

Tempo2

Generated by Doxygen 1.8.10

Thu Aug 11 2016 13:57:23

Contents

1	Main Page	1
2	User Guide	3
2.1	Tempo2 User Manual	3
2.1.1	About tempo2	3
2.1.2	Terminology and basic usage	3
3	Core Developers	5
4	Developer Guide	7
4.1	Tempo2 Developer Guide	7
4.1.1	About this guide	7
4.1.2	General code guidelines	7
4.1.3	Development workflow	7
4.1.4	Coding style	8
5	Directory structure	11
6	Plugin Documentation	13
6.1	Tempo2 Plugins	13
7	README	15
8	Todo List	19
9	Module Index	21
9.1	Modules	21
10	Class Index	23
10.1	Class List	23
11	File Index	25
11.1	File List	25
12	Module Documentation	27
12.1	libt2toolkit API	27

12.1.1 Detailed Description	27
12.2 libtempo2 External API	28
12.2.1 Detailed Description	28
13 Class Documentation	29
13.1 Cheby2D Struct Reference	29
13.1.1 Member Data Documentation	29
13.1.1.1 coeff	29
13.1.1.2 nx	29
13.1.1.3 ny	29
13.2 ChebyModel Struct Reference	29
13.2.1 Member Data Documentation	30
13.2.1.1 cheby	30
13.2.1.2 dispersion_constant	30
13.2.1.3 freq_end	30
13.2.1.4 freq_start	30
13.2.1.5 frequency_cheby	30
13.2.1.6 mjd_end	30
13.2.1.7 mjd_start	30
13.2.1.8 psrname	30
13.2.1.9 sitename	30
13.3 ChebyModelSet Struct Reference	30
13.3.1 Member Data Documentation	30
13.3.1.1 nsegments	30
13.3.1.2 segments	30
13.4 clock_correction Struct Reference	30
13.4.1 Detailed Description	31
13.4.2 Member Data Documentation	31
13.4.2.1 correction	31
13.4.2.2 corrects_to	31
13.5 complexVal Struct Reference	31
13.5.1 Member Data Documentation	31
13.5.1.1 imag	31
13.5.1.2 real	31
13.6 DynamicArray Struct Reference	31
13.6.1 Member Data Documentation	32
13.6.1.1 data	32
13.6.1.2 elem_size	32
13.6.1.3 nallocated	32
13.6.1.4 nelem	32

13.7	FitInfo Struct Reference	32
13.7.1	Detailed Description	32
13.7.2	Member Data Documentation	32
13.7.2.1	constraintCounters	32
13.7.2.2	constraintDerivs	32
13.7.2.3	constraintIndex	32
13.7.2.4	nConstraints	32
13.7.2.5	nParams	32
13.7.2.6	paramCounters	32
13.7.2.7	paramDerivs	33
13.7.2.8	paramIndex	33
13.7.2.9	updateFunctions	33
13.8	gwgeneralSrc Struct Reference	33
13.8.1	Member Data Documentation	33
13.8.1.1	across_g	33
13.8.1.2	across_im_g	33
13.8.1.3	aplus_g	33
13.8.1.4	aplus_im_g	33
13.8.1.5	asl_g	33
13.8.1.6	asl_im_g	34
13.8.1.7	ast_g	34
13.8.1.8	ast_im_g	34
13.8.1.9	avx_g	34
13.8.1.10	avx_im_g	34
13.8.1.11	avy_g	34
13.8.1.12	avy_im_g	34
13.8.1.13	dist_bin	34
13.8.1.14	h	34
13.8.1.15	h_im	34
13.8.1.16	inc_bin	34
13.8.1.17	kg	34
13.8.1.18	omega_g	34
13.8.1.19	phase_g	34
13.8.1.20	phi_bin	34
13.8.1.21	phi_g	34
13.8.1.22	phi_polar_g	34
13.8.1.23	theta_bin	34
13.8.1.24	theta_g	34
13.9	gwgenSpec Struct Reference	34
13.9.1	Member Data Documentation	35

13.9.1.1	sl_alpha	35
13.9.1.2	sl_amp	35
13.9.1.3	st_alpha	35
13.9.1.4	st_amp	35
13.9.1.5	tensor_alpha	35
13.9.1.6	tensor_amp	35
13.9.1.7	vl_alpha	35
13.9.1.8	vl_amp	35
13.10	gwSrc Struct Reference	35
13.10.1	Member Data Documentation	36
13.10.1.1	across_g	36
13.10.1.2	across_im_g	36
13.10.1.3	aplus_g	36
13.10.1.4	aplus_im_g	36
13.10.1.5	dist_bin	36
13.10.1.6	h	36
13.10.1.7	h_im	36
13.10.1.8	inc_bin	36
13.10.1.9	kg	36
13.10.1.10	omega_g	36
13.10.1.11	phase_g	36
13.10.1.12	phi_bin	36
13.10.1.13	phi_g	36
13.10.1.14	phi_polar_g	36
13.10.1.15	theta_bin	36
13.10.1.16	theta_g	36
13.11	interpolation_info Struct Reference	36
13.11.1	Member Data Documentation	36
13.11.1.1	n_posn_avail	36
13.11.1.2	n_vel_avail	37
13.11.1.3	posn_coeff	37
13.11.1.4	twot	37
13.11.1.5	vel_coeff	37
13.12	jpl_eph_data Struct Reference	37
13.12.1	Member Data Documentation	37
13.12.1.1	au	37
13.12.1.2	cache	37
13.12.1.3	curr_cache_loc	37
13.12.1.4	emrat	37
13.12.1.5	ephem_end	37

