 - photons_plug.C, 221
- findAngle
 - sw_delay.C, 262
- findCW_plug.C
 - graphicalInterface, 179
 - help, 179
 - plugVersionCheck, 179
- findCWs_plug.C
 - graphicalInterface, 180
 - help, 180
- findFirst
 - calcDMe_plug.C, 156
- findMax
 - delays_plug.C, 166
 - plotMany_plug.C, 229
 - spk_plug.C, 245
- findMaxVal
 - plotMany_plug.C, 229
- findMaxY
 - plk_plug.C, 227
- findMean
 - calcDMe_plug.C, 156
 - delays_plug.C, 166
 - plk_plug.C, 227
 - plotMany_plug.C, 230
 - spk_plug.C, 245

- findMeanD
 - plk_plug.C, [227](#)
- findMin
 - delays_plug.C, [166](#)
 - plotMany_plug.C, [230](#)
 - splk_plug.C, [245](#)
- findMinVal
 - plotMany_plug.C, [230](#)
- findMinY
 - plk_plug.C, [227](#)
- findOverlap
 - compareDsets_plug.C, [165](#)
- findSessions
 - calcDMe_plug.C, [156](#)
- findSmoothCurve
 - planet_plug.C, [223](#)
 - spectralModel_plug.C, [241](#)
- Findphi
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- finish_sessions
 - calcDMe_plug.C, [157](#)
- fit4
 - sigmaz_plug.C, [235](#)
 - TKspectrum.C, [354](#)
 - TKspectrum.h, [359](#)
- fitChisq
 - pulsar, [62](#)
- fitCosSineFunc
 - TKspectrum.C, [354](#)
 - TKspectrum.h, [359](#)
- fitCount
 - calcDMe_plug.C, [157](#)
- fitExponential
 - planet_plug.C, [223](#)
 - spectralModel_plug.C, [241](#)
- fitFlag
 - parameter, [52](#)
- fitFunc
 - pulsar, [62](#)
- fitFuncs
 - glitch_plug.C, [188](#)
- FitInfo, [31](#)
 - constraintCounters, [32](#)
 - constraintDerivs, [32](#)
 - constraintIndex, [32](#)
 - nConstraints, [32](#)
 - nParams, [32](#)
 - paramCounters, [32](#)
 - paramDerivs, [32](#)
 - paramIndex, [32](#)
 - tempo2.h, [313](#)
 - updateFunctions, [32](#)
- fitJump
 - pulsar, [62](#)
- fitMeanSineFunc
 - detectGWBnew_plug.C, [170](#)
 - TKspectrum.C, [354](#)
 - TKspectrum.h, [359](#)
- fitMeanSineFunc_IFUNC
 - TKspectrum.C, [354](#)
 - TKspectrum.h, [359](#)
- fitMode
 - pulsar, [62](#)
- fitNfree
 - pulsar, [62](#)
- fitObs
 - calcDMe_plug.C, [157](#)
- fitParamGlobalI
 - pulsar, [62](#)
- fitParamGlobalK
 - pulsar, [62](#)
- fitParamI
 - pulsar, [62](#)
- fitParamK
 - pulsar, [62](#)
- fitPolyFunc
 - detectGWBnew_plug.C, [170](#)
- fitSineFunc
 - planet_plug.C, [223](#)
 - spectralModel_plug.C, [241](#)
- fitf0
 - glitchS, [32](#)
- fitf0d
 - glitchS, [32](#)
- fitf1
 - glitchS, [32](#)
- fitinfo
 - pulsar, [62](#)
- fitph
 - glitchS, [33](#)
- fittd
 - glitchS, [33](#)
- fitv
 - sigmaz_plug.C, [235](#)
- fitwave_function
 - glast_plug.C, [185](#)
- fixData_plug.C
 - alpha, [181](#)
 - dayGap, [181](#)
 - determine1dStructureFunction, [181](#)
 - doPlugin1, [181](#)
 - doPlugin2, [181](#)
 - doPlugin3, [181](#)
 - doSummary, [181](#)
 - graphicalInterface, [181](#)
 - gwamp, [181](#)
 - help, [181](#)
 - nit, [181](#)
 - plotHistogram, [181](#)
 - plotout, [181](#)
 - plotoutSet, [181](#)
 - plugVersionCheck, [182](#)
 - script, [182](#)
- fixDec
 - publish_plug.C, [232](#)

- fixRA
 - publish_plug.C, 232
- fixedFormat
 - pulsar, 62
- fjumpID
 - pulsar, 62
- flagID
 - observation, 46
- flagStore
 - plk_plug.C, 228
- flagVal
 - observation, 46
- floorl
 - TKlongdouble.float128.h, 345
 - TKlongdouble.h, 347
- fname
 - observation, 46
- fnorm
 - lm_status_struct, 41
- forceGlobalFit
 - global.C, 126
 - tempo2.h, 323
- formBats
 - formBats.C, 120
 - tempo2.h, 319
- formBats.C, 119
 - formBats, 120
- formBatsAll
 - global.C, 126
 - tempo2.h, 319
- formBatsAll_fermi
 - fermi_plug.C, 178
- formCholeskyMatrix2
 - globalDCM_fitFunc_plug.C, 193
- formCholeskyMatrixPlugin
 - detectGWB_plug.C, 169
 - detectGWBnew_plug.C, 170
 - icLimit_plug.C, 215
- formResiduals
 - formResiduals.C, 121
 - tempo2.h, 319
- formResiduals.C, 120
 - averageResiduals, 121
 - formResiduals, 121
 - residualTracking, 121
- fortran_mod
 - tempo2.h, 319
 - tempo2Util.C, 332
- fortran_nint
 - tempo2.h, 319
 - tempo2Util.C, 332
- fortran_nlong
 - tempo2.h, 319
 - tempo2Util.C, 332
- fortranMod
 - applet_plug.C, 148
 - delays_plug.C, 166
 - general2_plug.C, 182
 - plk_plug.C, 227
 - plotMany_plug.C, 230
 - splk_plug.C, 245
- free_2dLL
 - TKmatrix.C, 350
- free_2df
 - TKmatrix.C, 350
 - TKmatrix.h, 351
- free_blas
 - TKmatrix.C, 350
 - TKmatrix.h, 351
- free_ivecior
 - efacEquad_plug.C, 175
 - glast_plug.C, 185
- free_uinv
 - TKmatrix.C, 350
 - TKmatrix.h, 351
- free_vector
 - GWevolve_plug.C, 204
 - interpolate_plug.C, 217
- freq
 - observation, 46
- freq1f
 - calcDMe_plug.C, 157
- freq2f
 - calcDMe_plug.C, 158
- freq_end
 - ChebyModel, 26
- freq_start
 - ChebyModel, 26
- freqArray
 - calcDMe_plug.C, 158
- freqOffset
 - calcDMe_plug.C, 158
- freqSSB
 - observation, 46
- frequency_cheby
 - ChebyModel, 26
- frequency_obs
 - T1Polyco, 74
- frequency_psr_0
 - T1Polyco, 74
- ftol
 - lm_control_struct, 40
- G_OMEGA
 - planet_plug.C, 224
 - spectralModel_plug.C, 242
- GLOBAL_COSVAL
 - detectGWB_plug.C, 169
- GLOBAL_MEANSUB
 - detectGWB_plug.C, 169
- GLOBAL_OMEGA
 - TKspectrum.C, 355
 - TKspectrum.h, 360
- GM
 - tempo2.h, 307
- GM_C3
 - tempo2.h, 307

- GMJ_C3
 - tempo2.h, [307](#)
- GMN_C3
 - tempo2.h, [307](#)
- GMS_C3
 - tempo2.h, [307](#)
- GMU_C3
 - tempo2.h, [307](#)
- GMV_C3
 - tempo2.h, [308](#)
- GRS80_A
 - bary_plug.C, [152](#)
 - observatory.C, [143](#)
- GRS80_F
 - bary_plug.C, [152](#)
 - observatory.C, [143](#)
- GRS80_to_ITRF
 - observatory.C, [143](#)
- GWanisobkgrd_plug.C
 - convertXY_celestial, [195](#)
 - doPlot, [195](#)
 - draw_grid, [195](#)
 - getTspan, [195](#)
 - graphicalInterface, [196](#)
 - help, [196](#)
 - plotPosn, [196](#)
 - plotResiduals, [196](#)
 - plotSpectrum, [196](#)
 - plugVersionCheck, [196](#)
- GWanisotropicbackground
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWbackground
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWbackground_read
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWbackground_write
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWbkgrd_plug.C
 - convertXY_celestial, [197](#)
 - doPlot, [197](#)
 - draw_grid, [197](#)
 - getTspan, [197](#)
 - graphicalInterface, [197](#)
 - help, [197](#)
 - plotPosn, [197](#)
 - plotResiduals, [197](#)
 - plotSpectrum, [197](#)
 - plugVersionCheck, [197](#)
- GWbkgrdfromfile_plug.C
 - convertXY_celestial, [198](#)
 - doGenPlot, [198](#)
 - doPlot, [198](#)
 - draw_grid, [198](#)
 - getTspan, [198](#)
 - graphicalInterface, [199](#)
 - help, [199](#)
 - NGWmax, [199](#)
 - plotGenPosn, [199](#)
 - plotGenSpectrum, [199](#)
 - plotPosn, [199](#)
 - plotResiduals, [199](#)
 - plotSpectrum, [199](#)
 - plugVersionCheck, [199](#)
- GWdetect_plug.C
 - graphicalInterface, [200](#)
 - help, [200](#)
 - searchGridPos, [200](#)
- GWdipolebackground
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWdipolebkgrd_plug.C
 - convertXY_celestial, [201](#)
 - doPlot, [201](#)
 - draw_grid, [201](#)
 - getTspan, [201](#)
 - graphicalInterface, [201](#)
 - help, [201](#)
 - plotPosn, [201](#)
 - plotResiduals, [201](#)
 - plotSpectrum, [201](#)
 - plugVersionCheck, [201](#)
- GWevolve_plug.C
 - BIG_G, [203](#)
 - calcAmp, [204](#)
 - const2, [204](#)
 - constA0, [204](#)
 - dxsav, [204](#)
 - ERRCON, [203](#)
 - FMAX, [203](#)
 - FREE_ARG, [203](#)
 - free_vector, [204](#)
 - graphicalInterface, [204](#)
 - help, [204](#)
 - kmax, [204](#)
 - kount, [204](#)
 - MAX_VAL, [203](#)
 - MAXSTP, [203](#)
 - NR_END, [203](#)
 - nrerror, [204](#)
 - ode, [204](#)
 - PCM, [203](#)
 - PGROW, [203](#)
 - PSHRNK, [203](#)
 - plugVersionCheck, [204](#)
 - psrange, [204](#)
 - RungeKuttaCashKarp, [204](#)
 - RungeKuttaStep, [204](#)
 - SAFETY, [203](#)
 - SIGN, [203](#)
 - SOLAR_MASS, [203](#)
 - SPEED_LIGHT, [204](#)
 - setup3C66B, [204](#)

- setupTest, [204](#)
 - TINY, [204](#)
 - ThetaEderivs, [204](#)
 - vector, [204](#)
 - xp, [205](#)
 - yp, [205](#)
- GWgeneralanisobkgrd_plug.C
 - convertXY_celestial, [206](#)
 - doPlot, [206](#)
 - draw_grid, [206](#)
 - getTspan, [206](#)
 - graphicalInterface, [206](#)
 - help, [206](#)
 - plotPosn, [206](#)
 - plotResiduals, [206](#)
 - plotSpectrum, [206](#)
 - plugVersionCheck, [206](#)
- GWgeneralanisotropicbackground
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWgeneralbackground
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWgeneralbackground_read
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWgeneralbackground_write
 - GWsim.C, [129](#)
 - GWsim.h, [132](#)
- GWgeneralbkgrd_plug.C
 - convertXY_celestial, [207](#)
 - doPlot, [207](#)
 - draw_grid, [207](#)
 - getTspan, [207](#)
 - graphicalInterface, [207](#)
 - help, [207](#)
 - plotPosn, [207](#)
 - plotResiduals, [207](#)
 - plotSpectrum, [207](#)
 - plugVersionCheck, [207](#)
- GWsens_plug.C
 - detectSource, [210](#)
 - doPlugin, [210](#)
 - getSensCurv, [210](#)
 - graphicalInterface, [210](#)
 - help, [210](#)
 - plugVersionCheck, [211](#)
- GWsim.C, [127](#)
 - calculateResidualGW, [129](#)
 - calculateResidualgeneralGW, [129](#)
 - dadt, [129](#)
 - dedt, [129](#)
 - dotProduct, [129](#)
 - dtdt, [129](#)
 - eccRes, [129](#)
 - eccResWithEnergy, [129](#)
 - Fe, [129](#)
 - Findphi, [129](#)
 - GWanisotropicbackground, [129](#)
 - GWbackground, [129](#)
 - GWbackground_read, [129](#)
 - GWbackground_write, [129](#)
 - GWdipolebackground, [129](#)
 - GWgeneralanisotropicbackground, [129](#)
 - GWgeneralbackground, [129](#)
 - GWgeneralbackground_read, [129](#)
 - GWgeneralbackground_write, [129](#)
 - gwsim_Ngrid, [130](#)
 - matrixMult, [130](#)
 - psrange, [130](#)
 - Rs, [130](#)
 - setupGW, [130](#)
 - setupPulsar_GWsim, [130](#)
 - setupgeneralGW, [130](#)
 - sphharm, [130](#)
- GWsim.h, [130](#)
 - calculateResidualGW, [132](#)
 - calculateResidualgeneralGW, [132](#)
 - dadt, [132](#)
 - dedt, [132](#)
 - dotProduct, [132](#)
 - dtdt, [132](#)
 - eccRes, [132](#)
 - eccResWithEnergy, [132](#)
 - Fe, [132](#)
 - Findphi, [132](#)
 - GWanisotropicbackground, [132](#)
 - GWbackground, [132](#)
 - GWbackground_read, [132](#)
 - GWbackground_write, [132](#)
 - GWdipolebackground, [132](#)
 - GWgeneralanisotropicbackground, [132](#)
 - GWgeneralbackground, [132](#)
 - GWgeneralbackground_read, [132](#)
 - GWgeneralbackground_write, [132](#)
 - gwSrc, [132](#)
 - gwgenSpec, [132](#)
 - gwgeneralSrc, [132](#)
 - matrixMult, [132](#)
 - psrange, [133](#)
 - Rs, [133](#)
 - setupGW, [133](#)
 - setupPulsar_GWsim, [133](#)
 - setupgeneralGW, [133](#)
 - sphharm, [133](#)
- GWsingle_plug.C
 - convertXY_celestial, [212](#)
 - doPlot, [212](#)
 - draw_grid, [212](#)
 - getTspan, [212](#)
 - graphicalInterface, [212](#)
 - help, [212](#)
 - plotPosn, [212](#)
 - plotResiduals, [212](#)
 - plotSpectrum, [212](#)
 - plugVersionCheck, [212](#)

- GWwhiteLimit_plug.C
 - checkReal, [213](#)
 - cumulativeHistogram, [213](#)
 - cumulativeHistogram2, [213](#)
 - getLimits, [214](#)
 - getThreshold, [214](#)
 - GramSchmidt, [214](#)
 - graphicalInterface, [214](#)
 - help, [214](#)
 - MAX_FLAG, [213](#)
 - MAX_FREQ, [213](#)
 - MAX_ITERATION, [213](#)
 - MAX_POLY, [213](#)
 - plugVersionCheck, [214](#)
 - SIGN, [213](#)
 - setupPulsar, [214](#)
 - shuffle, [214](#)
 - sortit, [214](#)
 - storeVal, [214](#)
 - writeCommands, [214](#)
- gammln
 - efacEquad_plug.C, [175](#)
- gammp
 - efacEquad_plug.C, [175](#)
- gasdev
 - readEphemeris.C, [253](#)
- gaussFunc
 - efacEquad_plug.C, [175](#)
- gcf
 - efacEquad_plug.C, [175](#)
- general2_plug.C
 - fortranMod, [182](#)
 - nint_derived, [182](#)
 - parseLine, [183](#)
 - plugVersionCheck, [183](#)
 - rnd8, [183](#)
 - tempoOutput, [183](#)
- general_plug.C
 - nint_derived, [184](#)
 - parseLine, [184](#)
 - plugVersionCheck, [184](#)
 - rnd8, [184](#)
 - tempoOutput, [184](#)
- genrand_int32
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- genrand_real1
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- get_EOP
 - eop.C, [119](#)
 - tempo2.h, [319](#)
- get_OneobsCoord
 - tempo2.h, [319](#)
- get_binObs
 - calcDMe_plug.C, [156](#)
- get_blas_cols
 - TKmatrix.C, [350](#)
- TKmatrix.h, [351](#)
- get_blas_rows
 - TKmatrix.C, [350](#)
 - TKmatrix.h, [351](#)
- get_constraint_name
 - constraints.C, [102](#)
 - constraints.h, [104](#)
- get_mjdref
 - photons_plug.C, [221](#)
- get_obsCoord
 - get_obsCoord.C, [122](#)
 - tempo2.h, [319](#)
- get_obsCoord.C, [121](#)
 - ang, [122](#)
 - get_obsCoord, [122](#)
 - get_obsCoord_IAU2000B, [122](#)
 - get_precessionMatrix, [122](#)
 - iau_c2t00b_, [122](#)
 - iau_cp_, [122](#)
 - iau_pom00_, [123](#)
 - iau_pxp_, [123](#)
 - iau_rxp_, [123](#)
 - iau_sxp_, [123](#)
 - iau_trxp_, [123](#)
 - iau_trxpv_, [123](#)
 - lmst, [123](#)
 - remove_white, [123](#)
- get_obsCoord_IAU2000B
 - get_obsCoord.C, [122](#)
 - tempo2.h, [319](#)
- get_precessionMatrix
 - get_obsCoord.C, [122](#)
- getCholeskyDiagonals
 - cholesky.C, [90](#)
- getCholeskyMatrix
 - cholesky.C, [90](#)
 - tempo2.h, [319](#)
- getClockCorrectionSequence
 - clkcorr.C, [98](#)
- getClockCorrections
 - clkcorr.C, [98](#)
 - tempo2.h, [319](#)
- getConstraintDeriv
 - constraints.C, [103](#)
 - doFit.C, [114](#)
- getCorrection
 - clkcorr.C, [98](#)
 - tempo2.h, [319](#)
- getCorrectionTT
 - clkcorr.C, [98](#)
 - tempo2.h, [319](#)
- getFitLabels
 - dmmodel_fitFunc_plug.C, [173](#)
- getHighFreqCovar
 - planet_plug.C, [223](#)
 - spectralModel_plug.C, [241](#)
- getHighFreqRes
 - planet_plug.C, [223](#)

- spectralModel_plug.C, 241
- getInputs
 - getInputs.C, 123
 - tempo2.h, 319
- getInputs.C, 123
 - getInputs, 123
 - printplugs, 124
 - setPlugPath, 124
- getKeplerian
 - T2model.C, 288
- getLabel
 - matrix_plug.C, 219
- getLimits
 - GWwhiteLimit_plug.C, 214
- getMeteorologicalValue
 - tropo.C, 364
- getNglobal
 - doFit.C, 114
- getNparams
 - doFit.C, 114
- getObservatory
 - observatory.C, 143
 - tempo2.h, 320
- getParamDeriv
 - doFit.C, 114
 - tempo2.h, 320
- getParameter
 - glast_plug.C, 185
 - T2model.C, 288
- getParameterValue
 - tempo2.h, 320
 - tempo2Util.C, 332
- getPeriod.C, 124
 - main, 124
- getPostKeplerian
 - T2model.C, 288
- getPowerSpectra
 - interpolate_plug.C, 217
- getRedNoiseRealisation
 - simRedNoise_plug.C, 238
- getSensCurv
 - GWsens_plug.C, 210
- getSpectra
 - icLimit_plug.C, 215
- getSpectrum
 - detectGWB_plug.C, 169
 - detectGWBnew_plug.C, 170
- getStatPS
 - icLimit_plug.C, 215
- getSurfaceAtmosphericPressure
 - tropo.C, 364
- getTempoNestMaxLike
 - doFit.C, 114
- getThreshold
 - GWwhiteLimit_plug.C, 214
- getTspan
 - GWanisobkgrd_plug.C, 195
 - GWbkgrd_plug.C, 197
- GWbkgrdfromfile_plug.C, 198
- GWdipolebkgrd_plug.C, 201
- GWgeneralanisobkgrd_plug.C, 206
- GWgeneralbkgrd_plug.C, 207
- GWsingle_plug.C, 212
- icLimit_plug.C, 216
- getValue
 - readParfile.C, 256
- getZenithWetDelay
 - tropo.C, 364
- getprtj
 - sigmaz_plug.C, 235
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- getweights
 - sigmaz_plug.C, 235
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- glast_plug.C
 - FREE_ARG, 185
 - fitwave_function, 185
 - free_ivector, 185
 - getParameter, 185
 - graphicalInterface, 185
 - help, 185
 - indexx_patrick, 185
 - ivector, 185
 - M, 185
 - NR_END, 185
 - NRANSI, 185
 - NSTACK, 185
 - nrrerror, 185
 - SWAP, 185
 - sla_CALDJ, 185
 - sla_CLDJ, 185
 - slaCalyd, 185
 - slaClyd, 185
- glep
 - glitchS, 33
- glf0
 - glitchS, 33
- glf0d
 - glitchS, 33
- glf1
 - glitchS, 33
- glitch_plug.C
 - changeFit, 188
 - checkMenu, 188
 - defineGlitchVal, 188
 - doPlot, 188
 - drawMenu, 188
 - fitFuncs, 188
 - glitchS, 188
 - global_fitf0, 190
 - global_fitf1, 190
 - global_footer, 190
 - global_glitch, 190
 - global_header, 190