13.12.1.6 ephemeris_start	37
13.12.1.7 ephemeris_step	38
13.12.1.8 ephemeris_version	38
13.12.1.9 ifile	38
13.12.1.10 info	38
13.12.1.11 ipt	38
13.12.1.12 kernel_size	38
13.12.1.13 ncoeff	38
13.12.1.14 ncon	38
13.12.1.15 pvsun	38
13.12.1.16 pvsun_t	38
13.12.1.17 recsize	38
13.12.1.18 swap_bytes	38
13.13 observation Struct Reference	38
13.13.1 Detailed Description	40
13.13.2 Member Data Documentation	40
13.13.2.1 addedNoise	40
13.13.2.2 averagebat	40
13.13.2.3 averagedmbat	40
13.13.2.4 averagedmerr	40
13.13.2.5 averagedmres	40
13.13.2.6 averageerr	40
13.13.2.7 averageres	40
13.13.2.8 bat	40
13.13.2.9 batCorr	40
13.13.2.10 bbat	40
13.13.2.11 clockCorr	40
13.13.2.12 correctionsTT	40
13.13.2.13 correctionTT_calcEph	40
13.13.2.14 correctionTT_TB	40
13.13.2.15 correctionTT_Teph	41
13.13.2.16 correctionUT1	41
13.13.2.17 delayCorr	41
13.13.2.18 deleted	41
13.13.2.19 earth_ssb	41
13.13.2.20 earthMoonBary_earth	41
13.13.2.21 earthMoonBary_ssb	41
13.13.2.22 efac	41
13.13.2.23 einsteinRate	41
13.13.2.24 equad	41

13.13.2.25flagID	41
13.13.2.26flagVal	41
13.13.2.27fname	41
13.13.2.28freq	42
13.13.2.29freqSSB	42
13.13.2.30jump	42
13.13.2.31jupiter_earth	42
13.13.2.32hclock_correction	42
13.13.2.33neptune_earth	42
13.13.2.34nFlags	42
13.13.2.35nphase	42
13.13.2.36nutations	42
13.13.2.37observatory_earth	42
13.13.2.38obsNjump	42
13.13.2.39origErr	42
13.13.2.40origsat	42
13.13.2.41pet	42
13.13.2.42phase	43
13.13.2.43phaseOffset	43
13.13.2.44planet_ssb	43
13.13.2.45planet_ssb_derv	43
13.13.2.46planet_ssb_tmr	43
13.13.2.47prefitResidual	43
13.13.2.48psrPos	43
13.13.2.49pulseN	43
13.13.2.50residual	43
13.13.2.51roemer	43
13.13.2.52sat	43
13.13.2.53sat_day	43
13.13.2.54sat_sec	43
13.13.2.55saturn_earth	43
13.13.2.56shapiroDelayJupiter	43
13.13.2.57shapiroDelayNeptune	44
13.13.2.58shapiroDelaySaturn	44
13.13.2.59shapiroDelaySun	44
13.13.2.60shapiroDelayUranus	44
13.13.2.61shapiroDelayVenus	44
13.13.2.62shklovskii	44
13.13.2.63siteVel	44
13.13.2.64sun_earth	44

13.13.2.65sun_ssb	44
13.13.2.66dis1	44
13.13.2.67dis2	44
13.13.2.68ellID	44
13.13.2.69TNDMErr	45
13.13.2.70TNDMSignal	45
13.13.2.71TNGroupErr	45
13.13.2.72TNGroupSignal	45
13.13.2.73TNRedErr	45
13.13.2.74TNRedSignal	45
13.13.2.75toaDMErr	45
13.13.2.76toaErr	45
13.13.2.77torb	45
13.13.2.78troposphericDelay	45
13.13.2.79uranus_earth	45
13.13.2.80venus_earth	45
13.13.2.81zenith	46
13.14observatory Struct Reference	46
13.14.1 Member Data Documentation	46
13.14.1.1 clock_name	46
13.14.1.2 code	46
13.14.1.3 height_grs80	46
13.14.1.4 latitude_grs80	46
13.14.1.5 longitude_grs80	46
13.14.1.6 name	46
13.14.1.7 x	46
13.14.1.8 y	46
13.14.1.9 z	46
13.15parameter Struct Reference	46
13.15.1 Detailed Description	47
13.15.2 Member Data Documentation	47
13.15.2.1 aSize	47
13.15.2.2 err	47
13.15.2.3 fitFlag	47
13.15.2.4 label	47
13.15.2.5 linkFrom	47
13.15.2.6 linkTo	47
13.15.2.7 nLinkFrom	47
13.15.2.8 nLinkTo	47
13.15.2.9 paramSet	47