- global_nglt, 190
- global_valf0, 190
- global_valf1, 190
- graphicalInterface, 189
- help, 189
- interactivePlot, 189
- LM_DWARF, 188
- LM_MACHEP, 188
- LM_SQRT_DWARF, 188
- LM_SQRT_GIANT, 188
- LM_USERTOL, 188
- lm_control_double, 191
- lm_control_float, 191
- lm_enorm, 189
- lm_infmsg, 191
- lm_lmdif, 189
- lm_lmpar, 189
- lm_printout_std, 189
- lm_qrfac, 189
- lm_qrsolv, 189
- lm_shortmsg, 191
- lmcurve_evaluate, 189
- lmcurve_fit, 190
- lmin, 190
- MAX, 188
- MAX_TIMES, 188
- MIN, 188
- nonlinearFunc, 190
- plot1, 190
- plot2, 190
- plot3, 190
- plot4, 190
- plot5, 190
- plot6, 190
- plot7, 190
- plot8, 190
- plot9, 190
- plugVersionCheck, 191
- SQR, 188
- glitchS, 32
 - fitf0, 32
 - fitf0d, 32
 - fitf1, 32
 - fitph, 33
 - fittd, 33
 - glep, 33
 - glf0, 33
 - glf0d, 33
 - glf1, 33
 - glitch_plug.C, 188
 - glph, 33
 - gltd, 33
- global.C, 124
 - CVSdisplayVersion, 126
 - clock_corrections, 126
 - covarFuncFile, 126
 - dcmFile, 126
 - displayCVSversion, 126
 - ECLIPTIC_OBLIQUITY, 126
 - EXPSMOOTH, 126
 - ephemeris_routines, 126
 - extra_delays, 126
 - FCALPHA, 126
 - FCFINAL, 126
 - forceGlobalFit, 126
 - formBatsAll, 126
 - MAX_FUNCTIONS, 126
 - MAX_OBSN, 126
 - MAX_PSR, 126
 - NEWFIT, 126
 - NFIT, 127
 - TEMPO2_ENVIRON, 127
 - TEMPO2_ERROR, 127
 - tempo2_plug_path, 127
 - tempo2_plug_path_len, 127
 - tempo2MachineType, 127
 - UPW, 127
 - updateBatsAll, 126
 - veryFast, 127
 - WNLEVEL, 127
- global_fitFunc_plug.C
 - globalFITfuncs, 192
 - gnpsr, 192
 - plugVersionCheck, 192
 - pluginFitFunc, 192
- global_fitf0
 - glitch_plug.C, 190
- global_fitf1
 - glitch_plug.C, 190
- global_footer
 - glitch_plug.C, 190
- global_glitch
 - glitch_plug.C, 190
- global_header
 - glitch_plug.C, 190
- global_nglt
 - glitch_plug.C, 190
- global_valf0
 - glitch_plug.C, 190
- global_valf1
 - glitch_plug.C, 190
- globalDCM_fitFunc_plug.C
 - formCholeskyMatrix2, 193
 - globalFITfuncs, 193
 - gnpsr, 193
 - multMatrix2, 193
 - multMatrixVec2, 193
 - plugVersionCheck, 193
 - pluginFitFunc, 193
 - readUinv, 193
 - TKbacksubstitution_svd2, 193
 - TKbidiagonal2, 193
 - TKpythag2, 193
 - TKsingularValueDecomposition_lsqr2, 193
- globalFITfuncs
 - doFit.C, 114

- global_fitFunc_plug.C, [192](#)
 - globalDCM_fitFunc_plug.C, [193](#)
- globalNfit
 - pulsar, [62](#)
- globalNoConstrain
 - pulsar, [62](#)
- globalOmega
 - TKspectrum.C, [355](#)
- glph
 - glitchS, [33](#)
- gltd
 - glitchS, [33](#)
- gnpsr
 - global_fitFunc_plug.C, [192](#)
 - globalDCM_fitFunc_plug.C, [193](#)
- gotOut
 - calcDMe_plug.C, [158](#)
- gr
 - calcDMe_plug.C, [158](#)
- grTemplate_plug.C
 - graphicalInterface, [194](#)
 - help, [194](#)
 - plugVersionCheck, [194](#)
- GramSchmidt
 - GWwhiteLimit_plug.C, [214](#)
- graphicalInterface
 - addRed_plug.C, [145](#)
 - analyticChol_plug.C, [146](#)
 - angle_plug.C, [147](#)
 - applet_plug.C, [148](#)
 - autoDM_plug.C, [149](#)
 - autoSpectralFit_plug.C, [150](#)
 - averageData_plug.C, [151](#)
 - bary_plug.C, [153](#)
 - basic_plug.C, [154](#)
 - calcDMe_plug.C, [156](#)
 - checkWhite_plug.C, [160](#)
 - cholSpectra_plug.C, [161](#)
 - clock_plug.C, [162](#)
 - compareBackends_plug.C, [163](#)
 - compareDsets_plug.C, [165](#)
 - delays_plug.C, [166](#)
 - designmatrix_plug.C, [167](#)
 - detectGWB_plug.C, [169](#)
 - detectGWBnew_plug.C, [171](#)
 - dm_plug.C, [172](#)
 - efacEquad_plug.C, [175](#)
 - fake_plug.C, [177](#)
 - fermi_plug.C, [178](#)
 - findCW_plug.C, [179](#)
 - findCWs_plug.C, [180](#)
 - fixData_plug.C, [181](#)
 - GWanisobkgd_plug.C, [196](#)
 - GWbkgrd_plug.C, [197](#)
 - GWbkgrdfromfile_plug.C, [199](#)
 - GWdetect_plug.C, [200](#)
 - GWdipolebkgd_plug.C, [201](#)
 - GWevolve_plug.C, [204](#)
 - GWgeneralanisobkgd_plug.C, [206](#)
 - GWgeneralbkgd_plug.C, [207](#)
 - GWsens_plug.C, [210](#)
 - GWsingle_plug.C, [212](#)
 - GWwhiteLimit_plug.C, [214](#)
 - glast_plug.C, [185](#)
 - glitch_plug.C, [189](#)
 - grTemplate_plug.C, [194](#)
 - gwm_plug.C, [208](#)
 - gwmStats_plug.C, [209](#)
 - icLimit_plug.C, [216](#)
 - interpolate_plug.C, [217](#)
 - mjk_plug.C, [220](#)
 - photons_plug.C, [221](#)
 - planet_plug.C, [223](#)
 - plk_plug.C, [227](#)
 - plotMany_plug.C, [230](#)
 - sigmaz_plug.C, [235](#)
 - simRedNoise_plug.C, [238](#)
 - simulDM_plug.C, [239](#)
 - spectralModel_plug.C, [241](#)
 - spectrum_plug.C, [244](#)
 - splk_plug.C, [245](#)
 - transform_plug.C, [246](#)
- gser
 - efacEquad_plug.C, [175](#)
- gtol
 - lm_control_struct, [40](#)
- gwSrc, [35](#)
 - across_g, [36](#)
 - across_im_g, [36](#)
 - aplus_g, [36](#)
 - aplus_im_g, [36](#)
 - dist_bin, [36](#)
 - GWsim.h, [132](#)
 - h, [36](#)
 - h_im, [36](#)
 - inc_bin, [36](#)
 - kg, [36](#)
 - omega_g, [36](#)
 - phase_g, [36](#)
 - phi_bin, [36](#)
 - phi_g, [36](#)
 - phi_polar_g, [36](#)
 - theta_bin, [36](#)
 - theta_g, [36](#)
- gwamp
 - fixData_plug.C, [181](#)
- gwb_decj
 - pulsar, [62](#)
- gwb_epoch
 - pulsar, [63](#)
- gwb_geom_c
 - pulsar, [63](#)
- gwb_geom_p
 - pulsar, [63](#)
- gwb_raj
 - pulsar, [63](#)

- gwb_width
 - pulsar, [63](#)
- gwecc_dec
 - pulsar, [63](#)
- gwecc_distance
 - pulsar, [63](#)
- gwecc_e
 - pulsar, [63](#)
- gwecc_epoch
 - pulsar, [63](#)
- gwecc_inc
 - pulsar, [63](#)
- gwecc_m1
 - pulsar, [63](#)
- gwecc_m2
 - pulsar, [63](#)
- gwecc_nodes_orientation
 - pulsar, [63](#)
- gwecc_orbital_period
 - pulsar, [63](#)
- gwecc_psrdist
 - pulsar, [63](#)
- gwecc_pulsarTermOn
 - pulsar, [63](#)
- gwecc_ra
 - pulsar, [63](#)
- gwecc_redshift
 - pulsar, [63](#)
- gwecc_theta_0
 - pulsar, [63](#)
- gwecc_theta_nodes
 - pulsar, [63](#)
- gwgenSpec, [35](#)
 - GWsim.h, [132](#)
 - sl_alpha, [35](#)
 - sl_amp, [35](#)
 - st_alpha, [35](#)
 - st_amp, [35](#)
 - tensor_alpha, [35](#)
 - tensor_amp, [35](#)
 - vl_alpha, [35](#)
 - vl_amp, [35](#)
- gwgeneralSrc, [33](#)
 - across_g, [34](#)
 - across_im_g, [34](#)
 - apls_g, [34](#)
 - apls_im_g, [34](#)
 - asl_g, [34](#)
 - asl_im_g, [34](#)
 - ast_g, [34](#)
 - ast_im_g, [34](#)
 - avx_g, [34](#)
 - avx_im_g, [34](#)
 - avy_g, [34](#)
 - avy_im_g, [34](#)
 - dist_bin, [34](#)
 - GWsim.h, [132](#)
 - h, [34](#)
 - h_im, [34](#)
 - inc_bin, [34](#)
 - kg, [34](#)
 - omega_g, [34](#)
 - phase_g, [34](#)
 - phi_bin, [34](#)
 - phi_g, [34](#)
 - phi_polar_g, [34](#)
 - theta_bin, [34](#)
 - theta_g, [34](#)
- gwm_decj
 - pulsar, [63](#)
- gwm_dphase
 - pulsar, [63](#)
- gwm_epoch
 - pulsar, [63](#)
- gwm_phi
 - pulsar, [63](#)
- gwm_plug.C
 - graphicalInterface, [208](#)
 - help, [208](#)
 - plugVersionCheck, [208](#)
- gwm_raj
 - pulsar, [63](#)
- gwmStats_plug.C
 - calculateAngularFactors, [209](#)
 - calculateD, [209](#)
 - graphicalInterface, [209](#)
 - help, [209](#)
 - MAX_CORR, [209](#)
 - plugVersionCheck, [209](#)
- gwsim_Ngrid
 - GWsim.C, [130](#)
- gwsrc_across_i
 - pulsar, [63](#)
- gwsrc_across_i_e
 - pulsar, [63](#)
- gwsrc_across_r
 - pulsar, [63](#)
- gwsrc_across_r_e
 - pulsar, [64](#)
- gwsrc_apls_i
 - pulsar, [64](#)
- gwsrc_apls_i_e
 - pulsar, [64](#)
- gwsrc_apls_r
 - pulsar, [64](#)
- gwsrc_apls_r_e
 - pulsar, [64](#)
- gwsrc_dec
 - pulsar, [64](#)
- gwsrc_epoch
 - pulsar, [64](#)
- gwsrc_psrdist
 - pulsar, [64](#)
- gwsrc_ra
 - pulsar, [64](#)
- h

- gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- h_im
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- HAVE_BLAS
 - config.h, [100](#)
- HAVE_DLERROR
 - config.h, [100](#)
- HAVE_DLFCN_H
 - config.h, [100](#)
- HAVE_FFTW3
 - config.h, [100](#)
- HAVE_GWSIM_H
 - tempo2.h, [308](#)
- HAVE_INTTYPES_H
 - config.h, [100](#)
- HAVE_LAPACK
 - config.h, [100](#)
- HAVE_LIBDL
 - config.h, [100](#)
- HAVE_LIBDLLOADER
 - config.h, [100](#)
- HAVE_LIBM
 - config.h, [100](#)
- HAVE_MEMORY_H
 - config.h, [100](#)
- HAVE_PGPLOT
 - config.h, [100](#)
- HAVE_PTHREAD
 - config.h, [100](#)
- HAVE_STDINT_H
 - config.h, [100](#)
- HAVE_STDLIB_H
 - config.h, [100](#)
- HAVE_STRING_H
 - config.h, [100](#)
- HAVE_STRINGS_H
 - config.h, [100](#)
- HAVE_SYS_STAT_H
 - config.h, [100](#)
- HAVE_SYS_TYPES_H
 - config.h, [100](#)
- HAVE_UNISTD_H
 - config.h, [100](#)
- HTest
 - fermi_plug.C, [178](#)
- handleFreqPoints
 - calcDMe_plug.C, [156](#)
- hardcopy
 - calcDMe_plug.C, [158](#)
- hdfunc
 - detectGWB_plug.C, [169](#)
 - detectGWBnew_plug.C, [171](#)
- hdfunc_cosineSub
 - detectGWB_plug.C, [169](#)
- hdfunc_meanSub
 - detectGWB_plug.C, [169](#)
- hdfunc_offs
 - detectGWB_plug.C, [169](#)
- hdfunc_removeCosine
 - detectGWB_plug.C, [169](#)
- header
 - calcDMe_plug.C, [158](#)
- header_line
 - TabulatedFunction, [77](#)
- height_grs80
 - observatory, [50](#)
- help
 - addRed_plug.C, [145](#)
 - analyticChol_plug.C, [146](#)
 - angle_plug.C, [147](#)
 - applet_plug.C, [148](#)
 - autoDM_plug.C, [149](#)
 - autoSpectralFit_plug.C, [150](#)
 - averageData_plug.C, [151](#)
 - bary_plug.C, [153](#)
 - basic_plug.C, [154](#)
 - calcDMe_plug.C, [156](#)
 - checkWhite_plug.C, [160](#)
 - cholSpectra_plug.C, [161](#)
 - clock_plug.C, [162](#)
 - compareBackends_plug.C, [163](#)
 - compareDsets_plug.C, [165](#)
 - delays_plug.C, [166](#)
 - designmatrix_plug.C, [167](#)
 - detectGWB_plug.C, [169](#)
 - detectGWBnew_plug.C, [171](#)
 - dm_plug.C, [172](#)
 - efacEquad_plug.C, [175](#)
 - findCW_plug.C, [179](#)
 - findCWs_plug.C, [180](#)
 - fixData_plug.C, [181](#)
 - GWanisobkgrd_plug.C, [196](#)
 - GWbkgrd_plug.C, [197](#)
 - GWbkgrdfromfile_plug.C, [199](#)
 - GWdetect_plug.C, [200](#)
 - GWdipolebkgrd_plug.C, [201](#)
 - GWevolve_plug.C, [204](#)
 - GWgeneralanisobkgrd_plug.C, [206](#)
 - GWgeneralbkgrd_plug.C, [207](#)
 - GWsens_plug.C, [210](#)
 - GWsingle_plug.C, [212](#)
 - GWwhiteLimit_plug.C, [214](#)
 - glast_plug.C, [185](#)
 - glitch_plug.C, [189](#)
 - grTemplate_plug.C, [194](#)
 - gwm_plug.C, [208](#)
 - gwmStats_plug.C, [209](#)
 - icLimit_plug.C, [216](#)
 - interpolate_plug.C, [218](#)
 - mjk_plug.C, [220](#)
 - planet_plug.C, [223](#)
 - plk_plug.C, [227](#)
 - plotMany_plug.C, [230](#)
 - sigmaz_plug.C, [235](#)

- simRedNoise_plug.C, 238
- simulDM_plug.C, 239
- spectralModel_plug.C, 241
- spectrum_plug.C, 244
- transform_plug.C, 247
- hms_turn
 - tempo2.h, 320
 - tempo2Util.C, 332
 - tempo2Util.h, 333
- IAU_K
 - photons_plug.C, 221
- IAU_KINV
 - photons_plug.C, 221
- IAU_TEPH0
 - photons_plug.C, 221
- IF99_TIMEEPH
 - tempo2.h, 308
- IF_deltaT
 - tt2tdb.C, 366
- IFTE_DeltaT
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_DeltaTDot
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_JD0
 - ifteph.h, 136
- IFTE_K
 - ifteph.h, 136
- IFTE_KM1
 - ifteph.h, 136
- IFTE_LC
 - ifteph.h, 136
- IFTE_MJD0
 - ifteph.h, 136
- IFTE_TEPH0
 - ifteph.h, 136
- IFTE_close_file
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_get_DeltaT_DeltaTDot
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_get_Vals
 - ifteph.C, 134
- IFTE_get_vE
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_get_vE_vEDot
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_get_vEDot
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_init
 - ifteph.C, 134
 - ifteph.h, 136
- IFTE_interpolation_info, 36
- np, 37
- nv, 37
- pc, 37
- twot, 37
- vc, 37
- IFTEPH_FILE
 - tempo2.h, 308
- IFTEphemeris, 37
 - buf, 38
 - endJD, 38
 - ephver, 38
 - f, 38
 - iinfo, 38
 - ipt, 38
 - irec, 38
 - L_C, 38
 - reclen, 38
 - startJD, 38
 - stepJD, 38
 - swap_endian, 38
 - title, 38
- IFTswap4
 - ifteph.C, 134
- IFTswap8
 - ifteph.C, 134
- IFTswap8N
 - ifteph.C, 134
- IFTswapDouble
 - ifteph.C, 134
- IFTswapDoubles
 - ifteph.C, 134
- IFTswapInt
 - ifteph.C, 134
- IFTswapInts
 - ifteph.C, 134
- ITMAX
 - efacEquad_plug.C, 175
- ITRF_to_GRS80
 - bary_plug.C, 153
 - observatory.C, 143
- iau_c2t00b_
 - get_obsCoord.C, 122
- iau_cp_
 - get_obsCoord.C, 122
- iau_pom00_
 - get_obsCoord.C, 123
- iau_pxp_
 - get_obsCoord.C, 123
- iau_rxp_
 - get_obsCoord.C, 123
- iau_sxp_
 - get_obsCoord.C, 123
- iau_trxp_
 - get_obsCoord.C, 123
- iau_trxpv_
 - get_obsCoord.C, 123
- icLimit_plug.C
 - calculateGWCholesky, 215

- calculateStatistic, 215
- calculateWeighting, 215
- createGWcovarianceFunction, 215
- formCholeskyMatrixPlugin, 215
- getSpectra, 215
- getStatPS, 215
- getTspan, 216
- graphicalInterface, 216
- help, 216
- plugVersionCheck, 216
- id_residual
 - tempo2.h, 320
- idPoint
 - compareDsets_plug.C, 165
 - delays_plug.C, 166
 - plk_plug.C, 227
 - splk_plug.C, 246
- idPoint2
 - compareDsets_plug.C, 165
- identify
 - spectrum_plug.C, 244
- ifile
 - jpl_eph_data, 39
- ifteph.C, 133
 - IFTE_DeltaT, 134
 - IFTE_DeltaTDot, 134
 - IFTE_close_file, 134
 - IFTE_get_DeltaT_DeltaTDot, 134
 - IFTE_get_Vals, 134
 - IFTE_get_vE, 134
 - IFTE_get_vE_vEDot, 134
 - IFTE_get_vEDot, 134
 - IFTE_init, 134
 - IFTswap4, 134
 - IFTswap8, 134
 - IFTswap8N, 134
 - IFTswapDouble, 134
 - IFTswapDoubles, 134
 - IFTswapInt, 134
 - IFTswapInts, 134
- ifteph.h, 135
 - IFTE_DeltaT, 136
 - IFTE_DeltaTDot, 136
 - IFTE_JD0, 136
 - IFTE_K, 136
 - IFTE_KM1, 136
 - IFTE_LC, 136
 - IFTE_MJD0, 136
 - IFTE_TEPH0, 136
 - IFTE_close_file, 136
 - IFTE_get_DeltaT_DeltaTDot, 136
 - IFTE_get_vE, 136
 - IFTE_get_vE_vEDot, 136
 - IFTE_get_vEDot, 136
 - IFTE_init, 136
- ifunc
 - t2fit_ifunc.C, 279
 - t2fit_ifunc.h, 281
- ifunc_weights
 - pulsar, 64
- ifuncE
 - pulsar, 64
- ifuncN
 - pulsar, 64
- ifuncT
 - pulsar, 64
- ifuncV
 - pulsar, 64
- iinfo
 - IFTEphemeris, 38
 - jpl_eph_data, 39
- imag
 - complexVal, 30
- impCount
 - calcDMe_plug.C, 158
- impObs
 - calcDMe_plug.C, 158
- inc_bin
 - gwSrc, 36
 - gwgeneralSrc, 34
- indexx8
 - sigmaz_plug.C, 235
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- indexx_patrick
 - glast_plug.C, 185
- indx
 - sigmaz_plug.C, 235
 - TKspectrum.C, 355
- info
 - lm_status_struct, 41
- init
 - calcDMe_plug.C, 156
- init_genrand
 - T2toolkit.C, 290
 - T2toolkit.h, 292
- init_ifte
 - tt2tdb.C, 366
- initObservatories
 - observatory.C, 143
- initialise
 - initialise.C, 137
 - tempo2.h, 320
- initialise.C, 136
 - allocateMemory, 137
 - destroyMemory, 137
 - destroyOne, 137
 - initialise, 137
 - initialiseOne, 137
- initialiseOne
 - initialise.C, 137
 - tempo2.h, 320
- initialize_ClockCorrections
 - clkcorr.C, 98
- initialize_meteorology_table
 - tropo.C, 364

initialize_meteorology_tables
 tropo.C, 364
 inner_product
 fermi_plug.C, 178
 interactivePlot
 glitch_plug.C, 189
 interpolate_plug.C
 choldc, 217
 FREE_ARG, 217
 free_vector, 217
 getPowerSpectra, 217
 graphicalInterface, 217
 help, 218
 lubksb, 218
 ludcmp, 218
 MAX_SAMPLES, 217
 matrixMult, 218
 NR_END, 217
 NRANSI, 217
 nrerror, 218
 plotModel, 218
 plotResiduals, 218
 plugVersionCheck, 218
 sample, 217
 sortSamples, 218
 TINY, 217
 vector, 218
 interpolateSplineSmooth
 calcDMe_plug.C, 156
 interpolateWeightedSmooth
 calcDMe_plug.C, 156
 interpolation_info, 38
 np, 38
 nv, 38
 pc, 38
 twot, 38
 vc, 38
 ipm
 pulsar, 64
 ipt
 IFTEphemeris, 38
 jpl_eph_data, 39
 irec
 IFTEphemeris, 38
 iterativeFit
 mjk_plug.C, 220
 ivector
 efacEquad_plug.C, 176
 glast_plug.C, 185

 JPL_EPHEM_AU_IN_KM
 jpleph.h, 140
 JPL_EPHEM_EARTH_MOON_RATIO
 jpleph.h, 140
 JPL_EPHEM_END_JD
 jpleph.h, 140
 JPL_EPHEM_EPHEMERIS_VERSION
 jpleph.h, 141
 JPL_EPHEM_KERNEL_NCOEFF
 jpleph.h, 141
 JPL_EPHEM_KERNEL_RECORD_SIZE
 jpleph.h, 141
 JPL_EPHEM_KERNEL_SIZE
 jpleph.h, 141
 JPL_EPHEM_KERNEL_SWAP_BYTES
 jpleph.h, 141
 JPL_EPHEM_N_CONSTANTS
 jpleph.h, 141
 JPL_EPHEM_START_JD
 jpleph.h, 141
 JPL_EPHEM_STEP
 jpleph.h, 141
 JPL_EPHEMERIS
 pulsar, 64
 JPL_HEADER_SIZE
 jpl_int.h, 138
 JPLlong
 jpl_int.h, 138
 JVmodel
 tempo2.h, 320
 jboFormat
 pulsar, 64
 jpl_close_ephemeris
 jpleph.c, 139
 jpleph.h, 141
 jpl_eph_data, 39
 au, 39
 cache, 39
 curr_cache_loc, 39
 emrat, 39
 ephem_end, 39
 ephem_start, 39
 ephem_step, 39
 ephemeris_version, 39
 ifile, 39
 iinfo, 39
 ipt, 39
 kernel_size, 40
 ncoeff, 40
 ncon, 40
 pvsun, 40
 resize, 40
 swap_bytes, 40
 jpl_get_double
 jpleph.c, 139
 jpleph.h, 141
 jpl_get_long
 jpleph.c, 139
 jpleph.h, 141
 jpl_init_ephemeris
 jpleph.c, 139
 jpleph.h, 141
 jpl_int.h, 138
 JPL_HEADER_SIZE, 138
 JPLlong, 138
 MAX_KERNEL_SIZE, 138
 jpl_pleph

- jpleph.c, 139
- jpleph.h, 141
- jpl_state
 - jpleph.c, 139
 - jpleph.h, 141
- jpleph.c, 138
 - FALSE, 139
 - jpl_close_ephemeris, 139
 - jpl_get_double, 139
 - jpl_get_long, 139
 - jpl_init_ephemeris, 139
 - jpl_pleph, 139
 - jpl_state, 139
 - SWAP_MACRO, 139
 - TRUE, 139
- jpleph.h, 140
 - DLL_FUNC, 140
 - JPL_EPHEM_AU_IN_KM, 140
 - JPL_EPHEM_EARTH_MOON_RATIO, 140
 - JPL_EPHEM_END_JD, 140
 - JPL_EPHEM_EPHEMERIS_VERSION, 141
 - JPL_EPHEM_KERNEL_NCOEFF, 141
 - JPL_EPHEM_KERNEL_RECORD_SIZE, 141
 - JPL_EPHEM_KERNEL_SIZE, 141
 - JPL_EPHEM_KERNEL_SWAP_BYTES, 141
 - JPL_EPHEM_N_CONSTANTS, 141
 - JPL_EPHEM_START_JD, 141
 - JPL_EPHEM_STEP, 141
 - jpl_close_ephemeris, 141
 - jpl_get_double, 141
 - jpl_get_long, 141
 - jpl_init_ephemeris, 141
 - jpl_pleph, 141
 - jpl_state, 141
 - make_sub_ephem, 141
- jump
 - observation, 46
- jumpStr
 - pulsar, 64
- jumpVal
 - pulsar, 64
- jumpValErr
 - pulsar, 64
- jupiter_earth
 - observation, 46
- kernel_size
 - jpl_eph_data, 40
- kg
 - gwSrc, 36
 - gwgeneralSrc, 34
- kind
 - T2Predictor, 76
- kmax
 - GWevolve_plug.C, 204
- KopeikinTerms
 - T2model.C, 288
- kount
 - GWevolve_plug.C, 204
- ksone
 - efacEquad_plug.C, 176
- kstwo
 - efacEquad_plug.C, 176
- L_C
 - IFTEphemeris, 38
- LD_PI
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 347
 - TKlongdouble.ld.h, 348
- LEAPSECOND_FILE
 - tempo2.h, 308
- LINE_LENGTH
 - cholesky.C, 89
- LM_DWARF
 - glitch_plug.C, 188
- LM_MACHEP
 - glitch_plug.C, 188
- LM_SQRT_DWARF
 - glitch_plug.C, 188
- LM_SQRT_GIANT
 - glitch_plug.C, 188
- LM_USERTOL
 - glitch_plug.C, 188
- LOG_OUTFILE
 - TKlog.h, 342
- LONGDOUBLE_IS_FLOAT128
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 347
- LONGDOUBLE_IS_IEEE754
 - TKlongdouble.ld.h, 349
- LONGDOUBLE_ONE
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 347
 - TKlongdouble.ld.h, 349
- LT_OBJDIR
 - config.h, 100
- label
 - parameter, 52
 - tempo2.h, 315
- lastUsedSession
 - calcDMe_plug.C, 158
- latitude_grs80
 - observatory, 51
- ld_fprintf
 - TKlongdouble.C, 344
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 347
 - TKlongdouble.ld.h, 348
- ld_printf
 - TKlongdouble.C, 344
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 347
 - TKlongdouble.ld.h, 348
- ld_sprintf
 - TKlongdouble.C, 344
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 348

- TKlongdouble.Id.h, 348
- ld_vsprintf
 - TKlongdouble.C, 344
- libt2toolkit API, 23
- libtempo2 External API, 24
- linearInterpolate
 - toa2utc.C, 363
- linfile
 - sigmaz_plug.C, 235
 - TKspectrum.C, 355
- linkFrom
 - parameter, 52
- linkTo
 - parameter, 53
- lm_control_double
 - glitch_plug.C, 191
- lm_control_float
 - glitch_plug.C, 191
- lm_control_struct, 40
 - epsilon, 40
 - ftol, 40
 - gtol, 40
 - maxcall, 40
 - printflags, 40
 - scale_diag, 40
 - stepbound, 40
 - xtol, 40
- lm_enorm
 - glitch_plug.C, 189
- lm_infmsg
 - glitch_plug.C, 191
- lm_lmdif
 - glitch_plug.C, 189
- lm_lmpar
 - glitch_plug.C, 189
- lm_printout_std
 - glitch_plug.C, 189
- lm_qrfac
 - glitch_plug.C, 189
- lm_qrsolv
 - glitch_plug.C, 189
- lm_shortmsg
 - glitch_plug.C, 191
- lm_status_struct, 41
 - fnorm, 41
 - info, 41
 - nfev, 41
- lmcurve_data_struct, 41
 - f, 41
 - t, 41
 - y, 41
- lmcurve_evaluate
 - glitch_plug.C, 189
- lmcurve_fit
 - glitch_plug.C, 190
- lmin
 - glitch_plug.C, 190
- lmst
 - get_obsCoord.C, 123
- lmst2
 - plk_plug.C, 227
 - plotMany_plug.C, 230
- load_EOP
 - eop.C, 119
- log10rms
 - T1Polyco, 74
- logdbg
 - TKlog.h, 342
- logerr
 - TKlog.h, 342
- logerr_check
 - TKlog.C, 340
 - TKlog.h, 343
- logicFlag
 - preProcess.C, 248
 - tempo2.h, 320
- logmsg
 - TKlog.h, 342
- logtchk
 - TKlog.h, 342
- logwarn
 - TKlog.h, 342
- lombScargle
 - checkWhite_plug.C, 160
- longdouble
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 347
 - TKlongdouble.Id.h, 349
- longitude_grs80
 - observatory, 51
- lookup_observatory_alias
 - observatory.C, 143
 - tempo2.h, 320
- lubksb
 - interpolate_plug.C, 218
- ludcmp
 - interpolate_plug.C, 218
- M
 - efacEquad_plug.C, 175
 - glast_plug.C, 185
- m2
 - textOutput.C, 334
- M_PII
 - cheby2d.c, 86
- MASYR2RADS
 - tempo2.h, 308
- MAX
 - glitch_plug.C, 188
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- MAX_BPJ_JUMPS
 - tempo2.h, 308
- MAX_CLK_CORR
 - tempo2.h, 308
- MAX_CLKCORR
 - tempo2.h, 308

MAX_COEFF
 tempo2.h, 308
 MAX_COMPANIONS
 tempo2.h, 308
 MAX_CORR
 gwmStats_plug.C, 209
 MAX_CURRENT
 sw_delay.C, 262
 MAX_DM
 simulDM_plug.C, 239
 MAX_DM_DERIVATIVES
 tempo2.h, 308
 MAX_DMX
 tempo2.h, 308
 MAX_FILELEN
 tempo2.h, 309
 MAX_FIT
 tempo2.h, 309
 MAX_FLAG
 GWwhiteLimit_plug.C, 213
 MAX_FLAG_LEN
 tempo2.h, 309
 MAX_FLAGS
 tempo2.h, 309
 MAX_FREQ
 autoSpectralFit_plug.C, 150
 GWwhiteLimit_plug.C, 213
 MAX_FREQ_DERIVATIVES
 tempo2.h, 309
 MAX_FUNCTIONS
 global.C, 126
 MAX_GWS
 sigmaz_plug.C, 234
 MAX_HIGHLIGHT
 delays_plug.C, 166
 MAX_ID
 spectrum_plug.C, 244
 MAX_IFUNC
 tempo2.h, 309
 MAX_ITER
 bootstrap.C, 80
 MAX_ITERATION
 GWwhiteLimit_plug.C, 213
 MAX_JUMPS
 tempo2.h, 309
 MAX_KERNEL_SIZE
 jpl_int.h, 138
 MAX_LEAPSEC
 tempo2.h, 309
 MAX_MSG
 tempo2.h, 309
 MAX_OBSN
 global.C, 126
 t2toolkit_global.C, 294
 tempo2.h, 323
 MAX_OBSN_VAL
 tempo2.h, 309
 MAX_PARAMS
 tempo2.h, 309
 MAX_POLY
 checkWhite_plug.C, 160
 GWwhiteLimit_plug.C, 213
 MAX_PSR
 global.C, 126
 tempo2.h, 323
 MAX_PSR_VAL
 tempo2.h, 309
 MAX_QUAD
 tempo2.h, 310
 MAX_SAMPLES
 interpolate_plug.C, 217
 MAX_SHOTS
 readEphemeris.C, 253
 MAX_SITE
 tempo2.h, 310
 MAX_STOREPRECISION
 tempo2.h, 310
 MAX_STRLEN
 tempo2.h, 310
 MAX_T2EFAC
 tempo2.h, 310
 MAX_T2EQUAD
 tempo2.h, 310
 MAX_TEL_CLK_OFFS
 tempo2.h, 310
 MAX_TEL_DX
 tempo2.h, 310
 MAX_TEL_DY
 tempo2.h, 310
 MAX_TEL_DZ
 tempo2.h, 310
 MAX_TIMES
 averageData_plug.C, 151
 dm_plug.C, 172
 glitch_plug.C, 188
 MAX_TNBN
 tempo2.h, 310
 MAX_TNDMEv
 tempo2.h, 310
 MAX_TNECORR
 tempo2.h, 310
 MAX_TNEF
 tempo2.h, 311
 MAX_TNEQ
 tempo2.h, 311
 MAX_TNGN
 tempo2.h, 311
 MAX_TNSQ
 tempo2.h, 311
 MAX_TOFFSET
 tempo2.h, 311
 MAX_VAL
 GWevolve_plug.C, 203
 MAX_WHITE
 tempo2.h, 311
 MAXSTP

- GWevolve_plug.C, 203
- MIN
 - glitch_plug.C, 188
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- MSSmodel
 - MSSmodel.C, 142
 - tempo2.h, 320
- MSSmodel.C, 141
 - MSSmodel, 142
 - updateMSS, 142
- main
 - getPeriod.C, 124
 - tempo2.C, 299
- make_sub_ephem
 - jpleph.h, 141
- makeClockCorrectionSequence
 - clkcorr.C, 98
- makechars
 - readJBO_bat.C, 255
- malloc_2dLL
 - TKmatrix.C, 350
- malloc_2df
 - TKmatrix.C, 350
 - TKmatrix.h, 351
- malloc_blas
 - TKmatrix.C, 350
 - TKmatrix.h, 351
- malloc_uinv
 - TKmatrix.C, 350
 - TKmatrix.h, 351
- mass2dd
 - DDGRmodel.C, 105
- mat20
 - sigmaz_plug.C, 235
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- matrix_plug.C
 - getLabel, 219
 - plugVersionCheck, 219
 - tempoOutput, 219
- matrixDMConstraintWeights
 - constraints.C, 103
- matrixMult
 - GWsim.C, 130
 - GWsim.h, 132
 - interpolate_plug.C, 218
- maxcall
 - lm_control_struct, 40
- mcl2
 - sw_delay.C, 262
- mean
 - calcDMe_plug.C, 158
- meanMJD
 - calcDMe_plug.C, 158
- meanMJDval
 - calcDMe_plug.C, 158
- meanVal
 - calcDMe_plug.C, 158
- met2mjd
 - fermi_plug.C, 178
- MeteorologyFunction, 42
 - siteName, 42
 - table, 42
- MeteorologyFunction_getEndMJD
 - tropo.C, 365
- MeteorologyFunction_getStartMJD
 - tropo.C, 365
- MeteorologyFunction_getValue
 - tropo.C, 365
- MeteorologyFunction_load
 - tropo.C, 365
- minPrec
 - storePrecision, 73
- mjd
 - EOPSample, 31
- mjd2date
 - sw_delay.C, 263
- mjd2met
 - fermi_plug.C, 178
- mjd2year
 - clock_plug.C, 162
 - dm_plug.C, 172
- mjd_end
 - ChebyModel, 26
- mjd_mid
 - T1Polyco, 74
- mjd_start
 - ChebyModel, 26
- mjk_plug.C
 - _itt, 220
 - graphicalInterface, 220
 - help, 220
 - iterativeFit, 220
 - NIT, 220
 - saveparams, 220
- model
 - ChebyModelInfo, 27
 - spectrum_plug.C, 244
- modelfcn
 - planet_plug.C, 223
- modelset
 - T2Predictor, 76
- multMatrix2
 - globalDCM_fitFunc_plug.C, 193
- multMatrixVec2
 - globalDCM_fitFunc_plug.C, 193
- nCompanion
 - pulsar, 65
- nConstraints
 - FitInfo, 32
- nDMEvents
 - pulsar, 65
- NE_SW_DEFAULT
 - tempo2.h, 311
- NEWFIT

- global.C, 126
- tempo2.h, 323
- NFIT
 - choleskyRoutines.h, 97
 - global.C, 127
- nFit
 - pulsar, 65
- nFlags
 - observation, 46
- NGWmax
 - GWbkgrdfromfile_plug.C, 199
- nGlobal
 - pulsar, 65
- NIT
 - mjk_plug.C, 220
- nJumps
 - pulsar, 65
- nLinkFrom
 - parameter, 53
- nLinkTo
 - parameter, 53
- NMF_hydrostatic
 - tropo.C, 365
- NMF_wet
 - tropo.C, 365
- nParam
 - pulsar, 65
- nParams
 - FitInfo, 32
- nPhaseJump
 - pulsar, 65
- nQuad
 - pulsar, 66
- NR_END
 - efacEquad_plug.C, 175
 - GWevolve_plug.C, 203
 - glast_plug.C, 185
 - interpolate_plug.C, 217
- NRANSI
 - glast_plug.C, 185
 - interpolate_plug.C, 217
- NSTACK
 - efacEquad_plug.C, 175
 - glast_plug.C, 185
- nSessions
 - calcDMe_plug.C, 158
- nStorePrecision
 - pulsar, 66
- nT2efac
 - pulsar, 66
- nT2equad
 - pulsar, 66
- nTNBandNoise
 - pulsar, 66
- nTNECORR
 - pulsar, 66
- nTNEF
 - pulsar, 66
- nTNEQ
 - pulsar, 66
- nTNGroupNoise
 - pulsar, 66
- nTNSQ
 - pulsar, 66
- nTNShapeletEvents
 - pulsar, 66
- nTelDX
 - pulsar, 66
- nTelDY
 - pulsar, 66
- nTelDZ
 - pulsar, 66
- nToffset
 - pulsar, 66
- nWhite
 - pulsar, 66
- nWhite_dm
 - pulsar, 66
- nallocated
 - DynamicArray, 30
- name
 - observatory, 51
 - pulsar, 64
- nbintype
 - sigmaz_plug.C, 236
 - TKspectrum.C, 355
- nclock_correction
 - observation, 46
- ncoeff
 - jpl_eph_data, 40
 - T1Polycy, 74
- ncon
 - jpl_eph_data, 40
- nconstraints
 - pulsar, 65
- ncubic
 - sigmaz_plug.C, 236
 - TKspectrum.C, 355
- ncubics
 - sigmaz_plug.C, 236
 - TKspectrum.C, 355
- ndim
 - sigmaz_plug.C, 236
 - TKspectrum.C, 355
- ndmx
 - pulsar, 65
- ne_sw
 - pulsar, 65
- nelem
 - DynamicArray, 30
- neptune_earth
 - observation, 46
- newTim
 - plk_plug.C, 227
- nf
 - calcDMe_plug.C, 158

- nfev
 - lm_status_struct, 41
- nformat
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- nint_derived
 - applet_plug.C, 148
 - general2_plug.C, 182
 - general_plug.C, 184
 - publish_plug.C, 232
- nit
 - fixData_plug.C, 181
- nits
 - pulsar, 65
- noWarnings
 - pulsar, 65
- nobs
 - pulsar, 65
- NonePredType
 - tempo2pred.h, 327
- nonlinearFunc
 - glitch_plug.C, 190
- notim
 - detectGWBnew_plug.C, 171
- np
 - IFTE_interpolation_info, 37
 - interpolation_info, 38
- nphase
 - observation, 47
- npt
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- npt1last
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- npt2last
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- nrerror
 - efacEquad_plug.C, 176
 - GWevolve_plug.C, 204
 - glast_plug.C, 185
 - interpolate_plug.C, 218
- nsegments
 - ChebyModelSet, 28
 - T1PolycoSet, 75
- ntau
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- ntunits
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- nusewt
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- nutations
 - observation, 47
- nv
 - IFTE_interpolation_info, 37
 - interpolation_info, 38
- nwriteres
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- nx
 - Cheby2D, 25
- nxunits
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- ny
 - Cheby2D, 25
- OBLQ
 - tempo2.h, 311
- OBSSYS_FILE
 - tempo2.h, 311
- OMEGA0
 - cholSpectra_plug.C, 161
 - detectGWB_plug.C, 169
 - detectGWBnew_plug.C, 171
- obsNjump
 - observation, 47
- observation, 42
 - addedNoise, 44
 - averagebat, 44
 - averageerr, 44
 - averageres, 44
 - bat, 45
 - batCorr, 45
 - bbat, 45
 - clockCorr, 45
 - correctionTT_TB, 45
 - correctionTT_Teph, 45
 - correctionUT1, 45
 - correctionsTT, 45
 - delayCorr, 45
 - deleted, 45
 - earth_ssb, 45
 - earthMoonBary_earth, 45
 - earthMoonBary_ssb, 45
 - efac, 46
 - einsteinRate, 46
 - equad, 46
 - flagID, 46
 - flagVal, 46
 - fname, 46
 - freq, 46
 - freqSSB, 46
 - jump, 46
 - jupiter_earth, 46
 - nFlags, 46
 - nclock_correction, 46
 - neptune_earth, 46
 - nphase, 47
 - nutations, 47
 - obsNjump, 47
 - observatory_earth, 47
 - origErr, 47

- origsat, [47](#)
- pet, [47](#)
- phase, [47](#)
- phaseOffset, [47](#)
- planet_ssb, [47](#)
- prefitResidual, [47](#)
- psrPos, [47](#)
- pulseN, [47](#)
- residual, [47](#)
- roemer, [47](#)
- sat, [48](#)
- sat_day, [48](#)
- sat_sec, [48](#)
- saturn_earth, [48](#)
- shapiroDelayJupiter, [48](#)
- shapiroDelayNeptune, [48](#)
- shapiroDelaySaturn, [48](#)
- shapiroDelaySun, [48](#)
- shapiroDelayUranus, [48](#)
- shapiroDelayVenus, [48](#)
- shklovskii, [48](#)
- siteVel, [48](#)
- sun_earth, [48](#)
- sun_ssb, [49](#)
- TNDMErr, [49](#)
- TNDMSignal, [49](#)
- TNGroupErr, [49](#)
- TNGroupSignal, [49](#)
- TNRedErr, [49](#)
- TNRedSignal, [49](#)
- tdis1, [49](#)
- tdis2, [49](#)
- tellID, [49](#)
- tempo2.h, [313](#)
- toaDMErr, [49](#)
- toaErr, [49](#)
- torb, [50](#)
- troposphericDelay, [50](#)
- uranus_earth, [50](#)
- venus_earth, [50](#)
- zenith, [50](#)
- observatory, [50](#)
 - clock_name, [50](#)
 - code, [50](#)
 - height_grs80, [50](#)
 - latitude_grs80, [51](#)
 - longitude_grs80, [51](#)
 - name, [51](#)
 - x, [51](#)
 - y, [51](#)
 - z, [51](#)
- observatory.C, [142](#)
 - fang, [143](#)
 - GRS80_A, [143](#)
 - GRS80_F, [143](#)
 - GRS80_to_ITRF, [143](#)
 - getObservatory, [143](#)
 - ITRF_to_GRS80, [143](#)
 - initObservatories, [143](#)
 - lookup_observatory_alias, [143](#)
 - readAliases, [144](#)
 - readObservatoryFile, [144](#)
- observatory_earth
 - observation, [47](#)
- ObservatoryAliasList, [51](#)
 - aliases, [51](#)
 - code, [51](#)
- obsn
 - pulsar, [66](#)
- obtainTimingResiduals
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [241](#)
- ode
 - GWevolve_plug.C, [204](#)
- offset
 - pulsar, [66](#)
- offset_e
 - pulsar, [66](#)
- offsetToCM
 - detectGWBnew_plug.C, [171](#)
- omega_g
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- open_file
 - read_fortran.h, [251](#)
- open_file2
 - read_fortran2.h, [252](#)
- origErr
 - observation, [47](#)
- origsat
 - observation, [47](#)
- othpl
 - doFit.C, [114](#)
- outDM
 - calcDMe_plug.C, [158](#)
- outFileName
 - calcDMe_plug.C, [158](#)
- outInterpCount
 - calcDMe_plug.C, [158](#)
- outSmoothCount
 - calcDMe_plug.C, [158](#)
- outX
 - calcDMe_plug.C, [158](#)
- outY
 - calcDMe_plug.C, [158](#)
- outer_product
 - fermi_plug.C, [178](#)
- output
 - calcDMe_plug.C, [156](#)
- outputCovarianceFunction
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [241](#)
- outputMatrix
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- outputResults