13.15.2.10prefit	48
13.15.2.11prefitErr	48
13.15.2.12shortlabel	48
13.15.2.13val	48
13.16pulsar Struct Reference	48
13.16.1 Detailed Description	53
13.16.2 Member Data Documentation	54
13.16.2.1 addTNGlobalEQ	54
13.16.2.2 auto_constraints	54
13.16.2.3 AverageDMResiduals	54
13.16.2.4 AverageEpochWidth	54
13.16.2.5 AverageFlag	54
13.16.2.6 AverageResiduals	54
13.16.2.7 binaryModel	54
13.16.2.8 bootStrap	54
13.16.2.9 calcShapiro	54
13.16.2.10cgw_angpol	54
13.16.2.11cgw_cosinc	54
13.16.2.12cgw_h0	54
13.16.2.13cgw_mc	54
13.16.2.14clk_offsE	54
13.16.2.15clk_offsT	54
13.16.2.16clk_offsV	54
13.16.2.17clkOffsN	54
13.16.2.18clock	54
13.16.2.19clockFromOverride	54
13.16.2.20constraints	55
13.16.2.21correctTroposphere	55
13.16.2.22covar	55
13.16.2.23decjStrPost	55
13.16.2.24decjStrPre	55
13.16.2.25decsim	55
13.16.2.26deleteFileName	55
13.16.2.27dilateFreq	55
13.16.2.28dmoffsCM	55
13.16.2.29dmoffsCM_error	55
13.16.2.30dmoffsCM_mjd	55
13.16.2.31dmoffsCM_weight	55
13.16.2.32dmoffsCMnum	55
13.16.2.33dmoffsDM	55

13.16.2.34dmoffsDM_error	55
13.16.2.35dmoffsDM_mjd	55
13.16.2.36dmoffsDM_weight	55
13.16.2.37dmoffsDMnum	55
13.16.2.38dmOffset	55
13.16.2.39eclCoord	56
13.16.2.40eopc04_file	56
13.16.2.41ephemeris	56
13.16.2.42filterStr	56
13.16.2.43fitChisq	56
13.16.2.44fitFunc	56
13.16.2.45fitinfo	56
13.16.2.46fitJump	56
13.16.2.47fitMode	56
13.16.2.48fitNfree	56
13.16.2.49fitParamGlobal	56
13.16.2.50fitParamGlobalK	56
13.16.2.51fitParamI	56
13.16.2.52fitParamK	56
13.16.2.53fixedFormat	56
13.16.2.54jumpID	56
13.16.2.55globalNfit	56
13.16.2.56globalNoConstrain	57
13.16.2.57gwb_decj	57
13.16.2.58gwb_epoch	57
13.16.2.59gwb_geom_c	57
13.16.2.60gwb_geom_p	57
13.16.2.61gwb_raj	57
13.16.2.62gwb_width	57
13.16.2.63gwecc_dec	57
13.16.2.64gwecc_distance	57
13.16.2.65gwecc_e	57
13.16.2.66gwecc_epoch	57
13.16.2.67gwecc_inc	57
13.16.2.68gwecc_m1	57
13.16.2.69gwecc_m2	57
13.16.2.70gwecc_nodes_orientation	57
13.16.2.71gwecc_orbital_period	57
13.16.2.72gwecc_psrdist	57
13.16.2.73gwecc_pulsarTermOn	57

13.16.2.74	gwecc_ra	57
13.16.2.75	gwecc_redshift	57
13.16.2.76	gwecc_theta_0	57
13.16.2.77	gwecc_theta_nodes	57
13.16.2.78	gwm_decj	57
13.16.2.79	gwm_dphase	57
13.16.2.80	gwm_epoch	57
13.16.2.81	gwm_phi	57
13.16.2.82	gwm_raj	57
13.16.2.83	gwsrc_across_i	58
13.16.2.84	gwsrc_across_i_e	58
13.16.2.85	gwsrc_across_r	58
13.16.2.86	gwsrc_across_r_e	58
13.16.2.87	gwsrc_aplus_i	58
13.16.2.88	gwsrc_aplus_i_e	58
13.16.2.89	gwsrc_aplus_r	58
13.16.2.90	gwsrc_aplus_r_e	58
13.16.2.91	gwsrc_dec	58
13.16.2.92	gwsrc_epoch	58
13.16.2.93	gwsrc_psrdist	58
13.16.2.94	gwsrc_ra	58
13.16.2.95	func_weights	58
13.16.2.96	funcE	58
13.16.2.97	funcN	58
13.16.2.98	funcT	58
13.16.2.99	funcV	58
13.16.2.100	pm	58
13.16.2.101	boFormat	58
13.16.2.102	PL_EPHEMERIS	58
13.16.2.103	ampStr	58
13.16.2.104	ampVal	58
13.16.2.105	ampValErr	59
13.16.2.106	ame	59
13.16.2.107	Companion	59
13.16.2.108	constraints	59
13.16.2.109	DMEvents	59
13.16.2.110	dmx	59
13.16.2.111	de_sw	59
13.16.2.112	Fit	59
13.16.2.113	Global	59