- sw_delay.C, [263](#)
- outputTMatrix
 - pulsar, [66](#)
- overPlotN
 - plk_plug.C, [227](#)
- overPlotShapiro
 - plk_plug.C, [227](#)
- PACKAGE
 - config.h, [100](#)
- PACKAGE_BUGREPORT
 - config.h, [100](#)
- PACKAGE_NAME
 - config.h, [101](#)
- PACKAGE_STRING
 - config.h, [101](#)
- PACKAGE_TARNAME
 - config.h, [101](#)
- PACKAGE_URL
 - config.h, [101](#)
- PACKAGE_VERSION
 - config.h, [101](#)
- PCM
 - GWevolve_plug.C, [203](#)
 - tempo2.h, [311](#)
- PGROW
 - GWevolve_plug.C, [203](#)
- PSHRNK
 - GWevolve_plug.C, [203](#)
- parFile
 - calcDMe_plug.C, [158](#)
- param
 - pulsar, [66](#)
- param_JUMP
 - tempo2.h, [318](#)
- param_LAST
 - tempo2.h, [318](#)
- param_ZERO
 - tempo2.h, [318](#)
- param_a0
 - tempo2.h, [316](#)
- param_a1
 - tempo2.h, [315](#)
- param_a1dot
 - tempo2.h, [316](#)
- param_a2dot
 - tempo2.h, [316](#)
- param_afac
 - tempo2.h, [317](#)
- param_b0
 - tempo2.h, [316](#)
- param_bp
 - tempo2.h, [316](#)
- param_bpja1
 - tempo2.h, [316](#)
- param_bpjec
 - tempo2.h, [316](#)
- param_bpjep
 - tempo2.h, [316](#)
- param_bpjom
 - tempo2.h, [316](#)
- param_bpjpb
 - tempo2.h, [316](#)
- param_bpjph
 - tempo2.h, [316](#)
- param_bpp
 - tempo2.h, [316](#)
- param_brake
 - tempo2.h, [318](#)
- param_cgw
 - tempo2.h, [317](#)
- param_clk_offs
 - tempo2.h, [317](#)
- param_daop
 - tempo2.h, [317](#)
- param_decj
 - tempo2.h, [315](#)
- param_df1
 - tempo2.h, [318](#)
- param_dm
 - tempo2.h, [315](#)
- param_dm_cos1yr
 - tempo2.h, [317](#)
- param_dm_sin1yr
 - tempo2.h, [317](#)
- param_dmassplanet
 - tempo2.h, [317](#)
- param_dmepoch
 - tempo2.h, [315](#)
- param_dmmodel
 - tempo2.h, [317](#)
- param_dmx
 - tempo2.h, [317](#)
- param_dmxr1
 - tempo2.h, [317](#)
- param_dmxr2
 - tempo2.h, [317](#)
- param_dr
 - tempo2.h, [316](#)
- param_dshk
 - tempo2.h, [317](#)
- param_dth
 - tempo2.h, [316](#)
- param_dtheta
 - tempo2.h, [316](#)
- param_e2dot
 - tempo2.h, [315](#)
- param_ecc
 - tempo2.h, [315](#)
- param_edot
 - tempo2.h, [315](#)
- param_ephver
 - tempo2.h, [317](#)
- param_eps1
 - tempo2.h, [316](#)
- param_eps1dot
 - tempo2.h, [317](#)

param_eps2
 tempo2.h, 316
param_eps2dot
 tempo2.h, 317
param_f
 tempo2.h, 315
param_fb
 tempo2.h, 315
param_fd
 tempo2.h, 316
param_fddc
 tempo2.h, 316
param_fddi
 tempo2.h, 316
param_finish
 tempo2.h, 316
param_gamma
 tempo2.h, 316
param_glep
 tempo2.h, 316
param_glf0
 tempo2.h, 316
param_glf0d
 tempo2.h, 316
param_glf1
 tempo2.h, 316
param_glf2
 tempo2.h, 316
param_glph
 tempo2.h, 316
param_gltid
 tempo2.h, 316
param_gwb_amp
 tempo2.h, 317
param_gwecc
 tempo2.h, 317
param_gwm_amp
 tempo2.h, 317
param_gwsingle
 tempo2.h, 317
param_h3
 tempo2.h, 317
param_h4
 tempo2.h, 317
param_ifunc
 tempo2.h, 317
param_iperharm
 tempo2.h, 317
param_kin
 tempo2.h, 316
param_kom
 tempo2.h, 316
param_label
 tempo2.h, 313
param_m2
 tempo2.h, 316
param_mtot
 tempo2.h, 316
param_nharm
 tempo2.h, 317
param_om
 tempo2.h, 315
param_om2dot
 tempo2.h, 316
param_omdot
 tempo2.h, 316
param_orbpx
 tempo2.h, 316
param_pb
 tempo2.h, 315
param_pbdot
 tempo2.h, 315
param_pepoch
 tempo2.h, 315
param_pmdec
 tempo2.h, 315
param_pmra
 tempo2.h, 315
param_pmrsv
 tempo2.h, 315
param_posepoch
 tempo2.h, 315
param_px
 tempo2.h, 315
param_quad_ifunc_c
 tempo2.h, 317
param_quad_ifunc_p
 tempo2.h, 317
param_quad_om
 tempo2.h, 317
param_raj
 tempo2.h, 315
param_shapmax
 tempo2.h, 316
param_sini
 tempo2.h, 315
param_start
 tempo2.h, 316
param_stateSwitchT
 tempo2.h, 318
param_stig
 tempo2.h, 317
param_t0
 tempo2.h, 315
param_tasc
 tempo2.h, 316
param_tel_dx
 tempo2.h, 317
param_tel_dy
 tempo2.h, 317
param_tel_dz
 tempo2.h, 317
param_tel_vx
 tempo2.h, 317
param_tel_vy
 tempo2.h, 317

- param_tel_vz
 - tempo2.h, [317](#)
- param_tel_x0
 - tempo2.h, [317](#)
- param_tel_y0
 - tempo2.h, [317](#)
- param_tel_z0
 - tempo2.h, [317](#)
- param_telEpoch
 - tempo2.h, [317](#)
- param_telx
 - tempo2.h, [317](#)
- param_tely
 - tempo2.h, [317](#)
- param_telz
 - tempo2.h, [317](#)
- param_track
 - tempo2.h, [316](#)
- param_tres
 - tempo2.h, [317](#)
- param_tspan
 - tempo2.h, [316](#)
- param_tzfrq
 - tempo2.h, [316](#)
- param_tzrmjd
 - tempo2.h, [316](#)
- param_wave_dm
 - tempo2.h, [317](#)
- param_wave_om
 - tempo2.h, [316](#)
- param_waveepoch
 - tempo2.h, [317](#)
- param_waveepoch_dm
 - tempo2.h, [317](#)
- param_xomdot
 - tempo2.h, [316](#)
- param_xpbdot
 - tempo2.h, [315](#)
- paramCounters
 - FitInfo, [32](#)
- paramDerivFunc
 - tempo2.h, [313](#)
- paramDerivs
 - FitInfo, [32](#)
- paramIndex
 - FitInfo, [32](#)
- paramSet
 - parameter, [53](#)
- paramUpdateFunc
 - tempo2.h, [313](#)
- parameter, [52](#)
 - aSize, [52](#)
 - err, [52](#)
 - fitFlag, [52](#)
 - label, [52](#)
 - linkFrom, [52](#)
 - linkTo, [53](#)
 - nLinkFrom, [53](#)
 - nLinkTo, [53](#)
 - paramSet, [53](#)
 - prefit, [53](#)
 - prefitErr, [53](#)
 - shortlabel, [53](#)
 - tempo2.h, [313](#)
 - val, [53](#)
- parse_longdouble
 - TKlongdouble.C, [344](#)
 - TKlongdouble.float128.h, [346](#)
 - TKlongdouble.h, [348](#)
 - TKlongdouble.Id.h, [349](#)
- parseExp
 - publish_plug.C, [232](#)
- parseLine
 - applet_plug.C, [148](#)
 - general2_plug.C, [183](#)
 - general_plug.C, [184](#)
- parseMinus
 - publish_plug.C, [232](#)
- passStr
 - pulsar, [66](#)
- pc
 - IFTE_interpolation_info, [37](#)
 - interpolation_info, [38](#)
- pcshft
 - polyco.C, [247](#)
- permax
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- pet
 - observation, [47](#)
- pgdevice
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- phase
 - observation, [47](#)
- phase_g
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- phaseJump
 - pulsar, [67](#)
- phaseJumpDir
 - pulsar, [67](#)
- phaseJumpID
 - pulsar, [67](#)
- phaseOffset
 - observation, [47](#)
- phi_bin
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- phi_g
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- phi_polar_g
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- photons_plug.C

- check_barycentered, [221](#)
- find_event_hdu, [221](#)
- get_mjdref, [221](#)
- graphicalInterface, [221](#)
- IAU_K, [221](#)
- IAU_KINV, [221](#)
- IAU_TEPH0, [221](#)
- tcb2tdb, [221](#)
- tdb2tcb, [221](#)
- planet_plug.C
 - calculateCholeskyCovarFunc, [223](#)
 - calculateDailyCovariance, [223](#)
 - calculateSpectra, [223](#)
 - doPlugin, [223](#)
 - fileOutput2, [223](#)
 - fileOutput3, [223](#)
 - findSmoothCurve, [223](#)
 - fitExponential, [223](#)
 - fitSineFunc, [223](#)
 - G_OMEGA, [224](#)
 - getHighFreqCovar, [223](#)
 - getHighFreqRes, [223](#)
 - graphicalInterface, [223](#)
 - help, [223](#)
 - modelFcn, [223](#)
 - obtainTimingResiduals, [224](#)
 - outputCovarianceFunction, [224](#)
 - outputMatrix, [224](#)
 - pgdevice, [224](#)
 - plot1, [224](#)
 - plot2, [224](#)
 - plot3, [224](#)
 - plot3a, [224](#)
 - plot4, [224](#)
 - plot5, [224](#)
 - plot6, [224](#)
 - plugVersionCheck, [224](#)
 - removeMean, [224](#)
 - skipprocess, [224](#)
 - skipstep2, [224](#)
 - T2fitSpectraRMS, [224](#)
 - writeFiles, [224](#)
- planet_ssb
 - observation, [47](#)
- planetShapiro
 - pulsar, [67](#)
- plk_plug.C
 - averagePts, [226](#)
 - binResiduals, [226](#)
 - callFit, [226](#)
 - changeFitParameters, [226](#)
 - changeParameters, [226](#)
 - checkMenu, [226](#)
 - checkMenu3, [227](#)
 - cholmode, [228](#)
 - covarFuncFile, [228](#)
 - dcmFile, [228](#)
 - deletePoint, [227](#)
 - displayStatistics, [227](#)
 - doPlot, [227](#)
 - drawAxisSel, [227](#)
 - drawMenu, [227](#)
 - drawMenu3, [227](#)
 - drawMenu3_2, [227](#)
 - drawOption, [227](#)
 - FITWAVES_harmonicStep, [228](#)
 - FITWAVES_n, [228](#)
 - FITWAVES_omega, [228](#)
 - FITWAVES_par, [228](#)
 - findMaxY, [227](#)
 - findMean, [227](#)
 - findMeanD, [227](#)
 - findMinY, [227](#)
 - flagStore, [228](#)
 - fortranMod, [227](#)
 - graphicalInterface, [227](#)
 - help, [227](#)
 - idPoint, [227](#)
 - lmst2, [227](#)
 - newTim, [227](#)
 - overPlotN, [227](#)
 - overPlotShapiro, [227](#)
 - plotFITWAVES_spec, [227](#)
 - plugVersionCheck, [228](#)
 - reFit, [228](#)
 - setLabel, [228](#)
 - setPlot, [228](#)
 - slaCalyd, [228](#)
 - slaClyd, [228](#)
 - sort, [228](#)
 - swapFit, [228](#)
 - viewModels, [228](#)
- plot1
 - glitch_plug.C, [190](#)
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- plot2
 - glitch_plug.C, [190](#)
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- plot3
 - glitch_plug.C, [190](#)
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- plot3a
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- plot4
 - glitch_plug.C, [190](#)
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- plot5
 - glitch_plug.C, [190](#)
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- plot6

- glitch_plug.C, [190](#)
- planet_plug.C, [224](#)
- spectralModel_plug.C, [242](#)
- plot7
 - glitch_plug.C, [190](#)
- plot8
 - glitch_plug.C, [190](#)
- plot9
 - glitch_plug.C, [190](#)
- plot_ppdot
 - basic_plug.C, [154](#)
- plotA_g
 - sigmaz_plug.C, [235](#)
- plotFITWAVES_spec
 - plk_plug.C, [227](#)
- plotGenPosn
 - GWbkgrdfromfile_plug.C, [199](#)
- plotGenSpectrum
 - GWbkgrdfromfile_plug.C, [199](#)
- plotHistogram
 - checkWhite_plug.C, [160](#)
 - fixData_plug.C, [181](#)
- plotMany_plug.C
 - calcYr, [229](#)
 - callFit, [229](#)
 - covarFuncFile2, [230](#)
 - doPlot, [229](#)
 - findMax, [229](#)
 - findMaxVal, [229](#)
 - findMean, [230](#)
 - findMin, [230](#)
 - findMinVal, [230](#)
 - fortranMod, [230](#)
 - graphicalInterface, [230](#)
 - help, [230](#)
 - lmst2, [230](#)
 - slaCalyd, [230](#)
 - slaClyd, [230](#)
- plotModel
 - interpolate_plug.C, [218](#)
- plotOmega_g
 - sigmaz_plug.C, [235](#)
- plotPosn
 - GWanisobkgrd_plug.C, [196](#)
 - GWbkgrd_plug.C, [197](#)
 - GWbkgrdfromfile_plug.C, [199](#)
 - GWdipolebkgrd_plug.C, [201](#)
 - GWgeneralanisobkgrd_plug.C, [206](#)
 - GWgeneralbkgrd_plug.C, [207](#)
 - GWsingle_plug.C, [212](#)
- plotResiduals
 - checkWhite_plug.C, [160](#)
 - GWanisobkgrd_plug.C, [196](#)
 - GWbkgrd_plug.C, [197](#)
 - GWbkgrdfromfile_plug.C, [199](#)
 - GWdipolebkgrd_plug.C, [201](#)
 - GWgeneralanisobkgrd_plug.C, [206](#)
 - GWgeneralbkgrd_plug.C, [207](#)
- GWsingle_plug.C, [212](#)
- interpolate_plug.C, [218](#)
- plotSpectrum
 - GWanisobkgrd_plug.C, [196](#)
 - GWbkgrd_plug.C, [197](#)
 - GWbkgrdfromfile_plug.C, [199](#)
 - GWdipolebkgrd_plug.C, [201](#)
 - GWgeneralanisobkgrd_plug.C, [206](#)
 - GWgeneralbkgrd_plug.C, [207](#)
 - GWsingle_plug.C, [212](#)
- plotout
 - fixData_plug.C, [181](#)
- plotoutSet
 - fixData_plug.C, [181](#)
- plugVersionCheck
 - addRed_plug.C, [145](#)
 - analyticChol_plug.C, [146](#)
 - applet_plug.C, [149](#)
 - autoDM_plug.C, [150](#)
 - averageData_plug.C, [152](#)
 - bary_plug.C, [153](#)
 - basic_plug.C, [154](#)
 - calcDMe_plug.C, [158](#)
 - checkWhite_plug.C, [160](#)
 - cholSpectra_plug.C, [161](#)
 - clock_plug.C, [163](#)
 - compareBackends_plug.C, [163](#)
 - compareDsets_plug.C, [165](#)
 - delays_plug.C, [166](#)
 - dm_plug.C, [172](#)
 - efacEquad_plug.C, [176](#)
 - fake_plug.C, [177](#)
 - fermi_plug.C, [178](#)
 - findCW_plug.C, [179](#)
 - fixData_plug.C, [182](#)
 - GWanisobkgrd_plug.C, [196](#)
 - GWbkgrd_plug.C, [197](#)
 - GWbkgrdfromfile_plug.C, [199](#)
 - GWdipolebkgrd_plug.C, [201](#)
 - GWevolve_plug.C, [204](#)
 - GWgeneralanisobkgrd_plug.C, [206](#)
 - GWgeneralbkgrd_plug.C, [207](#)
 - GWsens_plug.C, [211](#)
 - GWsingle_plug.C, [212](#)
 - GWwhiteLimit_plug.C, [214](#)
 - general2_plug.C, [183](#)
 - general_plug.C, [184](#)
 - glitch_plug.C, [191](#)
 - global_fitFunc_plug.C, [192](#)
 - globalDCM_fitFunc_plug.C, [193](#)
 - grTemplate_plug.C, [194](#)
 - gwm_plug.C, [208](#)
 - gwmStats_plug.C, [209](#)
 - icLimit_plug.C, [216](#)
 - interpolate_plug.C, [218](#)
 - matrix_plug.C, [219](#)
 - planet_plug.C, [224](#)
 - plk_plug.C, [228](#)

- publish_plug.C, 232
- sigmaz_plug.C, 236
- simRedNoise_plug.C, 238
- simulDM_plug.C, 239
- spectralModel_plug.C, 242
- spectrum_plug.C, 244
- splk_plug.C, 246
- transform_plug.C, 247
- plugin/GWanisobkgrd_plug.C, 194
- plugin/GWbkgrd_plug.C, 196
- plugin/GWbkgrdfromfile_plug.C, 197
- plugin/GWdetect_plug.C, 199
- plugin/GWdipolebkgrd_plug.C, 200
- plugin/GWevolve_plug.C, 201
- plugin/GWgeneralanisobkgrd_plug.C, 205
- plugin/GWgeneralbkgrd_plug.C, 206
- plugin/GWsens_plug.C, 209
- plugin/GWsingle_plug.C, 211
- plugin/GWwhiteLimit_plug.C, 212
- plugin/add_pulseNumber_plug.C, 144
- plugin/addRed_plug.C, 144
- plugin/analyticChol_plug.C, 146
- plugin/angle_plug.C, 147
- plugin/applet_plug.C, 147
- plugin/autoDM_plug.C, 149
- plugin/autoSpectralFit_plug.C, 150
- plugin/averageData_plug.C, 151
- plugin/bary_plug.C, 152
- plugin/basic_plug.C, 153
- plugin/calcDMe_plug.C, 154
- plugin/checkWhite_plug.C, 159
- plugin/cholSpectra_plug.C, 160
- plugin/clock_plug.C, 162
- plugin/compareBackends_plug.C, 163
- plugin/compareDsets_plug.C, 164
- plugin/delays_plug.C, 165
- plugin/designmatrix_plug.C, 166
- plugin/detectGWB_plug.C, 168
- plugin/detectGWBnew_plug.C, 169
- plugin/dm_plug.C, 171
- plugin/dmmodel_fitFunc_plug.C, 172
- plugin/efacEquad_plug.C, 173
- plugin/fake_plug.C, 176
- plugin/fermi_plug.C, 177
- plugin/findCW_plug.C, 178
- plugin/findCWs_plug.C, 179
- plugin/fixData_plug.C, 180
- plugin/general2_plug.C, 182
- plugin/general_plug.C, 183
- plugin/glast_plug.C, 184
- plugin/glitch_plug.C, 186
- plugin/global_fitFunc_plug.C, 191
- plugin/globalDCM_fitFunc_plug.C, 192
- plugin/grTemplate_plug.C, 193
- plugin/gwm_plug.C, 207
- plugin/gwmStats_plug.C, 208
- plugin/icLimit_plug.C, 214
- plugin/interpolate_plug.C, 216
- plugin/matrix_plug.C, 218
- plugin/mjk_plug.C, 219
- plugin/photons_plug.C, 220
- plugin/planet_plug.C, 221
- plugin/plk_plug.C, 225
- plugin/plotMany_plug.C, 228
- plugin/ppta_splug.C, 230
- plugin/publish_plug.C, 231
- plugin/sigmaz_plug.C, 232
- plugin/simRedNoise_plug.C, 237
- plugin/simulDM_plug.C, 238
- plugin/spectralModel_plug.C, 239
- plugin/spectrum_plug.C, 243
- plugin/splk_plug.C, 244
- plugin/transform_plug.C, 246
- pluginFitFunc
 - dmmodel_fitFunc_plug.C, 173
 - global_fitFunc_plug.C, 192
 - globalDCM_fitFunc_plug.C, 193
- polyco
 - polyco.C, 248
 - tempo2.h, 320
- polyco.C, 247
 - atimfake, 247
 - chebpc, 247
 - pcshft, 247
 - polyco, 248
 - tzFit, 248
- posPulsar
 - pulsar, 67
- ppta_splug.C
 - selectInterface, 231
- preProcess
 - preProcess.C, 248
 - tempo2.h, 320
- preProcess.C, 248
 - logicFlag, 248
 - preProcess, 248
 - processFlag, 248
 - processSimultaneous, 248
 - readWhiteNoiseModelFile, 249
 - useSelectFile, 249
- preProcessSimple
 - preProcessSimple.C, 249
 - tempo2.h, 320
- preProcessSimple.C, 249
 - preProcessSimple, 249
 - preProcessSimple1, 249
 - preProcessSimple2, 249
 - preProcessSimple3, 249
- preProcessSimple1
 - preProcessSimple.C, 249
 - tempo2.h, 320
- preProcessSimple2
 - preProcessSimple.C, 249
 - tempo2.h, 320
- preProcessSimple3
 - preProcessSimple.C, 249

- tempo2.h, [320](#)
- pred
 - sample, [73](#)
- prefit
 - parameter, [53](#)
- prefitErr
 - parameter, [53](#)
- prefitResidual
 - observation, [47](#)
- print_longdouble
 - TKlongdouble.C, [344](#)
- printGlitch
 - textOutput.C, [334](#)
- printflags
 - lm_control_struct, [40](#)
- printplugs
 - getInputs.C, [124](#)
- probks
 - efacEquad_plug.C, [176](#)
- processFlag
 - preProcess.C, [248](#)
 - tempo2.h, [320](#)
- processSimultaneous
 - preProcess.C, [248](#)
 - tempo2.h, [320](#)
- ProcessTempo2Objects
 - designmatrix_plug.C, [167](#)
- prtl
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- psr
 - ChebyModelInfo, [27](#)
- psrPos
 - observation, [47](#)
- psrangle
 - angle_plug.C, [147](#)
 - detectGWB_plug.C, [169](#)
 - detectGWBnew_plug.C, [171](#)
 - GWevolve_plug.C, [204](#)
 - GWsim.C, [130](#)
 - GWsim.h, [133](#)
- psrname
 - ChebyModel, [26](#)
 - T1Polyco, [74](#)
- publish_plug.C
 - dispParameter, [232](#)
 - fixDec, [232](#)
 - fixRA, [232](#)
 - nint_derived, [232](#)
 - parseExp, [232](#)
 - parseMinus, [232](#)
 - plugVersionCheck, [232](#)
 - rnd8, [232](#)
 - tempoOutput, [232](#)
- pulsar, [53](#)
 - addTNGlobalEQ, [60](#)
 - auto_constraints, [60](#)
 - AverageEpochWidth, [60](#)
 - AverageFlag, [60](#)
 - AverageResiduals, [60](#)
 - binaryModel, [60](#)
 - bootStrap, [60](#)
 - calcShapiro, [60](#)
 - cgw_angpol, [60](#)
 - cgw_cosinc, [60](#)
 - cgw_h0, [60](#)
 - cgw_mc, [60](#)
 - clk_offsE, [60](#)
 - clk_offsT, [60](#)
 - clk_offsV, [60](#)
 - clkOffsN, [60](#)
 - clock, [60](#)
 - clockFromOverride, [60](#)
 - constraints, [60](#)
 - correctTroposphere, [60](#)
 - covar, [61](#)
 - decjStrPost, [61](#)
 - decjStrPre, [61](#)
 - decsim, [61](#)
 - deleteFileName, [61](#)
 - dilateFreq, [61](#)
 - dmOffset, [61](#)
 - dmoffsCM, [61](#)
 - dmoffsCM_error, [61](#)
 - dmoffsCM_mjd, [61](#)
 - dmoffsCM_weight, [61](#)
 - dmoffsCMnum, [61](#)
 - dmoffsDM, [61](#)
 - dmoffsDM_error, [61](#)
 - dmoffsDM_mjd, [61](#)
 - dmoffsDM_weight, [61](#)
 - dmoffsDMnum, [61](#)
 - eclCoord, [61](#)
 - eopc04_file, [61](#)
 - ephemeris, [62](#)
 - filterStr, [62](#)
 - fitChisq, [62](#)
 - fitFunc, [62](#)
 - fitJump, [62](#)
 - fitMode, [62](#)
 - fitNfree, [62](#)
 - fitParamGlobal, [62](#)
 - fitParamGlobalK, [62](#)
 - fitParamI, [62](#)
 - fitParamK, [62](#)
 - fitinfo, [62](#)
 - fixedFormat, [62](#)
 - fjumpID, [62](#)
 - globalNfit, [62](#)
 - globalNoConstrain, [62](#)
 - gwb_decj, [62](#)
 - gwb_epoch, [63](#)
 - gwb_geom_c, [63](#)
 - gwb_geom_p, [63](#)
 - gwb_raj, [63](#)
 - gwb_width, [63](#)

gwecc_dec, 63
 gwecc_distance, 63
 gwecc_e, 63
 gwecc_epoch, 63
 gwecc_inc, 63
 gwecc_m1, 63
 gwecc_m2, 63
 gwecc_nodes_orientation, 63
 gwecc_orbital_period, 63
 gwecc_psrdist, 63
 gwecc_pulsarTermOn, 63
 gwecc_ra, 63
 gwecc_redshift, 63
 gwecc_theta_0, 63
 gwecc_theta_nodes, 63
 gwm_decj, 63
 gwm_dphase, 63
 gwm_epoch, 63
 gwm_phi, 63
 gwm_raj, 63
 gwsrc_across_i, 63
 gwsrc_across_i_e, 63
 gwsrc_across_r, 63
 gwsrc_across_r_e, 64
 gwsrc_aplus_i, 64
 gwsrc_aplus_i_e, 64
 gwsrc_aplus_r, 64
 gwsrc_aplus_r_e, 64
 gwsrc_dec, 64
 gwsrc_epoch, 64
 gwsrc_psrdist, 64
 gwsrc_ra, 64
 ifunc_weights, 64
 ifuncE, 64
 ifuncN, 64
 ifuncT, 64
 ifuncV, 64
 ipm, 64
 JPL_EPHEMERIS, 64
 jboFormat, 64
 jumpStr, 64
 jumpVal, 64
 jumpValErr, 64
 nCompanion, 65
 nDMEvents, 65
 nFit, 65
 nGlobal, 65
 nJumps, 65
 nParam, 65
 nPhaseJump, 65
 nQuad, 66
 nStorePrecision, 66
 nT2efac, 66
 nT2equad, 66
 nTNBandNoise, 66
 nTNECORR, 66
 nTNEF, 66
 nTNEQ, 66
 nTNGroupNoise, 66
 nTNSQ, 66
 nTNShapeletEvents, 66
 nTelDX, 66
 nTelDY, 66
 nTelDZ, 66
 nToffset, 66
 nWhite, 66
 nWhite_dm, 66
 name, 64
 nconstraints, 65
 ndmx, 65
 ne_sw, 65
 nits, 65
 noWarnings, 65
 nob, 65
 obsn, 66
 offset, 66
 offset_e, 66
 outputTMatrix, 66
 param, 66
 passStr, 66
 phaseJump, 67
 phaseJumpDir, 67
 phaseJumpID, 67
 planetShapiro, 67
 posPulsar, 67
 quad_across_i, 67
 quad_across_i_e, 67
 quad_across_r, 67
 quad_across_r_e, 67
 quad_aplus_i, 67
 quad_aplus_i_e, 67
 quad_aplus_r, 67
 quad_aplus_r_e, 67
 quad_ifunc_c_DEC, 67
 quad_ifunc_c_RA, 67
 quad_ifunc_geom_c, 67
 quad_ifunc_geom_p, 67
 quad_ifunc_p_DEC, 67
 quad_ifunc_p_RA, 67
 quad_ifuncE_c, 67
 quad_ifuncE_p, 68
 quad_ifuncN_c, 68
 quad_ifuncN_p, 68
 quad_ifuncT_c, 68
 quad_ifuncT_p, 68
 quad_ifuncV_c, 68
 quad_ifuncV_p, 68
 quadDEC, 68
 quadEpoch, 68
 quadRA, 68
 rajStrPost, 68
 rajStrPre, 68
 rasim, 68
 rescaleErrChisq, 68
 rmsPost, 68
 rmsPre, 68