13.16.2.114fits	59
13.16.2.115Jumps	59
13.16.2.116obs	59
13.16.2.117Warnings	59
13.16.2.118Param	60
13.16.2.119PhaseJump	60
13.16.2.120Quad	60
13.16.2.121StorePrecision	60
13.16.2.122T2efac	60
13.16.2.123T2equad	60
13.16.2.124TelDX	60
13.16.2.125TelDY	60
13.16.2.126TelDZ	60
13.16.2.127TNBandNoise	60
13.16.2.128TNECORR	60
13.16.2.129TNEF	60
13.16.2.130TNEQ	60
13.16.2.131TNGroupNoise	60
13.16.2.132TNShapeletEvents	60
13.16.2.133TNSQ	60
13.16.2.134TOffset	60
13.16.2.135White	60
13.16.2.136White_dm	60
13.16.2.137bsn	60
13.16.2.138Offset	60
13.16.2.139Offset_e	60
13.16.2.140OutputTMatrix	61
13.16.2.141Param	61
13.16.2.142PassStr	61
13.16.2.143PhaseJump	61
13.16.2.144PhaseJumpDir	61
13.16.2.145PhaseJumpID	61
13.16.2.146PlanetShapiro	61
13.16.2.147PosPulsar	61
13.16.2.148quad_across_i	61
13.16.2.149quad_across_i_e	61
13.16.2.150quad_across_r	61
13.16.2.151quad_across_r_e	61
13.16.2.152quad_aplus_i	61
13.16.2.153quad_aplus_i_e	61

13.16.2.154	quad_aplus_r	61
13.16.2.155	quad_aplus_r_e	61
13.16.2.156	quad_ifunc_c_DEC	61
13.16.2.157	quad_ifunc_c_RA	61
13.16.2.158	quad_ifunc_geom_c	61
13.16.2.159	quad_ifunc_geom_p	61
13.16.2.160	quad_ifunc_p_DEC	62
13.16.2.161	quad_ifunc_p_RA	62
13.16.2.162	quad_ifuncE_c	62
13.16.2.163	quad_ifuncE_p	62
13.16.2.164	quad_ifuncN_c	62
13.16.2.165	quad_ifuncN_p	62
13.16.2.166	quad_ifuncT_c	62
13.16.2.167	quad_ifuncT_p	62
13.16.2.168	quad_ifuncV_c	62
13.16.2.169	quad_ifuncV_p	62
13.16.2.170	quadDEC	62
13.16.2.171	quadEpoch	62
13.16.2.172	quadRA	62
13.16.2.173	qjStrPost	62
13.16.2.174	qjStrPre	62
13.16.2.175	qsim	62
13.16.2.176	qscaleErrChisq	62
13.16.2.177	qnsPost	62
13.16.2.178	qnsPre	62
13.16.2.179	qbust	62
13.16.2.180	qsetTelVelX	62
13.16.2.181	qsetTelVelY	62
13.16.2.182	qsetTelVelZ	62
13.16.2.183	qsetUnits	62
13.16.2.184	qmflag	62
13.16.2.185	qsorted	63
13.16.2.186	qstorePrec	63
13.16.2.187	qswm	63
13.16.2.188	qcMethod	63
13.16.2.189	q2efacFlagID	63
13.16.2.190	q2efacFlagVal	63
13.16.2.191	q12efacVal	63
13.16.2.192	q2equadFlagID	63
13.16.2.193	q2equadFlagVal	63

13.16.2.1912equadVal	63
13.16.2.1952globalEfacs	63
13.16.2.1961DX_e	63
13.16.2.1971DX_t	63
13.16.2.1981DX_v	63
13.16.2.1991DX_vel	63
13.16.2.2001DX_vel_e	63
13.16.2.2011DY_e	63
13.16.2.2021DY_t	63
13.16.2.2031DY_v	63
13.16.2.2041DY_vel	63
13.16.2.2051DY_vel_e	63
13.16.2.2061DZ_e	63
13.16.2.2071DZ_t	63
13.16.2.2081DZ_v	63
13.16.2.2091DZ_vel	64
13.16.2.2101DZ_vel_e	64
13.16.2.211tempo1	64
13.16.2.212meEphemeris	64
13.16.2.213NBandDMAmp	64
13.16.2.214NBandDMC	64
13.16.2.215NBandDMGam	64
13.16.2.216NBandNoiseAmp	64
13.16.2.217NBandNoiseC	64
13.16.2.218NBandNoiseGam	64
13.16.2.219NBandNoiseHF	64
13.16.2.220NBandNoiseLF	64
13.16.2.221NDMAmp	64
13.16.2.222NDMC	64
13.16.2.223NDMCoeffs	64
13.16.2.224NDMEvAmp	64
13.16.2.225NDMEvGam	64
13.16.2.226NDMEvLength	64
13.16.2.227NDMEvLin	64
13.16.2.228NDMEvOff	64
13.16.2.229NDMEvQuad	64
13.16.2.230NDMEvStart	64
13.16.2.231NDMGam	64
13.16.2.232NECORRFlagID	64
13.16.2.233NECORRFlagVal	64

13.16.2.231	NECORRVal	64
13.16.2.232	NEFFFlagID	65
13.16.2.233	NEFFFlagVal	65
13.16.2.234	NEFVal	65
13.16.2.235	NEQFlagID	65
13.16.2.236	NEQFlagVal	65
13.16.2.240	NEQVal	65
13.16.2.241	TNGlobalEF	65
13.16.2.242	TNGlobalEQ	65
13.16.2.243	TNGroupNoiseAmp	65
13.16.2.244	TNGroupNoiseC	65
13.16.2.245	TNGroupNoiseFlagID	65
13.16.2.246	TNGroupNoiseFlagVal	65
13.16.2.247	TNGroupNoiseGam	65
13.16.2.248	TNRedAmp	65
13.16.2.249	TNRedC	65
13.16.2.250	TNRedCoeffs	65
13.16.2.251	TNRedCorner	65
13.16.2.252	TNRedFlow	65
13.16.2.253	TNRedGam	65
13.16.2.254	TNShapeletEvFScale	65
13.16.2.255	TNShapeletEvN	65
13.16.2.256	TNShapeletEvPos	65
13.16.2.257	TNShapeletEvWidth	65
13.16.2.258	TNSQFlagID	65
13.16.2.259	TNSQFlagVal	65
13.16.2.260	TNSQVal	65
13.16.2.261	TNsubtractDM	65
13.16.2.262	TNsubtractRed	65
13.16.2.263	TBAextraCovar	66
13.16.2.264	Offset	66
13.16.2.265	Offset_f1	66
13.16.2.266	Offset_f2	66
13.16.2.267	Offset_t1	66
13.16.2.268	Offset_t2	66
13.16.2.269	OffsetFlags	66
13.16.2.270	OffsetSite	66
13.16.2.271	rsite	66
13.16.2.272	units	66
13.16.2.273	UseCalceph	66

13.16.2.274	useTNorth	66
13.16.2.275	usePulsar	66
13.16.2.276	wave_cos	66
13.16.2.277	wave_cos_dm	66
13.16.2.278	wave_cos_dm_err	66
13.16.2.279	wave_cos_err	66
13.16.2.280	wave_sine	66
13.16.2.281	wave_sine_dm	66
13.16.2.282	wave_sine_dm_err	66
13.16.2.283	wave_sine_err	66
13.16.2.284	waveScale	66
13.16.2.285	WhiteNoiseModelFile	67
13.17	storePrecision Struct Reference	67
13.17.1	Member Data Documentation	67
13.17.1.1	comment	67
13.17.1.2	minPrec	67
13.17.1.3	routine	67
13.18	T1Polyco Struct Reference	67
13.18.1	Member Data Documentation	68
13.18.1.1	binary_frequency	68
13.18.1.2	binary_phase	68
13.18.1.3	coeff	68
13.18.1.4	date_string	68
13.18.1.5	dm	68
13.18.1.6	doppler	68
13.18.1.7	frequency_obs	68
13.18.1.8	frequency_psr_0	68
13.18.1.9	log10rms	68
13.18.1.10	mjd_mid	68
13.18.1.11	ncoeff	68
13.18.1.12	psrname	68
13.18.1.13	reference_phase	68
13.18.1.14	siteName	68
13.18.1.15	span	68
13.18.1.16	utc_string	68
13.19	T1PolycoSet Struct Reference	68
13.19.1	Member Data Documentation	68
13.19.1.1	nsegments	68
13.19.1.2	segments	69
13.20	T2Predictor Struct Reference	69

13.20.1 Member Data Documentation	69
13.20.1.1 cheby	69
13.20.1.2 kind	69
13.20.1.3 modelset	69
13.20.1.4 t1	69
13.21 TabulatedFunction Struct Reference	69
13.21.1 Member Data Documentation	69
13.21.1.1 fileName	69
13.21.1.2 header_line	70
13.21.1.3 samples	70
13.22 TabulatedFunctionSample Struct Reference	70
13.22.1 Member Data Documentation	70
13.22.1.1 x	70
13.22.1.2 y	70
14 File Documentation	71
14.1 cholesky.h File Reference	71
14.1.1 Function Documentation	71
14.1.1.1 cholesky_covarFunc2matrix(double **m, double *covarFunc, int ndays, double *resx, double *resy, double *rese, int np, int nc)	71
14.1.1.2 cholesky_dmModel(double **m, double D, double d, double ref_freq, double *resx, double *resy, double *rese, int np, int nc)	71
14.1.1.3 cholesky_dmModelCovarParam(double **m, double alpha, double a, double b, double *resx, double *resy, double *rese, int np, int nc)	71
14.1.1.4 cholesky_ecm(double **m, char *fileName, double *resx, double *resy, double *rese, int np, int nc)	71
14.1.1.5 cholesky_formUinv(double **uinv, double **m, int np)	71
14.1.1.6 cholesky_powerlawModel(double **m, double modelAlpha, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)	71
14.1.1.7 cholesky_powerlawModel_withBeta(double **m, double modelAlpha, double beta, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)	71
14.1.1.8 cholesky_readFromCovarianceFunction(double **m, const char *fname, double *resx, double *resy, double *rese, int np, int nc)	72
14.2 choleskyRoutines.h File Reference	72
14.3 config.h File Reference	72
14.3.1 Macro Definition Documentation	73
14.3.1.1 _DARWIN_USE_64_BIT_INODE	73
14.3.1.2 F77_FUNC	73
14.3.1.3 F77_FUNC_	73
14.3.1.4 HAVE_BLAS	73
14.3.1.5 HAVE_CFITSIO	73
14.3.1.6 HAVE_DLERROR	73