- robust, 68
- setTelVelX, 68
- setTelVelY, 68
- setTelVelZ, 68
- setUnits, 68
- simflag, 68
- sorted, 68
- storePrec, 68
- swm, 69
- t2cMethod, 69
- T2efacFlagID, 69
- T2efacFlagVal, 69
- T2efacVal, 69
- T2equadFlagID, 69
- T2equadFlagVal, 69
- T2equadVal, 69
- T2globalEfac, 69
- TNBandDMAmp, 70
- TNBandDMC, 70
- TNBandDMGam, 70
- TNBandNoiseAmp, 70
- TNBandNoiseC, 70
- TNBandNoiseGam, 70
- TNBandNoiseHF, 70
- TNBandNoiseLF, 70
- TNDMAmp, 70
- TNDMC, 70
- TNDMCoeffs, 70
- TNDMEvAmp, 70
- TNDMEvGam, 70
- TNDMEvLength, 70
- TNDMEvLin, 70
- TNDMEvOff, 70
- TNDMEvQuad, 70
- TNDMEvStart, 70
- TNDMGam, 70
- TNECORRFlagID, 70
- TNECORRFlagVal, 70
- TNECORRVal, 70
- TNEFFlagID, 70
- TNEFFlagVal, 70
- TNEFVal, 70
- TNEQFlagID, 71
- TNEQFlagVal, 71
- TNEQVal, 71
- TNGlobalEF, 71
- TNGlobalEQ, 71
- TNGroupNoiseAmp, 71
- TNGroupNoiseC, 71
- TNGroupNoiseFlagID, 71
- TNGroupNoiseFlagVal, 71
- TNGroupNoiseGam, 71
- TNRedAmp, 71
- TNRedC, 71
- TNRedCoeffs, 71
- TNRedCorner, 71
- TNRedFLOw, 71
- TNRedGam, 71
- TNSQFlagID, 71
- TNSQFlagVal, 71
- TNSQVal, 71
- TNShapeletEvFScale, 71
- TNShapeletEvN, 71
- TNShapeletEvPos, 71
- TNShapeletEvWidth, 71
- TNsubtractDM, 71
- TNsubtractRed, 71
- tOffset, 71
- tOffset_f1, 72
- tOffset_f2, 72
- tOffset_t1, 72
- tOffset_t2, 72
- tOffsetFlags, 72
- tOffsetSite, 72
- telDX_e, 69
- telDX_t, 69
- telDX_v, 69
- telDX_vel, 69
- telDX_vel_e, 69
- telDY_e, 69
- telDY_t, 69
- telDY_v, 69
- telDY_vel, 69
- telDY_vel_e, 69
- telDZ_e, 69
- telDZ_t, 69
- telDZ_v, 69
- telDZ_vel, 69
- telDZ_vel_e, 69
- tempo1, 69
- tempo2.h, 313
- timeEphemeris, 70
- ToAextraCovar, 71
- tzrsite, 72
- units, 72
- useCalceph, 72
- useTNOrth, 72
- velPulsar, 72
- wave_cos, 72
- wave_cos_dm, 72
- wave_cos_dm_err, 72
- wave_cos_err, 72
- wave_sine, 72
- wave_sine_dm, 72
- wave_sine_dm_err, 72
- wave_sine_err, 72
- waveScale, 72
- whiteNoiseModelFile, 72
- pulseN
 - observation, 47
- pvsun
 - jpl_eph_data, 40
- quad_across_i
 - pulsar, 67
- quad_across_i_e
 - pulsar, 67

- quad_across_r
 - pulsar, [67](#)
- quad_across_r_e
 - pulsar, [67](#)
- quad_aplus_i
 - pulsar, [67](#)
- quad_aplus_i_e
 - pulsar, [67](#)
- quad_aplus_r
 - pulsar, [67](#)
- quad_aplus_r_e
 - pulsar, [67](#)
- quad_ifunc_c_DEC
 - pulsar, [67](#)
- quad_ifunc_c_RA
 - pulsar, [67](#)
- quad_ifunc_geom_c
 - pulsar, [67](#)
- quad_ifunc_geom_p
 - pulsar, [67](#)
- quad_ifunc_p_DEC
 - pulsar, [67](#)
- quad_ifunc_p_RA
 - pulsar, [67](#)
- quad_ifuncE_c
 - pulsar, [67](#)
- quad_ifuncE_p
 - pulsar, [68](#)
- quad_ifuncN_c
 - pulsar, [68](#)
- quad_ifuncN_p
 - pulsar, [68](#)
- quad_ifuncT_c
 - pulsar, [68](#)
- quad_ifuncT_p
 - pulsar, [68](#)
- quad_ifuncV_c
 - pulsar, [68](#)
- quad_ifuncV_p
 - pulsar, [68](#)
- quadDEC
 - pulsar, [68](#)
- quadEpoch
 - pulsar, [68](#)
- quadRA
 - pulsar, [68](#)
- RAND_M
 - T2toolkit.C, [290](#)
- RAND_N
 - T2toolkit.C, [290](#)
- RESETCOLOR
 - TKlog.h, [343](#)
- rajStrPost
 - pulsar, [68](#)
- rajStrPre
 - pulsar, [68](#)
- random
 - bootstrap.C, [80](#)
- random2
 - readEphemeris.C, [253](#)
- rasim
 - pulsar, [68](#)
- rawOut
 - calcDMe_plug.C, [158](#)
- reFit
 - plk_plug.C, [228](#)
- read_char
 - read_fortran.h, [251](#)
- read_character
 - read_fortran.h, [251](#)
- read_character2
 - read_fortran2.h, [252](#)
- read_double
 - read_fortran.h, [251](#)
- read_double2
 - read_fortran2.h, [252](#)
- read_float
 - read_fortran.h, [251](#)
- read_float2
 - read_fortran2.h, [252](#)
- read_fortran.h, [250](#)
 - c_fileptr, [251](#)
 - close_file, [251](#)
 - open_file, [251](#)
 - read_char, [251](#)
 - read_character, [251](#)
 - read_double, [251](#)
 - read_float, [251](#)
 - read_int, [251](#)
 - read_record_int, [251](#)
 - swapByte, [251](#)
- read_fortran2.h, [251](#)
 - c_fileptr2, [252](#)
 - close_file2, [252](#)
 - open_file2, [252](#)
 - read_character2, [252](#)
 - read_double2, [252](#)
 - read_float2, [252](#)
 - read_int2, [252](#)
 - read_record_int2, [252](#)
 - swapByte2, [252](#)
- read_int
 - read_fortran.h, [251](#)
- read_int2
 - read_fortran2.h, [252](#)
- read_record_int
 - read_fortran.h, [251](#)
- read_record_int2
 - read_fortran2.h, [252](#)
- readAliases
 - observatory.C, [144](#)
- readCurrentSheet
 - sw_delay.C, [263](#)
- readEphemeris
 - readEphemeris.C, [253](#)
 - tempo2.h, [320](#)

- readEphemeris.C, [253](#)
 - gasdev, [253](#)
 - MAX_SHOTS, [253](#)
 - random2, [253](#)
 - readEphemeris, [253](#)
- readEphemeris_calceph
 - readEphemeris_calceph.C, [254](#)
 - tempo2.h, [320](#)
- readEphemeris_calceph.C, [254](#)
 - convertUnits, [254](#)
 - readEphemeris_calceph, [254](#)
- readJBO_bat
 - readJBO_bat.C, [255](#)
 - tempo2.h, [320](#)
- readJBO_bat.C, [254](#)
 - date2mjd, [255](#)
 - makechars, [255](#)
 - readJBO_bat, [255](#)
 - swap4, [255](#)
 - swap8, [255](#)
- readObsFile
 - tempo2.h, [320](#)
- readObservatoryFile
 - observatory.C, [144](#)
- readOneEphemeris
 - tempo2.h, [321](#)
- readParfile
 - readParfile.C, [256](#)
 - tempo2.h, [321](#)
- readParfile.C, [255](#)
 - checkAllSet, [256](#)
 - checkLine, [256](#)
 - getValue, [256](#)
 - readParfile, [256](#)
 - readParfileGlobal, [256](#)
 - readSimpleParfile, [256](#)
 - readValue, [256](#)
 - removeCR, [256](#)
 - setupParameterFileDefaults, [256](#)
- readParfileGlobal
 - readParfile.C, [256](#)
 - tempo2.h, [321](#)
- readSimpleParfile
 - readParfile.C, [256](#)
 - tempo2.h, [321](#)
- readTim
 - readTimfile.C, [257](#)
- readTimfile
 - readTimfile.C, [257](#)
 - tempo2.h, [321](#)
- readTimfile.C, [256](#)
 - readTim, [257](#)
 - readTimfile, [257](#)
 - removeCR2, [257](#)
 - writeTim, [257](#)
- readUinv
 - globalDCM_fitFunc_plug.C, [193](#)
- readValue
 - readParfile.C, [256](#)
- readWhiteNoiseModelFile
 - preProcess.C, [249](#)
- readin
 - sigmaz_plug.C, [235](#)
 - TKspectrum.C, [354](#)
 - TKspectrum.h, [359](#)
- real
 - complexVal, [30](#)
- reclen
 - IFTEphemeris, [38](#)
- recordPrecision
 - storePrecision.C, [261](#)
 - tempo2.h, [321](#)
- resize
 - jpl_eph_data, [40](#)
- reference_phase
 - T1Polyco, [74](#)
- remove_white
 - get_obsCoord.C, [123](#)
- removeCR
 - readParfile.C, [256](#)
- removeCR2
 - readTimfile.C, [257](#)
- removeMean
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- rescaleErrChisq
 - pulsar, [68](#)
- resetDMandF0
 - calcDMe_plug.C, [156](#)
- residual
 - observation, [47](#)
- residualTracking
 - formResiduals.C, [121](#)
- rk4rms
 - age.C, [79](#)
- rmsPost
 - pulsar, [68](#)
- rmsPre
 - pulsar, [68](#)
- rnd8
 - applet_plug.C, [148](#)
 - general2_plug.C, [183](#)
 - general_plug.C, [184](#)
 - publish_plug.C, [232](#)
- robust
 - pulsar, [68](#)
- roemer
 - observation, [47](#)
- root2
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- routine
 - storePrecision, [73](#)
- Rs
 - GWsim.C, [130](#)
 - GWsim.h, [133](#)

- runPlugin
 - compareBackends_plug.C, 163
- RungeKuttaCashKarp
 - GWevolve_plug.C, 204
- RungeKuttaStep
 - GWevolve_plug.C, 204
- SAFETY
 - GWevolve_plug.C, 203
- SECDAY
 - fermi_plug.C, 178
 - tempo2.h, 311
- SECDAYI
 - tempo2.h, 311
- SI_UNITS
 - tempo2.h, 312
- SIGN
 - GWevolve_plug.C, 203
 - GWwhiteLimit_plug.C, 213
- SOLAR_MASS
 - GWevolve_plug.C, 203
 - tempo2.h, 312
- SOLAR_RADIUS
 - tempo2.h, 312
- SPEED_LIGHT
 - GWevolve_plug.C, 204
 - tempo2.h, 312
- SQR
 - glitch_plug.C, 188
- STDC_HEADERS
 - config.h, 101
- SWAP
 - efacEquad_plug.C, 175
 - glast_plug.C, 185
- SWAP_MACRO
 - jpleph.c, 139
- sample, 73
 - actual, 73
 - e, 73
 - interpolate_plug.C, 217
 - pred, 73
 - x, 73
 - y, 73
- samples
 - TabulatedFunction, 77
- sat
 - observation, 48
- sat_day
 - observation, 48
- sat_sec
 - observation, 48
- saturn_earth
 - observation, 48
- saveparams
 - mjk_plug.C, 220
- scale_diag
 - lm_control_struct, 40
- scale_param
 - units.C, 367
- script
 - fixData_plug.C, 182
- searchGridPos
 - GWdetect_plug.C, 200
- secularMotion
 - secularMotion.C, 258
 - tempo2.h, 321
- secularMotion.C, 257
 - secularMotion, 258
- secyear
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- segments
 - ChebyModelSet, 28
 - T1PolycoSet, 75
- selectData
 - dm_plug.C, 172
- selectInterface
 - ppta_splug.C, 231
- sessionSeparation
 - calcDMe_plug.C, 158
- setAllDeleted
 - calcDMe_plug.C, 156
- setFitParams
 - calcDMe_plug.C, 157
- setLabel
 - plk_plug.C, 228
- setPlot
 - plk_plug.C, 228
- setPlugPath
 - getInputs.C, 124
 - tempo2.h, 321
- setStart
 - tempo2.h, 321
- setTelVelX
 - pulsar, 68
- setTelVelY
 - pulsar, 68
- setTelVelZ
 - pulsar, 68
- setUnits
 - pulsar, 68
- setup3C66B
 - GWevolve_plug.C, 204
- setupGW
 - GWsim.C, 130
 - GWsim.h, 133
- setupParameterFileDefaults
 - readParfile.C, 256
 - tempo2.h, 321
- setupPulsar
 - GWwhiteLimit_plug.C, 214
- setupPulsar_GWsim
 - GWsim.C, 130
 - GWsim.h, 133
- setupTest
 - GWevolve_plug.C, 204
- setupgeneralGW

- GWsim.C, [130](#)
- GWsim.h, [133](#)
- shapiro_delay
 - shapiro_delay.C, [259](#)
 - tempo2.h, [321](#)
- shapiro_delay.C, [258](#)
- shapiro_delay, [259](#)
- shapiroDelayJupiter
 - observation, [48](#)
- shapiroDelayNeptune
 - observation, [48](#)
- shapiroDelaySaturn
 - observation, [48](#)
- shapiroDelaySun
 - observation, [48](#)
- shapiroDelayUranus
 - observation, [48](#)
- shapiroDelayVenus
 - observation, [48](#)
- shklovskii
 - observation, [48](#)
- shortlabel
 - parameter, [53](#)
- shuffle
 - checkWhite_plug.C, [160](#)
 - GWwhiteLimit_plug.C, [214](#)
- shufflePoints
 - checkWhite_plug.C, [160](#)
- shufflePts
 - sigmaz_plug.C, [235](#)
- sigmai
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- sigmaz_plug.C
 - addvar, [235](#)
 - calcSigmaz, [234](#)
 - calcSpline, [234](#)
 - calculateGWlim, [234](#)
 - convert_gravWaveBackground_fit, [234](#)
 - convert_gravWaveBackground_noFit, [235](#)
 - data, [235](#)
 - doplot, [235](#)
 - fit4, [235](#)
 - fitv, [235](#)
 - getprtj, [235](#)
 - getweights, [235](#)
 - graphicalInterface, [235](#)
 - help, [235](#)
 - indexx8, [235](#)
 - indx, [235](#)
 - linfile, [235](#)
 - MAX_GWS, [234](#)
 - mat20, [235](#)
 - nbintype, [236](#)
 - ncubic, [236](#)
 - ncubics, [236](#)
 - ndim, [236](#)
 - nformat, [236](#)
 - npt, [236](#)
 - npt1last, [236](#)
 - npt2last, [236](#)
 - ntau, [236](#)
 - ntunits, [236](#)
 - nusewt, [236](#)
 - nwriteres, [236](#)
 - nxunits, [236](#)
 - permax, [236](#)
 - plotA_g, [235](#)
 - plotOmega_g, [235](#)
 - plugVersionCheck, [236](#)
 - prtl, [236](#)
 - readin, [235](#)
 - root2, [236](#)
 - secyear, [236](#)
 - shufflePts, [235](#)
 - sigmai, [236](#)
 - simWhiteFunc, [235](#)
 - sortTimes, [235](#)
 - SplineBlend, [235](#)
 - SplineCurve, [235](#)
 - SplineKnots, [235](#)
 - SplinePoint, [235](#)
 - tauday, [236](#)
 - tauensure, [236](#)
 - taulog, [236](#)
 - taumax, [236](#)
 - taumin, [236](#)
 - tausec, [236](#)
 - tauyear, [236](#)
 - tdiffmin, [236](#)
 - tmax, [236](#)
 - tmin, [237](#)
 - utfirst, [237](#)
 - utjd, [237](#)
 - utjd1, [237](#)
 - utjd2, [237](#)
 - utjdlast, [237](#)
 - utlast, [237](#)
 - utmean, [237](#)
 - XY, [234](#)
 - xmax, [237](#)
 - xmin, [237](#)
 - simRedNoise_plug.C
 - doPlugin, [238](#)
 - getRedNoiseRealisation, [238](#)
 - graphicalInterface, [238](#)
 - help, [238](#)
 - plugVersionCheck, [238](#)
 - simWhiteFunc
 - sigmaz_plug.C, [235](#)
 - simflag
 - pulsar, [68](#)
 - simplePlot
 - tempo2.h, [321](#)
 - simulDM_plug.C
 - doPlot, [239](#)

- doplugin, [239](#)
- graphicalInterface, [239](#)
- help, [239](#)
- MAX_DM, [239](#)
- plugVersionCheck, [239](#)
- sineFunc
 - TKspectrum.C, [354](#)
 - TKspectrum.h, [359](#)
- sinefunc
 - t2fit_ifunc.C, [279](#)
 - t2fit_ifunc.h, [281](#)
- sinl
 - TKlongdouble.float128.h, [346](#)
 - TKlongdouble.h, [347](#)
- siteName
 - MeteorologyFunction, [42](#)
- siteVel
 - observation, [48](#)
- sitename
 - ChebyModel, [26](#)
 - T1Polyco, [74](#)
- skipprocess
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- skipstep2
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- sl_alpha
 - gwgenSpec, [35](#)
- sl_amp
 - gwgenSpec, [35](#)
- sla_CALDJ
 - glast_plug.C, [185](#)
- sla_CLDJ
 - glast_plug.C, [185](#)
- slaCalyd
 - clock_plug.C, [162](#)
 - dm_plug.C, [172](#)
 - glast_plug.C, [185](#)
 - plk_plug.C, [228](#)
 - plotMany_plug.C, [230](#)
- slaClyd
 - clock_plug.C, [162](#)
 - dm_plug.C, [172](#)
 - glast_plug.C, [185](#)
 - plk_plug.C, [228](#)
 - plotMany_plug.C, [230](#)
- smoothWidth
 - calcDMe_plug.C, [158](#)
- solarWindModel
 - dm_delays.C, [111](#)
 - sw_delay.C, [263](#)
 - tempo2.h, [321](#)
- sort
 - efacEquad_plug.C, [176](#)
 - plk_plug.C, [228](#)
- sortSamples
 - interpolate_plug.C, [218](#)
- sortTimes
 - sigmaz_plug.C, [235](#)
- sortToAs
 - sortToAs.C, [260](#)
 - tempo2.h, [321](#)
- sortToAs.C, [259](#)
 - compareObs, [260](#)
 - sortToAs, [260](#)
- sorted
 - pulsar, [68](#)
- sortit
 - GWwhiteLimit_plug.C, [214](#)
- span
 - T1Polyco, [75](#)
- spectralModel_plug.C
 - calculateCholeskyCovarFunc, [241](#)
 - calculateDailyCovariance, [241](#)
 - calculateSpectra, [241](#)
 - doPlugin, [241](#)
 - fileOutput2, [241](#)
 - fileOutput3, [241](#)
 - findSmoothCurve, [241](#)
 - fitExponential, [241](#)
 - fitSineFunc, [241](#)
 - G_OMEGA, [242](#)
 - getHighFreqCovar, [241](#)
 - getHighFreqRes, [241](#)
 - graphicalInterface, [241](#)
 - help, [241](#)
 - obtainTimingResiduals, [241](#)
 - outputCovarianceFunction, [241](#)
 - outputMatrix, [242](#)
 - pgdevice, [242](#)
 - plot1, [242](#)
 - plot2, [242](#)
 - plot3, [242](#)
 - plot3a, [242](#)
 - plot4, [242](#)
 - plot5, [242](#)
 - plot6, [242](#)
 - plugVersionCheck, [242](#)
 - removeMean, [242](#)
 - skipprocess, [242](#)
 - skipstep2, [242](#)
 - writeFiles, [242](#)
- spectrum_plug.C
 - checkMenu, [244](#)
 - covarFuncFile, [244](#)
 - dcmFile, [244](#)
 - doPlugin, [244](#)
 - drawMenu, [244](#)
 - drawOption, [244](#)
 - graphicalInterface, [244](#)
 - help, [244](#)
 - identify, [244](#)
 - MAX_ID, [244](#)
 - model, [244](#)
 - plugVersionCheck, [244](#)

- sphharm
 - GWsim.C, [130](#)
 - GWsim.h, [133](#)
- SplineBlend
 - sigmaz_plug.C, [235](#)
- SplineCurve
 - sigmaz_plug.C, [235](#)
- SplineKnots
 - sigmaz_plug.C, [235](#)
- splineOut
 - calcDMe_plug.C, [158](#)
- SplinePoint
 - sigmaz_plug.C, [235](#)
- splk_plug.C
 - callFit, [245](#)
 - covarFuncFile, [246](#)
 - dcmFile, [246](#)
 - deletePoint, [245](#)
 - doPlot, [245](#)
 - findMax, [245](#)
 - findMean, [245](#)
 - findMin, [245](#)
 - fortranMod, [245](#)
 - graphicalInterface, [245](#)
 - idPoint, [246](#)
 - plugVersionCheck, [246](#)
- st_alpha
 - gwgenSpec, [35](#)
- st_amp
 - gwgenSpec, [35](#)
- standardConstraintFunctions
 - constraints.C, [103](#)
 - constraints.h, [104](#)
- start_sessions
 - calcDMe_plug.C, [159](#)
- startJD
 - IFTEphemeris, [38](#)
- stepJD
 - IFTEphemeris, [38](#)
- stepbound
 - lm_control_struct, [40](#)
- storePrec
 - pulsar, [68](#)
- storePrecision, [73](#)
 - comment, [73](#)
 - minPrec, [73](#)
 - routine, [73](#)
 - tempo2.h, [313](#)
- storePrecision.C, [260](#)
 - recordPrecision, [261](#)
- storeVal
 - GWwhiteLimit_plug.C, [214](#)
- sun_earth
 - observation, [48](#)
- sun_ssb
 - observation, [49](#)
- surfaceAtmosphericPressureTables
 - tropo.C, [365](#)
- sw_delay.C, [261](#)
 - amod, [262](#)
 - calcRotN, [262](#)
 - convertEcliptic, [262](#)
 - elsun2, [262](#)
 - findAngle, [262](#)
 - MAX_CURRENT, [262](#)
 - mcl2, [262](#)
 - mjd2date, [263](#)
 - outputResults, [263](#)
 - readCurrentSheet, [263](#)
 - solarWindModel, [263](#)
- swap4
 - readJBO_bat.C, [255](#)
- swap8
 - readJBO_bat.C, [255](#)
- swap_bytes
 - jpl_eph_data, [40](#)
- swap_endian
 - IFTEphemeris, [38](#)
- swapByte
 - read_fortran.h, [251](#)
- swapByte2
 - read_fortran2.h, [252](#)
- swapFit
 - plk_plug.C, [228](#)
- swm
 - pulsar, [69](#)
- t
 - lmcurve_data_struct, [41](#)
- T1
 - tempo2pred.h, [327](#)
- t1
 - T2Predictor, [76](#)
- T1P_grabInt
 - t1polyco.c, [264](#)
- T1P_grabLongDouble
 - t1polyco.c, [264](#)
- T1P_grabString
 - t1polyco.c, [264](#)
- T1Polyco, [74](#)
 - binary_frequency, [74](#)
 - binary_phase, [74](#)
 - coeff, [74](#)
 - date_string, [74](#)
 - dm, [74](#)
 - doppler, [74](#)
 - frequency_obs, [74](#)
 - frequency_psr_0, [74](#)
 - log10rms, [74](#)
 - mjd_mid, [74](#)
 - ncoeff, [74](#)
 - psrname, [74](#)
 - reference_phase, [74](#)
 - sitename, [74](#)
 - span, [75](#)
 - utc_string, [75](#)
- T1Polyco_GetFrequency

- t1polyco.c, [264](#)
 - tempo2pred_int.h, [330](#)
- T1Polyco_GetPhase
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [330](#)
- T1Polyco_Read
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- T1Polyco_Read_NewFormat
 - t1polyco.c, [264](#)
- T1Polyco_Write
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- T1PolycoSet, [75](#)
 - nsegments, [75](#)
 - segments, [75](#)
- T1PolycoSet_Destroy
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- T1PolycoSet_GetFrequency
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- T1PolycoSet_GetNearest
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- T1PolycoSet_GetPhase
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- T1PolycoSet_Read
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- T1PolycoSet_Write
 - t1polyco.c, [264](#)
 - tempo2pred_int.h, [331](#)
- t1polyco.c, [263](#)
 - T1P_grabInt, [264](#)
 - T1P_grabLongDouble, [264](#)
 - T1P_grabString, [264](#)
 - T1Polyco_GetFrequency, [264](#)
 - T1Polyco_GetPhase, [264](#)
 - T1Polyco_Read, [264](#)
 - T1Polyco_Read_NewFormat, [264](#)
 - T1Polyco_Write, [264](#)
 - T1PolycoSet_Destroy, [264](#)
 - T1PolycoSet_GetFrequency, [264](#)
 - T1PolycoSet_GetNearest, [264](#)
 - T1PolycoSet_GetPhase, [264](#)
 - T1PolycoSet_Read, [264](#)
 - T1PolycoSet_Write, [264](#)
- T2-PTAmodel.C, [264](#)
 - computeU, [265](#)
 - T2_PTAmodel, [265](#)
 - updateT2_PTA, [265](#)
- T2_PTAmodel
 - T2-PTAmodel.C, [265](#)
 - tempo2.h, [321](#)
- T2_SVD_TOL
 - t2fit.C, [270](#)
- T2C_IAU2000B
 - tempo2.h, [312](#)
- T2C_TEMPO
 - tempo2.h, [312](#)
- t2Fit
 - t2fit.C, [270](#)
 - t2fit.h, [272](#)
- t2Fit_buildConstraintsMatrix
 - t2fit.C, [270](#)
 - t2fit.h, [272](#)
- t2Fit_buildDesignMatrix
 - t2fit.C, [270](#)
 - t2fit.h, [272](#)
- t2Fit_fillFitInfo
 - t2fit.C, [270](#)
 - t2fit.h, [272](#)
- t2Fit_fillFitInfo_INNER
 - t2fit.C, [270](#)
- t2Fit_fillGlobalFitInfo
 - t2fit.C, [270](#)
 - t2fit.h, [272](#)
- t2Fit_getFitData
 - t2fit.C, [271](#)
 - t2fit.h, [272](#)
- t2Fit_updateParameters
 - t2fit.C, [271](#)
 - t2fit.h, [272](#)
- t2FitFunc_binaryModels
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2FitFunc_dmmodelCM
 - t2fit_dmmodel.C, [273](#)
 - t2fit_dmmodel.h, [274](#)
- t2FitFunc_dmmodelDM
 - t2fit_dmmodel.C, [273](#)
 - t2fit_dmmodel.h, [275](#)
- t2FitFunc_fitwaves
 - t2fit_fitwaves.C, [275](#)
 - t2fit_fitwaves.h, [276](#)
- t2FitFunc_ifunc
 - t2fit_ifunc.C, [279](#)
 - t2fit_ifunc.h, [281](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2FitFunc_jump
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2FitFunc_miscDm
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2FitFunc_planet
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2FitFunc_sifunc
 - t2fit_ifunc.C, [279](#)
 - t2fit_ifunc.h, [281](#)
- t2FitFunc_stdDm
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)

t2FitFunc_stdFreq
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2FitFunc_stdGlitch
 t2fit_glitch.C, 277
 t2fit_glitch.h, 278
 t2FitFunc_stdGravWav
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2FitFunc_stdPosition
 t2fit_position.C, 281
 t2fit_position.h, 282
 t2FitFunc_telPos
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2FitFunc_zero
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 T2Predictor, 75
 cheby, 76
 kind, 76
 modelset, 76
 t1, 76
 T2Predictor_Copy
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_Destroy
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_FRead
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_FWrite
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_GetEndFreq
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_GetEndMJD
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_GetFrequency
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_GetPSRName
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_GetPhase
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_GetPlan
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_GetPlan_Ext
 tempo2pred.c, 325
 tempo2pred.h, 327
 T2Predictor_GetSiteName
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_GetStartFreq
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_GetStartMJD
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_Init
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_Insert
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_Keep
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_Kind
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_Read
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2Predictor_Write
 tempo2pred.c, 325
 tempo2pred.h, 328
 T2PredictorKind
 tempo2pred.h, 327
 t2UpdateFunc_binaryModels
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2UpdateFunc_dmmodelCM
 t2fit_dmmodel.C, 273
 t2fit_dmmodel.h, 275
 t2UpdateFunc_dmmodelIDM
 t2fit_dmmodel.C, 273
 t2fit_dmmodel.h, 275
 t2UpdateFunc_fitwaves
 t2fit_fitwaves.C, 275
 t2fit_fitwaves.h, 276
 t2UpdateFunc_ifunc
 t2fit_ifunc.C, 279
 t2fit_ifunc.h, 281
 t2fit_stdFitFuncs.h, 286
 t2UpdateFunc_jump
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2UpdateFunc_miscDm
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2UpdateFunc_planet
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2UpdateFunc_simpleAdd
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286
 t2UpdateFunc_simpleMinus
 t2fit_stdFitFuncs.C, 284
 t2fit_stdFitFuncs.h, 286

- t2UpdateFunc_stdFreq
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2UpdateFunc_stdGlitch
 - t2fit_glitch.C, [277](#)
 - t2fit_glitch.h, [278](#)
- t2UpdateFunc_stdGravWav
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2UpdateFunc_stdPosition
 - t2fit_position.C, [281](#)
 - t2fit_position.h, [282](#)
- t2UpdateFunc_telPos
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- t2UpdateFunc_zero
 - t2fit_stdFitFuncs.C, [284](#)
 - t2fit_stdFitFuncs.h, [286](#)
- T2accel.C, [265](#)
 - accel_lsqr, [267](#)
 - accel_multMatrix, [267](#)
 - accel_multMatrixVec, [267](#)
 - accel_uinv, [267](#)
 - F77_dgels, [266](#), [267](#)
 - F77_dgemm, [266](#), [267](#)
 - F77_dgemv, [266](#), [267](#)
 - F77_dpotf2, [267](#)
 - F77_dtptri, [267](#)
 - F77_dtrmm, [267](#)
 - useT2accel, [267](#)
- T2accel.h, [268](#)
 - ACCEL_LSQ, [268](#)
 - ACCEL_MULTMATRIX, [268](#)
 - ACCEL_UINV, [269](#)
 - accel_lsqr, [269](#)
 - accel_multMatrix, [269](#)
 - accel_multMatrixVec, [269](#)
 - accel_uinv, [269](#)
 - useT2accel, [269](#)
- t2cMethod
 - pulsar, [69](#)
- T2calculateCholesky
 - choleskyRoutines.C, [93](#)
 - choleskyRoutines.h, [96](#)
- T2calculateCovarFunc
 - choleskyRoutines.C, [93](#)
 - choleskyRoutines.h, [96](#)
- T2calculateDailyCovariance
 - choleskyRoutines.C, [93](#)
 - choleskyRoutines.h, [96](#)
- T2calculateSpectra
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- T2cholDecomposition
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- T2cubicFit
 - choleskyRoutines.C, [94](#)
- choleskyRoutines.h, [96](#)
- T2efacFlagID
 - pulsar, [69](#)
- T2efacFlagVal
 - pulsar, [69](#)
- T2efacVal
 - pulsar, [69](#)
- T2equadFlagID
 - pulsar, [69](#)
- T2equadFlagVal
 - pulsar, [69](#)
- T2equadVal
 - pulsar, [69](#)
- T2findSmoothCurve
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- t2fit.C, [269](#)
 - T2_SVD_TOL, [270](#)
 - t2Fit, [270](#)
 - t2Fit_buildConstraintsMatrix, [270](#)
 - t2Fit_buildDesignMatrix, [270](#)
 - t2Fit_fillFitInfo, [270](#)
 - t2Fit_fillFitInfo_INNER, [270](#)
 - t2Fit_fillGlobalFitInfo, [270](#)
 - t2Fit_getFitData, [271](#)
 - t2Fit_updateParameters, [271](#)
- t2fit.h, [271](#)
 - t2Fit, [272](#)
 - t2Fit_buildConstraintsMatrix, [272](#)
 - t2Fit_buildDesignMatrix, [272](#)
 - t2Fit_fillFitInfo, [272](#)
 - t2Fit_fillGlobalFitInfo, [272](#)
 - t2Fit_getFitData, [272](#)
 - t2Fit_updateParameters, [272](#)
- t2fit_dmmodel.C, [272](#)
 - t2FitFunc_dmmodelCM, [273](#)
 - t2FitFunc_dmmodelDM, [273](#)
 - t2UpdateFunc_dmmodelCM, [273](#)
 - t2UpdateFunc_dmmodelDM, [273](#)
- t2fit_dmmodel.h, [273](#)
 - t2FitFunc_dmmodelCM, [274](#)
 - t2FitFunc_dmmodelDM, [275](#)
 - t2UpdateFunc_dmmodelCM, [275](#)
 - t2UpdateFunc_dmmodelDM, [275](#)
- t2fit_fitwaves.C, [275](#)
 - t2FitFunc_fitwaves, [275](#)
 - t2UpdateFunc_fitwaves, [275](#)
- t2fit_fitwaves.h, [276](#)
 - t2FitFunc_fitwaves, [276](#)
 - t2UpdateFunc_fitwaves, [276](#)
- t2fit_glitch.C, [276](#)
 - t2FitFunc_stdGlitch, [277](#)
 - t2UpdateFunc_stdGlitch, [277](#)
- t2fit_glitch.h, [278](#)
 - t2FitFunc_stdGlitch, [278](#)
 - t2UpdateFunc_stdGlitch, [278](#)
- t2fit_ifunc.C, [278](#)
 - ifunc, [279](#)

- sinfunc, [279](#)
 - t2FitFunc_ifunc, [279](#)
 - t2FitFunc_sifunc, [279](#)
 - t2UpdateFunc_ifunc, [279](#)
- t2fit_ifunc.h, [280](#)
 - ifunc, [281](#)
 - sinfunc, [281](#)
 - t2FitFunc_ifunc, [281](#)
 - t2FitFunc_sifunc, [281](#)
 - t2UpdateFunc_ifunc, [281](#)
- t2fit_position.C, [281](#)
 - t2FitFunc_stdPosition, [281](#)
 - t2UpdateFunc_stdPosition, [281](#)
- t2fit_position.h, [281](#)
 - t2FitFunc_stdPosition, [282](#)
 - t2UpdateFunc_stdPosition, [282](#)
- t2fit_stdFitFuncs.C, [283](#)
 - t2FitFunc_binaryModels, [284](#)
 - t2FitFunc_jump, [284](#)
 - t2FitFunc_miscDm, [284](#)
 - t2FitFunc_planet, [284](#)
 - t2FitFunc_stdDm, [284](#)
 - t2FitFunc_stdFreq, [284](#)
 - t2FitFunc_stdGravWav, [284](#)
 - t2FitFunc_telPos, [284](#)
 - t2FitFunc_zero, [284](#)
 - t2UpdateFunc_binaryModels, [284](#)
 - t2UpdateFunc_jump, [284](#)
 - t2UpdateFunc_miscDm, [284](#)
 - t2UpdateFunc_planet, [284](#)
 - t2UpdateFunc_simpleAdd, [284](#)
 - t2UpdateFunc_simpleMinus, [284](#)
 - t2UpdateFunc_stdFreq, [284](#)
 - t2UpdateFunc_stdGravWav, [284](#)
 - t2UpdateFunc_telPos, [284](#)
 - t2UpdateFunc_zero, [284](#)
- t2fit_stdFitFuncs.h, [284](#)
 - t2FitFunc_binaryModels, [286](#)
 - t2FitFunc_ifunc, [286](#)
 - t2FitFunc_jump, [286](#)
 - t2FitFunc_miscDm, [286](#)
 - t2FitFunc_planet, [286](#)
 - t2FitFunc_stdDm, [286](#)
 - t2FitFunc_stdFreq, [286](#)
 - t2FitFunc_stdGravWav, [286](#)
 - t2FitFunc_telPos, [286](#)
 - t2FitFunc_zero, [286](#)
 - t2UpdateFunc_binaryModels, [286](#)
 - t2UpdateFunc_ifunc, [286](#)
 - t2UpdateFunc_jump, [286](#)
 - t2UpdateFunc_miscDm, [286](#)
 - t2UpdateFunc_planet, [286](#)
 - t2UpdateFunc_simpleAdd, [286](#)
 - t2UpdateFunc_simpleMinus, [286](#)
 - t2UpdateFunc_stdFreq, [286](#)
 - t2UpdateFunc_stdGravWav, [286](#)
 - t2UpdateFunc_telPos, [286](#)
 - t2UpdateFunc_zero, [286](#)
- T2fitSpectra
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- T2fitSpectraRMS
 - planet_plug.C, [224](#)
- T2get_covFunc_automatic
 - choleskyAutomatic.C, [92](#)
 - choleskyRoutines.h, [96](#)
- T2getHighFreqRes
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- T2getWhiteNoiseLevel
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- T2getWhiteRes
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- T2globalEfac
 - pulsar, [69](#)
- T2guess_vals
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [96](#)
- T2interpolate
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [97](#)
- T2model
 - T2model.C, [288](#)
 - tempo2.h, [321](#)
- T2model.C, [287](#)
 - addKeplerianJumps, [288](#)
 - calcGR, [288](#)
 - computeU, [288](#)
 - deriveKeplerian, [288](#)
 - derivePostKeplerian, [288](#)
 - getKeplerian, [288](#)
 - getParameter, [288](#)
 - getPostKeplerian, [288](#)
 - KopeikinTerms, [288](#)
 - T2model, [288](#)
 - updateParameters, [288](#)
 - updateT2, [288](#)
- T2obtainTimingResiduals
 - choleskyRoutines.C, [94](#)
 - choleskyRoutines.h, [97](#)
- T2toolkit.C, [288](#)
 - genrand_int32, [290](#)
 - genrand_real1, [290](#)
 - init_genrand, [290](#)
 - RAND_M, [290](#)
 - RAND_N, [290](#)
 - TKconvertFloat1, [290](#)
 - TKconvertFloat2, [290](#)
 - TKfindMax_d, [290](#)
 - TKfindMax_f, [290](#)
 - TKfindMedian_d, [290](#)
 - TKfindMedian_f, [290](#)
 - TKfindMin_d, [290](#)
 - TKfindMin_f, [290](#)

- TKfindRMS_d, 290
- TKfindRMS_f, 290
- TKfindRMSweight_d, 290
- TKgaussDev, 290
- TKmean_d, 290
- TKmean_f, 290
- TKranDev, 290
- TKrange_d, 290
- TKrange_f, 290
- TKretMax_d, 290
- TKretMax_f, 290
- TKretMin_d, 290
- TKretMin_f, 290
- TKretMin_i, 290
- TKsetSeed, 291
- TKsign_d, 291
- TKsort_2f, 291
- TKsort_3d, 291
- TKsort_d, 291
- TKsort_f, 291
- TKvariance_d, 291
- TKzeromean_d, 291
- T2toolkit.h, 291
 - genrand_int32, 292
 - genrand_real1, 292
 - init_genrand, 292
 - TKconvertFloat1, 292
 - TKconvertFloat2, 292
 - TKfindMax_d, 292
 - TKfindMax_f, 292
 - TKfindMedian_d, 292
 - TKfindMedian_f, 292
 - TKfindMin_d, 292
 - TKfindMin_f, 292
 - TKfindRMS_d, 292
 - TKfindRMS_f, 292
 - TKfindRMSweight_d, 292
 - TKgaussDev, 292
 - TKmean_d, 292
 - TKmean_f, 292
 - TKranDev, 292
 - TKrange_d, 292
 - TKrange_f, 293
 - TKretMax_d, 293
 - TKretMax_f, 293
 - TKretMin_d, 293
 - TKretMin_f, 293
 - TKretMin_i, 293
 - TKsetSeed, 293
 - TKsign_d, 293
 - TKsort_2f, 293
 - TKsort_3d, 293
 - TKsort_d, 293
 - TKsort_f, 293
 - TKvariance_d, 293
 - TKzeromean_d, 293
- t2toolkit_global.C, 293
 - MAX_OBSN, 294
- T2writeCovarFuncModel
 - choleskyRoutines.C, 94
 - choleskyRoutines.h, 97
- TDB_UNITS
 - tempo2.h, 312
- TDBTDT_FILE
 - tempo2.h, 312
- TEMPO2_ARCH
 - config.h, 101
- TEMPO2_ENVIRON
 - global.C, 127
 - tempo2.h, 323
- TEMPO2_ERROR
 - global.C, 127
 - tempo2.h, 323
- TEMPO2_h_HASH
 - tempo2.h, 312
- TEMPO2_h_MAJOR_VER
 - tempo2.h, 312
- TEMPO2_h_MINOR_VER
 - tempo2.h, 312
- TEMPO2_h_VER
 - tempo2.h, 312
- TINY
 - GWevolve_plug.C, 204
 - interpolate_plug.C, 217
- TK_MAX_ERROR_LEN
 - TKlog.h, 343
- TK_MAX_ERRORS
 - TKlog.h, 343
- TK_STORE_ERROR
 - TKlog.h, 343
- TK_STORE_WARNING
 - TKlog.h, 343
- TK_dft
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- TK_errorCount
 - TKlog.C, 340
 - TKlog.h, 343
- TK_errorlog
 - TKlog.C, 340
 - TKlog.h, 343
- TK_fft
 - TKspectrum.C, 354
 - TKspectrum.h, 359
- TK_fitSine
 - TKspectrum.h, 360
- TK_fitSinusoids
 - TKspectrum.C, 354
 - TKspectrum.h, 360
- TK_warnCount
 - TKlog.C, 341
 - TKlog.h, 343
- TK_warnlog
 - TKlog.C, 341
 - TKlog.h, 343
- TK_weightLS

- TKspectrum.C, [354](#)
- TKspectrum.h, [360](#)
- TKaveragePts
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)
- TKbacksubstitution_svd
 - TKsvd.C, [361](#)
 - TKsvd.h, [362](#)
- TKbacksubstitution_svd2
 - globalDCM_fitFunc_plug.C, [193](#)
- TKbidiagonal
 - TKsvd.C, [361](#)
 - TKsvd.h, [362](#)
- TKbidiagonal2
 - globalDCM_fitFunc_plug.C, [193](#)
- TKboxcar
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)
- TKcalcSigmaz
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)
- TKcholesky.h, [334](#)
 - cholesky_covarFunc2matrix, [335](#)
 - cholesky_dmModel, [335](#)
 - cholesky_dmModelCovarParam, [335](#)
 - cholesky_ecm, [335](#)
 - cholesky_formUinv, [335](#)
 - cholesky_powerlawModel, [335](#)
 - cholesky_powerlawModel_withBeta, [335](#)
 - cholesky_readFromCovarianceFunction, [335](#)
- TKcmonot
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)
- TKconstrainedLeastSquares
 - TKfit.C, [337](#)
 - TKfit.h, [339](#)
- TKconvertFloat1
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKconvertFloat2
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfindMax_Ld
 - TKfit.C, [337](#)
- TKfindMax_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfindMax_f
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfindMedian_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfindMedian_f
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfindMin_d
 - T2toolkit.C, [290](#)
- T2toolkit.h, [292](#)
- TKfindMin_f
 - T2toolkit.C, [290](#)
- TKfindPoly_d
 - TKfit.C, [337](#)
 - TKfit.h, [339](#)
- TKfindRMS_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfindRMS_f
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfindRMSweight_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKfirstDifference
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)
- TKfit.C, [335](#)
 - TKconstrainedLeastSquares, [337](#)
 - TKfindMax_Ld, [337](#)
 - TKfindPoly_d, [337](#)
 - TKfit_getPulsarDesignMatrix, [337](#)
 - TKfitPoly, [337](#)
 - TKleastSquares, [337](#)
 - TKleastSquares_global_pulsar, [337](#)
 - TKleastSquares_single_pulsar, [337](#)
 - TKleastSquares_svd, [337](#)
 - TKleastSquares_svd_noErr, [337](#)
 - TKleastSquares_svd_passN, [337](#)
 - TKleastSquares_svd_psr, [337](#)
 - TKleastSquares_svd_psr_dcm, [337](#)
 - TKremovePoly_d, [337](#)
 - TKremovePoly_f, [337](#)
 - TKrobustConstrainedLeastSquares, [337](#)
 - TKrobustLeastSquares, [338](#)
- TKfit.h, [338](#)
 - TKconstrainedLeastSquares, [339](#)
 - TKfindPoly_d, [339](#)
 - TKfitPoly, [339](#)
 - TKleastSquares, [339](#)
 - TKleastSquares_svd, [339](#)
 - TKleastSquares_svd_noErr, [339](#)
 - TKremovePoly_d, [339](#)
 - TKremovePoly_f, [339](#)
 - TKrobustConstrainedLeastSquares, [339](#)
 - TKrobustLeastSquares, [339](#)
- TKfit_getPulsarDesignMatrix
 - TKfit.C, [337](#)
- TKfitPoly
 - TKfit.C, [337](#)
 - TKfit.h, [339](#)
- TKgaussDev
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKhann
 - TKspectrum.C, [355](#)