14.3.1.7	HAVE_DLFCN_H	73
14.3.1.8	HAVE_FFTW3	73
14.3.1.9	HAVE_INTTYPES_H	73
14.3.1.10	HAVE_LAPACK	73
14.3.1.11	HAVE_LIBDL	73
14.3.1.12	HAVE_LIBDLLOADER	73
14.3.1.13	HAVE_LIBM	73
14.3.1.14	HAVE_MEMORY_H	73
14.3.1.15	HAVE_PGPLOT	73
14.3.1.16	HAVE_PTHREAD	73
14.3.1.17	HAVE_STDINT_H	73
14.3.1.18	HAVE_STDLIB_H	73
14.3.1.19	HAVE_STRING_H	73
14.3.1.20	HAVE_STRINGS_H	73
14.3.1.21	HAVE_SYS_STAT_H	73
14.3.1.22	HAVE_SYS_TYPES_H	73
14.3.1.23	HAVE_UNISTD_H	73
14.3.1.24	LT_OBJDIR	73
14.3.1.25	PACKAGE	73
14.3.1.26	PACKAGE_BUGREPORT	73
14.3.1.27	PACKAGE_NAME	73
14.3.1.28	PACKAGE_STRING	74
14.3.1.29	PACKAGE_TARNAME	74
14.3.1.30	PACKAGE_URL	74
14.3.1.31	PACKAGE_VERSION	74
14.3.1.32	STDC_HEADERS	74
14.3.1.33	TEMPO2_ARCH	74
14.3.1.34	VERSION	74
14.4	constraints.h File Reference	74
14.4.1	Function Documentation	74
14.4.1.1	autosetDMCM(pulsar *psr, double dmstep, double cmstep, double start, double end, bool fixCMgrid)	74
14.4.1.2	computeConstraintWeights(pulsar *psr)	74
14.4.1.3	consFunc_dmmodel_cw(pulsar *psr, int ipsr, int i, int k, int order)	74
14.4.1.4	consFunc_dmmodel_cw_year(pulsar *psr, int ipsr, int i, int k, int order)	74
14.4.1.5	consFunc_dmmodel_dm1(pulsar *psr, int ipsr, int i, int k, int order)	74
14.4.1.6	consFunc_dmmodel_mean(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.7	consFunc_ifunc(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.8	consFunc_ifunc_year(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.9	consFunc_qifunc_c_year(pulsar *psr, int ipsr, int i, int k, int order)	75

14.4.1.10 consFunc_qifunc_p_year(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.11 consFunc_quad_ifunc_c(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.12 consFunc_quad_ifunc_p(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.13 consFunc_tel_dx(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.14 consFunc_tel_dy(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.15 consFunc_tel_dz(pulsar *psr, int ipsr, int i, int k, int order)	75
14.4.1.16 CONSTRAINTfuncs(pulsar *psr, int ipsr, int nparams, int iconstraint, double *OUT)	75
14.4.1.17 get_constraint_name(enum constraint c)	75
14.4.1.18 standardConstraintFunctions(pulsar *psr, int ipsr, int iconstraint, int iparam, int constraintk, int k)	75
14.5 documentation/1_USER_GUIDE.md File Reference	75
14.6 documentation/2_developers.md File Reference	75
14.7 documentation/3_DEVELOPER_GUIDE.md File Reference	75
14.8 documentation/4_directories.md File Reference	75
14.9 documentation/5_plugins.md File Reference	75
14.10 dynarr.h File Reference	75
14.10.1 Function Documentation	76
14.10.1.1 DynamicArray_free(DynamicArray *)	76
14.10.1.2 DynamicArray_init(DynamicArray *, size_t elemSize)	76
14.10.1.3 DynamicArray_push_back(DynamicArray *, void *elem)	76
14.10.1.4 DynamicArray_resize(DynamicArray *, size_t nelem)	76
14.11 enum_str.h File Reference	76
14.11.1 Variable Documentation	76
14.11.1.1 constraint_str	76
14.11.1.2 label_str	76
14.12 GWsim.h File Reference	76
14.12.1 Typedef Documentation	77
14.12.1.1 gwgeneralSrc	77
14.12.1.2 gwgenSpec	77
14.12.1.3 gwSrc	77
14.12.2 Function Documentation	77
14.12.2.1 calculateResidualgeneralGW(longdouble *kp, gwgeneralSrc *gw, longdouble time, longdouble dist)	77
14.12.2.2 calculateResidualGW(longdouble *kp, gwSrc *gw, longdouble time, longdouble dist)	77
14.12.2.3 dadt(double ec, double a, double m1, double m2)	77
14.12.2.4 dedt(double ec, double a, double m1, double m2)	77
14.12.2.5 dotProduct(longdouble *m1, longdouble *m2)	77
14.12.2.6 dtdt(double ec, double t, double p)	77