- TKspectrum.h, 360
- TKinterpolateSplineSmoothFixedXPts
 - TKspectrum.C, 355
 - TKspectrum.h, 360
- TKleastSquares
 - TKfit.C, 337
 - TKfit.h, 339
- TKleastSquares_global_pulsar
 - TKfit.C, 337
- TKleastSquares_single_pulsar
 - TKfit.C, 337
- TKleastSquares_svd
 - TKfit.C, 337
 - TKfit.h, 339
- TKleastSquares_svd_noErr
 - TKfit.C, 337
 - TKfit.h, 339
- TKleastSquares_svd_passN
 - TKfit.C, 337
- TKleastSquares_svd_psr
 - TKfit.C, 337
- TKleastSquares_svd_psr_dcm
 - TKfit.C, 337
- TKlog.C, 339
 - _TKchklog, 340
 - debugFlag, 340
 - logerr_check, 340
 - TK_errorCount, 340
 - TK_errorlog, 340
 - TK_warnCount, 341
 - TK_warnlog, 341
 - tcheck, 340
 - timer_clk, 340
 - writeResiduals, 341
- TKlog.h, 341
 - _LOG, 342
 - _TKchklog, 343
 - BOLDCOLOR, 342
 - DEPRECATED, 342
 - debugFlag, 343
 - ENDERR, 342
 - ENDL, 342
 - ERRORCOLOR, 342
 - LOG_OUTFILE, 342
 - logdbg, 342
 - logerr, 342
 - logerr_check, 343
 - logmsg, 342
 - logtchk, 342
 - logwarn, 342
 - RESETCOLOR, 343
 - TK_MAX_ERROR_LEN, 343
 - TK_MAX_ERRORS, 343
 - TK_STORE_ERROR, 343
 - TK_STORE_WARNING, 343
 - TK_errorCount, 343
 - TK_errorlog, 343
 - TK_warnCount, 343
 - TK_warnlog, 343
 - tcheck, 343
 - timer_clk, 343
 - WARNCOLOR, 343
 - WHEREARG, 343
 - WHEREERR, 343
 - WHERESTR, 343
 - WHERECHK, 343
 - WHEREWARN, 343
 - writeResiduals, 343
- TKlomb_d
 - TKspectrum.C, 355
 - TKspectrum.h, 360
- TKlongdouble.C, 343
 - BUFSIZE, 344
 - ld_fprintf, 344
 - ld_printf, 344
 - ld_sprintf, 344
 - ld_vsprintf, 344
 - parse_longdouble, 344
 - print_longdouble, 344
- TKlongdouble.float128.h, 345
 - cosl, 345
 - FMT_LD, 345
 - fabsl, 345
 - floorl, 345
 - LD_PI, 346
 - LONGDOUBLE_IS_FLOAT128, 346
 - LONGDOUBLE_ONE, 346
 - ld_fprintf, 346
 - ld_printf, 346
 - ld_sprintf, 346
 - longdouble, 346
 - parse_longdouble, 346
 - sinl, 346
 - USE_BUILTIN_LONGDOUBLE, 346
- TKlongdouble.h, 346
 - cosl, 347
 - FMT_LD, 347
 - fabsl, 347
 - floorl, 347
 - LD_PI, 347
 - LONGDOUBLE_IS_FLOAT128, 347
 - LONGDOUBLE_ONE, 347
 - ld_fprintf, 347
 - ld_printf, 347
 - ld_sprintf, 348
 - longdouble, 347
 - parse_longdouble, 348
 - sinl, 347
 - USE_BUILTIN_LONGDOUBLE, 347
- TKlongdouble.ld.h, 348
 - LD_PI, 348
 - LONGDOUBLE_IS_IEEE754, 349
 - LONGDOUBLE_ONE, 349
 - ld_fprintf, 348
 - ld_printf, 348
 - ld_sprintf, 348

- longdouble, [349](#)
- parse_longdouble, [349](#)
- USE_BUILTIN_LONGDOUBLE, [349](#)
- TKmatrix.C, [349](#)
 - free_2dLL, [350](#)
 - free_2df, [350](#)
 - free_blas, [350](#)
 - free_uinv, [350](#)
 - get_blas_cols, [350](#)
 - get_blas_rows, [350](#)
 - malloc_2dLL, [350](#)
 - malloc_2df, [350](#)
 - malloc_blas, [350](#)
 - malloc_uinv, [350](#)
 - TKmultMatrix, [350](#)
 - TKmultMatrix_sq, [350](#)
 - TKmultMatrixVec, [350](#)
 - TKmultMatrixVec_sq, [350](#)
- TKmatrix.h, [350](#)
 - free_2df, [351](#)
 - free_blas, [351](#)
 - free_uinv, [351](#)
 - get_blas_cols, [351](#)
 - get_blas_rows, [351](#)
 - malloc_2df, [351](#)
 - malloc_blas, [351](#)
 - malloc_uinv, [351](#)
 - TKmultMatrix, [351](#)
 - TKmultMatrix_sq, [351](#)
 - TKmultMatrixVec, [351](#)
 - TKmultMatrixVec_sq, [351](#)
- TKmean_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKmean_f
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKmultMatrix
 - TKmatrix.C, [350](#)
 - TKmatrix.h, [351](#)
- TKmultMatrix_sq
 - TKmatrix.C, [350](#)
 - TKmatrix.h, [351](#)
- TKmultMatrixVec
 - TKmatrix.C, [350](#)
 - TKmatrix.h, [351](#)
- TKmultMatrixVec_sq
 - TKmatrix.C, [350](#)
 - TKmatrix.h, [351](#)
- TKpythag
 - TKsvd.C, [361](#)
 - TKsvd.h, [362](#)
- TKpythag2
 - globalDCM_fitFunc_plug.C, [193](#)
- TKranDev
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKrange_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [292](#)
- TKrange_f
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [293](#)
- TKremovePoly_d
 - TKfit.C, [337](#)
 - TKfit.h, [339](#)
- TKremovePoly_f
 - TKfit.C, [337](#)
 - TKfit.h, [339](#)
- TKretMax_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [293](#)
- TKretMax_f
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [293](#)
- TKretMin_d
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [293](#)
- TKretMin_f
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [293](#)
- TKretMin_i
 - T2toolkit.C, [290](#)
 - T2toolkit.h, [293](#)
- TKrobustConstrainedLeastSquares
 - TKfit.C, [337](#)
 - TKfit.h, [339](#)
- TKrobustLeastSquares
 - TKfit.C, [338](#)
 - TKfit.h, [339](#)
- TKsetSeed
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TKsign_d
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TKsingularValueDecomposition_lsq
 - TKsvd.C, [361](#)
 - TKsvd.h, [362](#)
- TKsingularValueDecomposition_lsq2
 - globalDCM_fitFunc_plug.C, [193](#)
- TKsort_2f
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TKsort_3d
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TKsort_d
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TKsort_f
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TKsortit
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)

- TKspectrum
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)
- TKspectrum.C, [351](#)
 - ABS, [354](#)
 - addvar, [355](#)
 - calcSpectra, [354](#)
 - calcSpectra_ri, [354](#)
 - calcSpectra_ri_T, [354](#)
 - calcSpectraErr, [354](#)
 - data, [355](#)
 - fit4, [354](#)
 - fitCosSineFunc, [354](#)
 - fitMeanSineFunc, [354](#)
 - fitMeanSineFunc_IFUNC, [354](#)
 - GLOBAL_OMEGA, [355](#)
 - getprtj, [354](#)
 - getweights, [354](#)
 - globalOmega, [355](#)
 - indexx8, [354](#)
 - indx, [355](#)
 - linfile, [355](#)
 - MAX, [354](#)
 - MIN, [354](#)
 - mat20, [354](#)
 - nbintype, [355](#)
 - ncubic, [355](#)
 - ncubics, [355](#)
 - ndim, [355](#)
 - nformat, [356](#)
 - npt, [356](#)
 - npt1last, [356](#)
 - npt2last, [356](#)
 - ntau, [356](#)
 - ntunits, [356](#)
 - nusewt, [356](#)
 - nwriteres, [356](#)
 - nxunits, [356](#)
 - permax, [356](#)
 - prtl, [356](#)
 - readin, [354](#)
 - root2, [356](#)
 - secyear, [356](#)
 - sigmai, [356](#)
 - sineFunc, [354](#)
 - TK_dft, [354](#)
 - TK_fft, [354](#)
 - TK_fitSinusoids, [354](#)
 - TK_weightLS, [354](#)
 - TKaveragePts, [355](#)
 - TKboxcar, [355](#)
 - TKcalcSigmaz, [355](#)
 - TKcmonot, [355](#)
 - TKfirstDifference, [355](#)
 - TKhann, [355](#)
 - TKinterpolateSplineSmoothFixedXPts, [355](#)
 - TKlomb_d, [355](#)
 - TKsortit, [355](#)
 - TKspectrum, [355](#)
 - TKspline_interpolate, [355](#)
 - tauday, [356](#)
 - tauensure, [356](#)
 - taulog, [356](#)
 - taumax, [356](#)
 - taumin, [356](#)
 - tausec, [356](#)
 - tauyear, [356](#)
 - tdiffmin, [356](#)
 - tmax, [356](#)
 - tmin, [356](#)
 - utfirst, [356](#)
 - utjd, [356](#)
 - utjd1, [356](#)
 - utjd2, [356](#)
 - utjdlast, [357](#)
 - utlast, [357](#)
 - utmean, [357](#)
 - verbose_calc_spectra, [357](#)
 - xmax, [357](#)
 - xmin, [357](#)
- TKspectrum.h, [357](#)
 - ABS, [359](#)
 - calcSpectra, [359](#)
 - calcSpectra_ri, [359](#)
 - calcSpectra_ri_T, [359](#)
 - calcSpectraErr, [359](#)
 - complexVal, [359](#)
 - fit4, [359](#)
 - fitCosSineFunc, [359](#)
 - fitMeanSineFunc, [359](#)
 - fitMeanSineFunc_IFUNC, [359](#)
 - GLOBAL_OMEGA, [360](#)
 - getprtj, [359](#)
 - getweights, [359](#)
 - indexx8, [359](#)
 - MAX, [359](#)
 - MIN, [359](#)
 - mat20, [359](#)
 - readin, [359](#)
 - sineFunc, [359](#)
 - TK_dft, [359](#)
 - TK_fft, [359](#)
 - TK_fitSine, [360](#)
 - TK_fitSinusoids, [360](#)
 - TK_weightLS, [360](#)
 - TKaveragePts, [360](#)
 - TKboxcar, [360](#)
 - TKcalcSigmaz, [360](#)
 - TKcmonot, [360](#)
 - TKfirstDifference, [360](#)
 - TKhann, [360](#)
 - TKinterpolateSplineSmoothFixedXPts, [360](#)
 - TKlomb_d, [360](#)
 - TKsortit, [360](#)
 - TKspectrum, [360](#)
 - TKspline_interpolate, [360](#)

- verbose_calc_spectra, [360](#)
- TKspline_interpolate
 - TKspectrum.C, [355](#)
 - TKspectrum.h, [360](#)
- TKsvd.C, [360](#)
 - TKbacksubstitution_svd, [361](#)
 - TKbidiagonal, [361](#)
 - TKpythag, [361](#)
 - TKsingularValueDecomposition_Isq, [361](#)
- TKsvd.h, [362](#)
 - TKbacksubstitution_svd, [362](#)
 - TKbidiagonal, [362](#)
 - TKpythag, [362](#)
 - TKsingularValueDecomposition_Isq, [362](#)
- TKvariance_d
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TKzeromean_d
 - T2toolkit.C, [291](#)
 - T2toolkit.h, [293](#)
- TNBandDMAmp
 - pulsar, [70](#)
- TNBandDMC
 - pulsar, [70](#)
- TNBandDMGam
 - pulsar, [70](#)
- TNBandNoiseAmp
 - pulsar, [70](#)
- TNBandNoiseC
 - pulsar, [70](#)
- TNBandNoiseGam
 - pulsar, [70](#)
- TNBandNoiseHF
 - pulsar, [70](#)
- TNBandNoiseLF
 - pulsar, [70](#)
- TNDMAmp
 - pulsar, [70](#)
- TNDMC
 - pulsar, [70](#)
- TNDMCoeffs
 - pulsar, [70](#)
- TNDMErr
 - observation, [49](#)
- TNDMEvAmp
 - pulsar, [70](#)
- TNDMEvGam
 - pulsar, [70](#)
- TNDMEvLength
 - pulsar, [70](#)
- TNDMEvLin
 - pulsar, [70](#)
- TNDMEvOff
 - pulsar, [70](#)
- TNDMEvQuad
 - pulsar, [70](#)
- TNDMEvStart
 - pulsar, [70](#)
- TNDMGam
 - pulsar, [70](#)
- TNDMSignal
 - observation, [49](#)
- TNECORRFlagID
 - pulsar, [70](#)
- TNECORRFlagVal
 - pulsar, [70](#)
- TNECORRVal
 - pulsar, [70](#)
- TNEFFlagID
 - pulsar, [70](#)
- TNEFFlagVal
 - pulsar, [70](#)
- TNEFVal
 - pulsar, [70](#)
- TNEQFlagID
 - pulsar, [71](#)
- TNEQFlagVal
 - pulsar, [71](#)
- TNEQVal
 - pulsar, [71](#)
- TNGlobalEF
 - pulsar, [71](#)
- TNGlobalEQ
 - pulsar, [71](#)
- TNGroupErr
 - observation, [49](#)
- TNGroupNoiseAmp
 - pulsar, [71](#)
- TNGroupNoiseC
 - pulsar, [71](#)
- TNGroupNoiseFlagID
 - pulsar, [71](#)
- TNGroupNoiseFlagVal
 - pulsar, [71](#)
- TNGroupNoiseGam
 - pulsar, [71](#)
- TNGroupSignal
 - observation, [49](#)
- TNRedAmp
 - pulsar, [71](#)
- TNRedC
 - pulsar, [71](#)
- TNRedCoeffs
 - pulsar, [71](#)
- TNRedCorner
 - pulsar, [71](#)
- TNRedErr
 - observation, [49](#)
- TNRedFlow
 - pulsar, [71](#)
- TNRedGam
 - pulsar, [71](#)
- TNRedSignal
 - observation, [49](#)
- TNSQFlagID
 - pulsar, [71](#)

- TNSQFlagVal
 - pulsar, [71](#)
- TNSQVal
 - pulsar, [71](#)
- TNShapeletEvFScale
 - pulsar, [71](#)
- TNShapeletEvN
 - pulsar, [71](#)
- TNShapeletEvPos
 - pulsar, [71](#)
- TNShapeletEvWidth
 - pulsar, [71](#)
- TNsubtractDM
 - pulsar, [71](#)
- TNsubtractRed
 - pulsar, [71](#)
- tOffset
 - pulsar, [71](#)
- tOffset_f1
 - pulsar, [72](#)
- tOffset_f2
 - pulsar, [72](#)
- tOffset_t1
 - pulsar, [72](#)
- tOffset_t2
 - pulsar, [72](#)
- tOffsetFlags
 - pulsar, [72](#)
- tOffsetSite
 - pulsar, [72](#)
- TRUE
 - jpleph.c, [139](#)
- TSUN
 - tempo2.h, [312](#)
- table
 - ClockCorrectionFunction, [29](#)
 - MeteorologyFunction, [42](#)
- TabulatedFunction, [77](#)
 - fileName, [77](#)
 - header_line, [77](#)
 - samples, [77](#)
- TabulatedFunction_getEndX
 - tabulatedfunction.C, [295](#)
 - tabulatedfunction.h, [297](#)
- TabulatedFunction_getStartX
 - tabulatedfunction.C, [295](#)
 - tabulatedfunction.h, [297](#)
- TabulatedFunction_getValue
 - tabulatedfunction.C, [295](#)
 - tabulatedfunction.h, [297](#)
- TabulatedFunction_load
 - tabulatedfunction.C, [295](#)
 - tabulatedfunction.h, [297](#)
- TabulatedFunctionSample, [77](#)
 - x, [77](#)
 - y, [77](#)
- tabulatedfunction.C, [294](#)
 - TabulatedFunction_getEndX, [295](#)
- TabulatedFunction_getStartX, [295](#)
- TabulatedFunction_getValue, [295](#)
- TabulatedFunction_load, [295](#)
- tabulatedfunction.h, [296](#)
 - TabulatedFunction_getEndX, [297](#)
 - TabulatedFunction_getStartX, [297](#)
 - TabulatedFunction_getValue, [297](#)
 - TabulatedFunction_load, [297](#)
- tai2tt
 - tempo2.h, [321](#)
- tai2tt.C, [297](#)
- tai2ut1
 - tai2ut1.C, [298](#)
 - tempo2.h, [321](#)
- tai2ut1.C, [297](#)
 - tai2ut1, [298](#)
 - ut1red, [298](#)
- tauday
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- tauensure
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- taulog
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- taumax
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- taumin
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- tausec
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- tauyear
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- tcb2tdb
 - photons_plug.C, [221](#)
- tcheck
 - TKlog.C, [340](#)
 - TKlog.h, [343](#)
- tdb2tcb
 - photons_plug.C, [221](#)
- tdiffmin
 - sigmaz_plug.C, [236](#)
 - TKspectrum.C, [356](#)
- tdis1
 - observation, [49](#)
- tdis2
 - observation, [49](#)
- telDX_e
 - pulsar, [69](#)
- telDX_t
 - pulsar, [69](#)
- telDX_v
 - pulsar, [69](#)

- telDX_vel
 - pulsar, [69](#)
- telDX_vel_e
 - pulsar, [69](#)
- telDY_e
 - pulsar, [69](#)
- telDY_t
 - pulsar, [69](#)
- telDY_v
 - pulsar, [69](#)
- telDY_vel
 - pulsar, [69](#)
- telDY_vel_e
 - pulsar, [69](#)
- telDZ_e
 - pulsar, [69](#)
- telDZ_t
 - pulsar, [69](#)
- telDZ_v
 - pulsar, [69](#)
- telDZ_vel
 - pulsar, [69](#)
- telDZ_vel_e
 - pulsar, [69](#)
- telID
 - observation, [49](#)
- tempo1
 - pulsar, [69](#)
- tempo2.C, [298](#)
 - clock_corrections, [299](#)
 - ephemeris_routines, [299](#)
 - extra_delays, [299](#)
 - main, [299](#)
 - thwart_annoying_dynamic_library_stuff, [299](#)
- tempo2.h, [299](#)
 - AU_DIST, [307](#)
 - AULTSC, [307](#)
 - allocateMemory, [318](#)
 - autoConstraints, [318](#)
 - BIG_G, [307](#)
 - BTJmodel, [318](#)
 - BTXmodel, [318](#)
 - BTmodel, [318](#)
 - bootstrap, [318](#)
 - CVSdisplayVersion, [318](#)
 - calcRMS, [318](#)
 - calculate_bclt, [318](#)
 - compute_tropospheric_delays, [318](#)
 - constraint, [314](#)
 - constraint_LAST, [315](#)
 - constraint_dmmodel_cw_0, [314](#)
 - constraint_dmmodel_cw_1, [314](#)
 - constraint_dmmodel_cw_2, [314](#)
 - constraint_dmmodel_cw_3, [314](#)
 - constraint_dmmodel_cw_px, [314](#)
 - constraint_dmmodel_cw_year_cos, [314](#)
 - constraint_dmmodel_cw_year_cos2, [314](#)
 - constraint_dmmodel_cw_year_sin, [314](#)
 - constraint_dmmodel_cw_year_sin2, [314](#)
 - constraint_dmmodel_cw_year_xcos, [314](#)
 - constraint_dmmodel_cw_year_xsin, [314](#)
 - constraint_dmmodel_dm1, [314](#)
 - constraint_dmmodel_mean, [314](#)
 - constraint_ifunc_0, [314](#)
 - constraint_ifunc_1, [314](#)
 - constraint_ifunc_2, [314](#)
 - constraint_ifunc_year_cos, [314](#)
 - constraint_ifunc_year_cos2, [314](#)
 - constraint_ifunc_year_sin, [314](#)
 - constraint_ifunc_year_sin2, [314](#)
 - constraint_ifunc_year_xcos, [314](#)
 - constraint_ifunc_year_xsin, [314](#)
 - constraint_label, [313](#)
 - constraint_qifunc_c_year_cos, [315](#)
 - constraint_qifunc_c_year_cos2, [315](#)
 - constraint_qifunc_c_year_sin, [315](#)
 - constraint_qifunc_c_year_sin2, [315](#)
 - constraint_qifunc_c_year_xcos, [315](#)
 - constraint_qifunc_c_year_xsin, [315](#)
 - constraint_qifunc_p_year_cos, [315](#)
 - constraint_qifunc_p_year_cos2, [315](#)
 - constraint_qifunc_p_year_sin, [314](#)
 - constraint_qifunc_p_year_sin2, [315](#)
 - constraint_qifunc_p_year_xcos, [315](#)
 - constraint_qifunc_p_year_xsin, [315](#)
 - constraint_quad_ifunc_c_0, [314](#)
 - constraint_quad_ifunc_c_1, [314](#)
 - constraint_quad_ifunc_c_2, [314](#)
 - constraint_quad_ifunc_p_0, [314](#)
 - constraint_quad_ifunc_p_1, [314](#)
 - constraint_quad_ifunc_p_2, [314](#)
 - constraint_tel_dx_0, [314](#)
 - constraint_tel_dx_1, [314](#)
 - constraint_tel_dx_2, [314](#)
 - constraint_tel_dy_0, [314](#)
 - constraint_tel_dy_1, [314](#)
 - constraint_tel_dy_2, [314](#)
 - constraint_tel_dz_0, [314](#)
 - constraint_tel_dz_1, [314](#)
 - constraint_tel_dz_2, [314](#)
 - constraintDerivFunc, [313](#)
 - copyPSR, [318](#)
 - copyParam, [318](#)
 - covarFuncFile, [322](#)
 - DDGRmodel, [318](#)
 - DDHmodel, [318](#)
 - DDKmodel, [318](#)
 - DDSmodel, [318](#)
 - DDmodel, [318](#)
 - DM_CONST, [307](#)
 - DM_CONST_SI, [307](#)
 - dcmFile, [322](#)
 - defineClockCorrectionSequence, [318](#)
 - destroyMemory, [318](#)
 - destroyOne, [318](#)
 - displayCVSversion, [322](#)

displayMsg, 318
 displayParameters, 318
 dm_delays, 318
 dms_turn, 319
 doFit, 319
 doFitAll, 319
 doFitDCM, 319
 doFitGlobal, 319
 dotproduct, 319
 ECLIPTIC_OBLIQUITY, 323
 ECLIPTIC_OBLIQUITY_VAL, 307
 ELL1Hmodel, 319
 ELL1model, 319
 equ2ecl, 319
 FB90_TIMEEPH, 307
 FITfuncs, 319
 FitInfo, 313
 forceGlobalFit, 323
 formBats, 319
 formBatsAll, 319
 formResiduals, 319
 fortran_mod, 319
 fortran_nint, 319
 fortran_nlong, 319
 GM, 307
 GM_C3, 307
 GMJ_C3, 307
 GMN_C3, 307
 GMS_C3, 307
 GMU_C3, 307
 GMV_C3, 308
 get_EOP, 319
 get_OneobsCoord, 319
 get_obsCoord, 319
 get_obsCoord_IAU2000B, 319
 getCholeskyMatrix, 319
 getClockCorrections, 319
 getCorrection, 319
 getCorrectionTT, 319
 getInputs, 319
 getObservatory, 320
 getParamDeriv, 320
 getParameterValue, 320
 HAVE_GWSIM_H, 308
 hms_turn, 320
 IF99_TIMEEPH, 308
 IFTEPH_FILE, 308
 id_residual, 320
 initialise, 320
 initialiseOne, 320
 JVmodel, 320
 LEAPSECOND_FILE, 308
 label, 315
 logicFlag, 320
 lookup_observatory_alias, 320
 MASR2RADS, 308
 MAX_BPJ_JUMPS, 308
 MAX_CLK_CORR, 308
 MAX_CLKCORR, 308
 MAX_COEFF, 308
 MAX_COMPANIONS, 308
 MAX_DM_DERIVATIVES, 308
 MAX_DMX, 308
 MAX_FILELEN, 309
 MAX_FIT, 309
 MAX_FLAG_LEN, 309
 MAX_FLAGS, 309
 MAX_FREQ_DERIVATIVES, 309
 MAX_IFUNC, 309
 MAX_JUMPS, 309
 MAX_LEAPSEC, 309
 MAX_MSG, 309
 MAX_OBSN, 323
 MAX_OBSN_VAL, 309
 MAX_PARAMS, 309
 MAX_PSR, 323
 MAX_PSR_VAL, 309
 MAX_QUAD, 310
 MAX_SITE, 310
 MAX_STOREPRECISION, 310
 MAX_STRLEN, 310
 MAX_T2EFAC, 310
 MAX_T2EQUAD, 310
 MAX_TEL_CLK_OFFS, 310
 MAX_TEL_DX, 310
 MAX_TEL_DY, 310
 MAX_TEL_DZ, 310
 MAX_TNBN, 310
 MAX_TNDMEv, 310
 MAX_TNECORR, 310
 MAX_TNEF, 311
 MAX_TNEQ, 311
 MAX_TNGN, 311
 MAX_TNSQ, 311
 MAX_TOFFSET, 311
 MAX_WHITE, 311
 MSSmodel, 320
 NE_SW_DEFAULT, 311
 NEWFIT, 323
 OBLQ, 311
 OBSSYS_FILE, 311
 observation, 313
 PCM, 311
 param_JUMP, 318
 param_LAST, 318
 param_ZERO, 318
 param_a0, 316
 param_a1, 315
 param_a1dot, 316
 param_a2dot, 316
 param_afac, 317
 param_b0, 316
 param_bp, 316
 param_bpja1, 316
 param_bpjec, 316
 param_bpjep, 316

param_bpjom, 316
param_bpjpb, 316
param_bpjph, 316
param_bpp, 316
param_brake, 318
param_cgw, 317
param_clk_offs, 317
param_daop, 317
param_decj, 315
param_df1, 318
param_dm, 315
param_dm_cos1yr, 317
param_dm_sin1yr, 317
param_dmassplanet, 317
param_dmepoch, 315
param_dmmodel, 317
param_dmx, 317
param_dmxr1, 317
param_dmxr2, 317
param_dr, 316
param_dshk, 317
param_dth, 316
param_dtheta, 316
param_e2dot, 315
param_ecc, 315
param_edot, 315
param_ephver, 317
param_eps1, 316
param_eps1dot, 317
param_eps2, 316
param_eps2dot, 317
param_f, 315
param_fb, 315
param_fd, 316
param_fddc, 316
param_fddi, 316
param_finish, 316
param_gamma, 316
param_glep, 316
param_glf0, 316
param_glf0d, 316
param_glf1, 316
param_glf2, 316
param_glph, 316
param_gltd, 316
param_gwb_amp, 317
param_gwecc, 317
param_gwm_amp, 317
param_gwsingle, 317
param_h3, 317
param_h4, 317
param_ifunc, 317
param_iperharm, 317
param_kin, 316
param_kom, 316
param_label, 313
param_m2, 316
param_mtot, 316
param_nharm, 317
param_om, 315
param_om2dot, 316
param_omdot, 316
param_orbpx, 316
param_pb, 315
param_pbdot, 315
param_pepoch, 315
param_pmdec, 315
param_pmra, 315
param_pmr, 315
param_posepoch, 315
param_px, 315
param_quad_ifunc_c, 317
param_quad_ifunc_p, 317
param_quad_om, 317
param_raj, 315
param_shapmax, 316
param_sini, 315
param_start, 316
param_stateSwitchT, 318
param_stig, 317
param_t0, 315
param_tasc, 316
param_tel_dx, 317
param_tel_dy, 317
param_tel_dz, 317
param_tel_vx, 317
param_tel_vy, 317
param_tel_vz, 317
param_tel_x0, 317
param_tel_y0, 317
param_tel_z0, 317
param_telEpoch, 317
param_telx, 317
param_tely, 317
param_telz, 317
param_track, 316
param_tres, 317
param_tspan, 316
param_tzfrfq, 316
param_tzrmjd, 316
param_wave_dm, 317
param_wave_om, 316
param_waveepoch, 317
param_waveepoch_dm, 317
param_xomdot, 316
param_xpbdot, 315
paramDerivFunc, 313
paramUpdateFunc, 313
parameter, 313
polyco, 320
preProcess, 320
preProcessSimple, 320
preProcessSimple1, 320
preProcessSimple2, 320
preProcessSimple3, 320
processFlag, 320

processSimultaneous, 320
 pulsar, 313
 readEphemeris, 320
 readEphemeris_calceph, 320
 readJBO_bat, 320
 readObsFile, 320
 readOneEphemeris, 321
 readParfile, 321
 readParfileGlobal, 321
 readSimpleParfile, 321
 readTimfile, 321
 recordPrecision, 321
 SECDAY, 311
 SECDAYI, 311
 SI_UNITS, 312
 SOLAR_MASS, 312
 SOLAR_RADIUS, 312
 SPEED_LIGHT, 312
 secularMotion, 321
 setPlugPath, 321
 setStart, 321
 setupParameterFileDefaults, 321
 shapiro_delay, 321
 simplePlot, 321
 solarWindModel, 321
 sortToAs, 321
 storePrecision, 313
 T2_PTAmodel, 321
 T2C_IAU2000B, 312
 T2C_TEMPO, 312
 T2model, 321
 TDB_UNITS, 312
 TDBTDT_FILE, 312
 TEMPO2_ENVIRON, 323
 TEMPO2_ERROR, 323
 TEMPO2_h_HASH, 312
 TEMPO2_h_MAJOR_VER, 312
 TEMPO2_h_MINOR_VER, 312
 TEMPO2_h_VER, 312
 TSUN, 312
 tai2tt, 321
 tai2ut1, 321
 tempo2_plug_path, 323
 tempo2_plug_path_len, 323
 tempo2MachineType, 323
 textOutput, 321
 toa2utc, 321
 transform_units, 321
 tt2tb, 321
 turn_deg, 321
 turn_dms, 321
 turn_hms, 322
 UT1_FILE, 312
 updateBT, 322
 updateBTJ, 322
 updateBTX, 322
 updateBatsAll, 322
 updateDD, 322
 updateDDGR, 322
 updateDDH, 322
 updateDDK, 322
 updateDDS, 322
 updateELL1, 322
 updateELL1H, 322
 updateJV, 322
 updateMSS, 322
 updateParameters, 322
 updateT2, 322
 updateT2_PTA, 322
 useSelectFile, 322
 utc2tai, 322
 vectorPulsar, 322
 vectorscale, 322
 vectorsum, 322
 veryFast, 323
 writeTim, 322
 zoom_graphics, 322
 tempo2_GetNumberOfParameters
 designmatrix_plug.C, 167
 tempo2_plug_path
 global.C, 127
 tempo2.h, 323
 tempo2_plug_path_len
 global.C, 127
 tempo2.h, 323
 tempo2_verbose
 tempo2pred.c, 325
 tempo2MachineType
 global.C, 127
 tempo2.h, 323
 tempo2Util.C, 331
 copyPSR, 332
 copyParam, 332
 displayMsg, 332
 dms_turn, 332
 dotproduct, 332
 equ2ecl, 332
 fortran_mod, 332
 fortran_nint, 332
 fortran_nlong, 332
 getParameterValue, 332
 hms_turn, 332
 turn_deg, 332
 turn_dms, 332
 turn_hms, 332
 vectorscale, 332
 vectorsum, 332
 tempo2Util.h, 333
 dms_turn, 333
 hms_turn, 333
 turn_deg, 333
 tempo2pred.c, 324
 T2Predictor_Copy, 325
 T2Predictor_Destroy, 325
 T2Predictor_FRead, 325
 T2Predictor_FWrite, 325

- T2Predictor_GetEndFreq, [325](#)
- T2Predictor_GetEndMJD, [325](#)
- T2Predictor_GetFrequency, [325](#)
- T2Predictor_GetPSRName, [325](#)
- T2Predictor_GetPhase, [325](#)
- T2Predictor_GetPlan, [325](#)
- T2Predictor_GetPlan_Ext, [325](#)
- T2Predictor_GetSiteName, [325](#)
- T2Predictor_GetStartFreq, [325](#)
- T2Predictor_GetStartMJD, [325](#)
- T2Predictor_Init, [325](#)
- T2Predictor_Insert, [325](#)
- T2Predictor_Keep, [325](#)
- T2Predictor_Kind, [325](#)
- T2Predictor_Read, [325](#)
- T2Predictor_Write, [325](#)
- tempo2_verbose, [325](#)
- tempo2pred.h, [326](#)
 - Cheby, [327](#)
 - ChebyModelSet_OutOfRange, [328](#)
 - NonePredType, [327](#)
 - T1, [327](#)
 - T2Predictor_Copy, [327](#)
 - T2Predictor_Destroy, [327](#)
 - T2Predictor_FRead, [327](#)
 - T2Predictor_FWrite, [327](#)
 - T2Predictor_GetEndFreq, [327](#)
 - T2Predictor_GetEndMJD, [327](#)
 - T2Predictor_GetFrequency, [327](#)
 - T2Predictor_GetPSRName, [328](#)
 - T2Predictor_GetPhase, [327](#)
 - T2Predictor_GetPlan, [327](#)
 - T2Predictor_GetPlan_Ext, [327](#)
 - T2Predictor_GetSiteName, [328](#)
 - T2Predictor_GetStartFreq, [328](#)
 - T2Predictor_GetStartMJD, [328](#)
 - T2Predictor_Init, [328](#)
 - T2Predictor_Insert, [328](#)
 - T2Predictor_Keep, [328](#)
 - T2Predictor_Kind, [328](#)
 - T2Predictor_Read, [328](#)
 - T2Predictor_Write, [328](#)
 - T2PredictorKind, [327](#)
- tempo2pred_int.h, [328](#)
 - Cheby2D_Construct, [330](#)
 - Cheby2D_Construct_x_Derivative, [330](#)
 - Cheby2D_Test, [330](#)
 - ChebyModel_Construct, [330](#)
 - ChebyModel_Copy, [330](#)
 - ChebyModel_Destroy, [330](#)
 - ChebyModel_GetFrequency, [330](#)
 - ChebyModel_GetPhase, [330](#)
 - ChebyModel_Init, [330](#)
 - ChebyModel_Read, [330](#)
 - ChebyModel_Test, [330](#)
 - ChebyModel_Write, [330](#)
 - ChebyModelSet_Construct, [330](#)
 - ChebyModelSet_Destroy, [330](#)
 - ChebyModelSet_GetFrequency, [330](#)
 - ChebyModelSet_GetNearest, [330](#)
 - ChebyModelSet_GetPhase, [330](#)
 - ChebyModelSet_Init, [330](#)
 - ChebyModelSet_Insert, [330](#)
 - ChebyModelSet_Keep, [330](#)
 - ChebyModelSet_Read, [330](#)
 - ChebyModelSet_Test, [330](#)
 - ChebyModelSet_Write, [330](#)
 - T1Polyco_GetFrequency, [330](#)
 - T1Polyco_GetPhase, [330](#)
 - T1Polyco_Read, [331](#)
 - T1Polyco_Write, [331](#)
 - T1PolycoSet_Destroy, [331](#)
 - T1PolycoSet_GetFrequency, [331](#)
 - T1PolycoSet_GetNearest, [331](#)
 - T1PolycoSet_GetPhase, [331](#)
 - T1PolycoSet_Read, [331](#)
 - T1PolycoSet_Write, [331](#)
- tempoOutput
 - add_pulseNumber_plug.C, [144](#)
 - general2_plug.C, [183](#)
 - general_plug.C, [184](#)
 - matrix_plug.C, [219](#)
 - publish_plug.C, [232](#)
- tensor_alpha
 - gwgenSpec, [35](#)
- tensor_amp
 - gwgenSpec, [35](#)
- testCheby2D
 - cheby2d.c, [87](#)
- testFunc
 - cheby2d.c, [87](#)
- textOutput
 - tempo2.h, [321](#)
 - textOutput.C, [334](#)
- textOutput.C, [333](#)
 - calcRMS, [334](#)
 - dglep, [334](#)
 - m2, [334](#)
 - printGlitch, [334](#)
 - textOutput, [334](#)
- theta_bin
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- theta_g
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- ThetaEderivs
 - GWevolve_plug.C, [204](#)
- thwart_annoying_dynamic_library_stuff
 - tempo2.C, [299](#)
- timFile
 - calcDMe_plug.C, [159](#)
- timeEphemeris
 - pulsar, [70](#)
- timer_clk
 - TKlog.C, [340](#)

- TKlog.h, 343
- title
 - calcDMe_plug.C, 159
 - IFTEphemeris, 38
- tmax
 - sigmaz_plug.C, 236
 - TKspectrum.C, 356
- tmin
 - sigmaz_plug.C, 237
 - TKspectrum.C, 356
- ToAextraCovar
 - pulsar, 71
- toa2utc
 - tempo2.h, 321
 - toa2utc.C, 363
- toa2utc.C, 362
 - convertTOA, 363
 - linearInterpolate, 363
 - toa2utc, 363
 - toa2utc_nist, 363
 - USE_NEW_CLK_CORR, 363
- toa2utc_nist
 - toa2utc.C, 363
- toaDMErr
 - observation, 49
- toaErr
 - observation, 49
- toffset
 - cholSpectra_plug.C, 161
- torb
 - observation, 50
- transform_plug.C
 - graphicalInterface, 246
 - help, 247
 - plugVersionCheck, 247
- transform_units
 - tempo2.h, 321
 - units.C, 367
- tropo.C, 363
 - compute_tropospheric_delays, 364
 - getMeteorologicalValue, 364
 - getSurfaceAtmosphericPressure, 364
 - getZenithWetDelay, 364
 - initialize_meteorology_table, 364
 - initialize_meteorology_tables, 364
 - MeteorologyFunction_getEndMJD, 365
 - MeteorologyFunction_getStartMJD, 365
 - MeteorologyFunction_getValue, 365
 - MeteorologyFunction_load, 365
 - NMF_hydrostatic, 365
 - NMF_wet, 365
 - surfaceAtmosphericPressureTables, 365
 - zenithWetDelayTables, 365
- troposphericDelay
 - observation, 50
- tt2tb
 - tempo2.h, 321
 - tt2tdb.C, 366
- tt2tdb.C, 365
 - FB_deltaT, 366
 - IF_deltaT, 366
 - init_ifte, 366
 - tt2tb, 366
- turn_deg
 - tempo2.h, 321
 - tempo2Util.C, 332
 - tempo2Util.h, 333
- turn_dms
 - tempo2.h, 321
 - tempo2Util.C, 332
- turn_hms
 - tempo2.h, 322
 - tempo2Util.C, 332
- twot
 - IFTE_interpolation_info, 37
 - interpolation_info, 38
- tzFit
 - polyco.C, 248
- tzrsite
 - pulsar, 72
- UPW
 - choleskyRoutines.h, 97
 - global.C, 127
- USE_BUILTIN_LONGDOUBLE
 - TKlongdouble.float128.h, 346
 - TKlongdouble.h, 347
 - TKlongdouble.ld.h, 349
- USE_NEW_CLK_CORR
 - toa2utc.C, 363
- UT1_FILE
 - tempo2.h, 312
- units
 - pulsar, 72
- units.C, 366
 - scale_param, 367
 - transform_units, 367
 - xform_mjd, 367
- updateBT
 - BTmodel.C, 82
 - tempo2.h, 322
- updateBTJ
 - BTJmodel.C, 81
 - tempo2.h, 322
- updateBTX
 - BTXmodel.C, 83
 - tempo2.h, 322
- updateBatsAll
 - global.C, 126
 - tempo2.h, 322
- updateDD
 - DDmodel.C, 108
 - tempo2.h, 322
- updateDDGR
 - DDGRmodel.C, 105
 - tempo2.h, 322
- updateDDH

- DDHmodel.C, [106](#)
- tempo2.h, [322](#)
- updateDDK
 - DDKmodel.C, [107](#)
 - tempo2.h, [322](#)
- updateDDS
 - DDSmodel.C, [109](#)
 - tempo2.h, [322](#)
- updateDMvals
 - dmmmodel_fitFunc_plug.C, [173](#)
- updateELL1
 - ELL1model.C, [118](#)
 - tempo2.h, [322](#)
- updateELL1H
 - ELL1Hmodel.C, [117](#)
 - tempo2.h, [322](#)
- updateFunctions
 - FitInfo, [32](#)
- updateGlobalParameters
 - doFit.C, [114](#)
- updateJV
 - tempo2.h, [322](#)
- updateMSS
 - MSSmodel.C, [142](#)
 - tempo2.h, [322](#)
- updateParameters
 - doFit.C, [114](#)
 - T2model.C, [288](#)
 - tempo2.h, [322](#)
- updateT2
 - T2model.C, [288](#)
 - tempo2.h, [322](#)
- updateT2_PTA
 - T2-PTAmodel.C, [265](#)
 - tempo2.h, [322](#)
- uranus_earth
 - observation, [50](#)
- useCalceph
 - pulsar, [72](#)
- useSelectFile
 - preProcess.C, [249](#)
 - tempo2.h, [322](#)
- useT2accel
 - T2accel.C, [267](#)
 - T2accel.h, [269](#)
- useTNOrth
 - pulsar, [72](#)
- ut1red
 - tai2ut1.C, [298](#)
- utc2tai
 - tempo2.h, [322](#)
- utc2tai.C, [367](#)
- utc_string
 - T1Polyco, [75](#)
- utfirst
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [356](#)
- utjd
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [356](#)
- utjd1
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [356](#)
- utjd2
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [356](#)
- utjdlast
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [357](#)
- utlast
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [357](#)
- utmean
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [357](#)
- VERSION
 - config.h, [101](#)
- val
 - parameter, [53](#)
- valID
 - calcDMe_plug.C, [159](#)
- vc
 - IFTE_interpolation_info, [37](#)
 - interpolation_info, [38](#)
- vector
 - GWevolve_plug.C, [204](#)
 - interpolate_plug.C, [218](#)
- vectorPulsar
 - tempo2.h, [322](#)
 - vectorPulsar.C, [368](#)
- vectorPulsar.C, [367](#)
- vectorPulsar, [368](#)
- vectorscale
 - tempo2.h, [322](#)
 - tempo2Util.C, [332](#)
- vectorsum
 - tempo2.h, [322](#)
 - tempo2Util.C, [332](#)
- velPulsar
 - pulsar, [72](#)
- venus_earth
 - observation, [50](#)
- verbose_calc_spectra
 - TKspectrum.C, [357](#)
 - TKspectrum.h, [360](#)
- veryFast
 - global.C, [127](#)
 - tempo2.h, [323](#)
- viewModels
 - plk_plug.C, [228](#)
- vl_alpha
 - gwgenSpec, [35](#)
- vl_amp
 - gwgenSpec, [35](#)
- WARNCOLOR

- TKlog.h, [343](#)
- WHEREARG
 - TKlog.h, [343](#)
- WHEREERR
 - TKlog.h, [343](#)
- WHERESTR
 - TKlog.h, [343](#)
- WHERECHK
 - TKlog.h, [343](#)
- WHEREWARN
 - TKlog.h, [343](#)
- WNLEVEL
 - choleskyRoutines.h, [97](#)
 - global.C, [127](#)
- wave_cos
 - pulsar, [72](#)
- wave_cos_dm
 - pulsar, [72](#)
- wave_cos_dm_err
 - pulsar, [72](#)
- wave_cos_err
 - pulsar, [72](#)
- wave_sine
 - pulsar, [72](#)
- wave_sine_dm
 - pulsar, [72](#)
- wave_sine_dm_err
 - pulsar, [72](#)
- wave_sine_err
 - pulsar, [72](#)
- waveScale
 - pulsar, [72](#)
- whiteNoiseModelFile
 - pulsar, [72](#)
- write_debug_files
 - detectGWB_plug.C, [169](#)
 - detectGWBnew_plug.C, [171](#)
- write_python_files
 - detectGWBnew_plug.C, [171](#)
- writeCommands
 - GWwhiteLimit_plug.C, [214](#)
- WriteDesignMatrix
 - designmatrix_plug.C, [167](#)
- writeFiles
 - planet_plug.C, [224](#)
 - spectralModel_plug.C, [242](#)
- writeResiduals
 - TKlog.C, [341](#)
 - TKlog.h, [343](#)
- writeTim
 - readTimfile.C, [257](#)
 - tempo2.h, [322](#)
- x
 - observatory, [51](#)
 - sample, [73](#)
 - TabulatedFunctionSample, [77](#)
 - XY, [78](#)
- X_DISPLAY_MISSING
 - config.h, [101](#)
- XY, [78](#)
 - sigmaz_plug.C, [234](#)
 - x, [78](#)
 - y, [78](#)
- xform_mjd
 - units.C, [367](#)
- xlab
 - calcDMe_plug.C, [159](#)
- xmax
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [357](#)
- xmin
 - sigmaz_plug.C, [237](#)
 - TKspectrum.C, [357](#)
- xp
 - EOPSample, [31](#)
 - GWevolve_plug.C, [205](#)
- xtol
 - lm_control_struct, [40](#)
- y
 - lmcurve_data_struct, [41](#)
 - observatory, [51](#)
 - sample, [73](#)
 - TabulatedFunctionSample, [77](#)
 - XY, [78](#)
- ylab
 - calcDMe_plug.C, [159](#)
- yp
 - EOPSample, [31](#)
 - GWevolve_plug.C, [205](#)
- z
 - observatory, [51](#)
- zenith
 - observation, [50](#)
- zenithWetDelayTables
 - tropo.C, [365](#)
- zoom_graphics
 - tempo2.h, [322](#)