14.12.2.7 eccRes(pulsar *psr, int i, int *coalesceFlag, double *prev_p, double *prev_e, double *prev_a, double *prev_epoch, double *prev_theta)	77
14.12.2.8 eccResWithEnergy(pulsar *psr, int i, int *coalesceFlag, double *prev_p, double *prev_e, double *prev_a, double *prev_epoch, double *prev_theta, float *eOut)	77
14.12.2.9 Fe(double ec)	77
14.12.2.10 Findphi(double prob, double amp, double phase)	78
14.12.2.11 GWanisotropicbackground(gwSrc *gw, int numberGW, long *idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double **harmlist, int nharms)	78
14.12.2.12 GWbackground(gwSrc *gw, int numberGW, long *idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin)	78
14.12.2.13 GWbackground_read(gwSrc *gw, FILE *file, int ireal)	78
14.12.2.14 GWbackground_write(gwSrc *gw, FILE *file, int ngw, int ireal)	78
14.12.2.15 GWdipolebackground(gwSrc *gw, int numberGW, long *idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double *dipoamps)	78
14.12.2.16 GWgeneralanisotropicbackground(gwgeneralSrc *gw, int *numberGW, long *idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin, double ***harmlist, int *nharms)	78
14.12.2.17 GWgeneralbackground(gwgeneralSrc *gw, int *numberGW, long *idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin)	78
14.12.2.18 GWgeneralbackground_read(gwgeneralSrc *gw, FILE *file, int ireal)	78
14.12.2.19 GWgeneralbackground_write(gwgeneralSrc *gw, FILE *file, int ngw, int ireal)	78
14.12.2.20 matrixMult(longdouble m1[3][3], longdouble m2[3][3], longdouble out[3][3])	78
14.12.2.21 psrangle(double centre_long, double centre_lat, double psr_long, double psr_lat)	78
14.12.2.22 Rs(double m1)	78
14.12.2.23 setupgeneralGW(gwgeneralSrc *gw)	78
14.12.2.24 setupGW(gwSrc *gw)	78
14.12.2.25 setupPulsar_GWsim(longdouble ra_p, longdouble dec_p, longdouble *kp)	78
14.12.2.26 spharm(int l, int m, double x)	78
14.13 ifteph.h File Reference	78
14.13.1 Macro Definition Documentation	79
14.13.1.1 IFTE_JD0	79
14.13.1.2 IFTE_K	79
14.13.1.3 IFTE_KM1	79
14.13.1.4 IFTE_LC	79
14.13.1.5 IFTE_MJD0	79
14.13.1.6 IFTE_TEPH0	79
14.13.2 Function Documentation	79
14.13.2.1 IFTE_close_file()	79
14.13.2.2 IFTE_DeltaT(double Teph0, double Teph1)	79
14.13.2.3 IFTE_DeltaTDot(double Teph0, double Teph1)	79
14.13.2.4 IFTE_get_DeltaT_DeltaTDot(double Teph0, double Teph1, double *DeltaT, double *DeltaTDot)	79

14.13.2.5 IFTE_get_vE(double Teph0, double Teph1, double *vE)	79
14.13.2.6 IFTE_get_vE_vEDot(double Teph0, double Teph1, double *ve, double *vEDot)	79
14.13.2.7 IFTE_get_vEDot(double Teph0, double Teph1, double *vEDot)	79
14.13.2.8 IFTE_init(const char *fname)	79
14.14jpl_int.h File Reference	79
14.14.1 Macro Definition Documentation	80
14.14.1.1 JPL_HEADER_SIZE	80
14.14.1.2 MAX_CHEBY	80
14.15jpleph.h File Reference	80
14.15.1 Macro Definition Documentation	81
14.15.1.1 DLL_FUNC	81
14.15.1.2 JPL_EPH_FSEEK_ERROR	81
14.15.1.3 JPL_EPH_INVALID_INDEX	81
14.15.1.4 JPL_EPH_OUTSIDE_RANGE	81
14.15.1.5 JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS	81
14.15.1.6 JPL_EPH_READ_ERROR	81
14.15.1.7 JPL_EPHEM_AU_IN_KM	81
14.15.1.8 JPL_EPHEM_EARTH_MOON_RATIO	81
14.15.1.9 JPL_EPHEM_END_JD	81
14.15.1.10JPL_EPHEM_EPHEMERIS_VERSION	81
14.15.1.11JPL_EPHEM_IPT_ARRAY	81
14.15.1.12JPL_EPHEM_KERNEL_NCOEFF	81
14.15.1.13JPL_EPHEM_KERNEL_RECORD_SIZE	81
14.15.1.14JPL_EPHEM_KERNEL_SIZE	81
14.15.1.15JPL_EPHEM_KERNEL_SWAP_BYTES	81
14.15.1.16JPL_EPHEM_N_CONSTANTS	81
14.15.1.17JPL_EPHEM_START_JD	81
14.15.1.18JPL_EPHEM_STEP	81
14.15.1.19jpl_get_pvsun	81
14.15.1.20JPL_INIT_FILE_CORRUPT	81
14.15.1.21JPL_INIT_FILE_NOT_FOUND	81
14.15.1.22JPL_INIT_FREAD2_FAILED	81
14.15.1.23JPL_INIT_FREAD3_FAILED	81
14.15.1.24JPL_INIT_FREAD4_FAILED	81
14.15.1.25JPL_INIT_FREAD5_FAILED	81
14.15.1.26JPL_INIT_FREAD_FAILED	81
14.15.1.27JPL_INIT_FSEEK_FAILED	81
14.15.1.28JPL_INIT_MEMORY_FAILURE	82
14.15.1.29JPL_INIT_NO_ERROR	82
14.15.1.30JPL_INIT_NOT_CALLED	82

14.15.2 Function Documentation	82
14.15.2.1 jpl_close_ephemeris(void *ephem)	82
14.15.2.2 jpl_get_constant(const int idx, void *ephem, char *constant_name)	82
14.15.2.3 jpl_get_double(const void *ephem, const int value)	82
14.15.2.4 jpl_get_long(const void *ephem, const int value)	82
14.15.2.5 jpl_init_ephemeris(const char *ephemeris_filename, char nam[][6], double *val)	82
14.15.2.6 jpl_init_error_code(void)	82
14.15.2.7 jpl_pleph(void *ephem, const double et, const int ntarg, const int ncent, double rd[], const int calc_velocity)	82
14.15.2.8 jpl_state(void *ephem, const double et, const int list[14], double pv[][6], double nut[4], const int bary)	82
14.15.2.9 make_sub_ephem(void *ephem, const char *sub_filename, const double start← _jd, const double end_jd)	82
14.16 read_fortran.h File Reference	82
14.16.1 Function Documentation	83
14.16.1.1 close_file()	83
14.16.1.2 open_file(char *fname)	83
14.16.1.3 read_char()	83
14.16.1.4 read_character(int len, char *str)	83
14.16.1.5 read_double()	83
14.16.1.6 read_float()	83
14.16.1.7 read_int()	83
14.16.1.8 read_record_int()	83
14.16.2 Variable Documentation	83
14.16.2.1 c_fileptr	83
14.16.2.2 swapByte	83
14.17 read_fortran2.h File Reference	83
14.17.1 Function Documentation	83
14.17.1.1 close_file2()	83
14.17.1.2 open_file2(char *fname, int *swap)	83
14.17.1.3 read_character2(int len, char *str)	83
14.17.1.4 read_double2()	84
14.17.1.5 read_float2()	84
14.17.1.6 read_int2()	84
14.17.1.7 read_record_int2()	84
14.17.2 Variable Documentation	84
14.17.2.1 c_fileptr2	84
14.17.2.2 swapByte2	84
14.18 README.md File Reference	84
14.19 T2accel.h File Reference	84
14.19.1 Macro Definition Documentation	84

14.19.1.1 ACCEL_LSQ	84
14.19.1.2 ACCEL_MULTMATRIX	84
14.19.1.3 ACCEL_UINV	84
14.19.2 Function Documentation	84
14.19.2.1 accel_lsqr(double **dm, double *data, double *oparm, int ndata, int nparam, double **Ocvr)	84
14.19.2.2 accel_multMatrix(double *m1, double *m2, int ndata, int ndata2, int npol, double *out)	84
14.19.2.3 accel_multMatrixVec(double *m1, double *v, int ndata, int npol, double *out) . .	85
14.19.2.4 accel_uinv(double *_m, int n)	85
14.19.3 Variable Documentation	85
14.19.3.1 useT2accel	85
14.20t2fit.h File Reference	85
14.20.1 Function Documentation	85
14.20.1.1 t2Fit(pulsar *psr, unsigned int npsr, const char *covarFuncFile)	85
14.20.1.2 t2Fit_buildConstraintsMatrix(pulsar *psr, int ipsr, int iconstraint, double *afunc) .	85
14.20.1.3 t2Fit_buildDesignMatrix(pulsar *psr, int ipsr, double x, int ipos, double *afunc) . .	85
14.20.1.4 t2Fit_fillFitInfo(pulsar *psr, FitInfo &OUT, const FitInfo &globals)	85
14.20.1.5 t2Fit_fillGlobalFitInfo(pulsar *psr, unsigned int npsr, FitInfo &OUT)	85
14.20.1.6 t2Fit_getFitData(pulsar *psr, double *x, double *y, double *e, int *ip)	85
14.20.1.7 t2Fit_updateParameters(pulsar *psr, int ipsr, double *val, double *error)	85
14.21t2fit_dmmodel.h File Reference	85
14.21.1 Function Documentation	86
14.21.1.1 t2FitFunc_dmmodelCM(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	86
14.21.1.2 t2FitFunc_dmmodelDM(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	86
14.21.1.3 t2UpdateFunc_dmmodelCM(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	86
14.21.1.4 t2UpdateFunc_dmmodelDM(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	86
14.22t2fit_dmother.h File Reference	86
14.22.1 Function Documentation	86
14.22.1.1 t2FitFunc_dmsinusoids(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	86
14.22.1.2 t2FitFunc_dmx(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k) .	86
14.22.1.3 t2FitFunc_fd(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k) . . .	86
14.22.1.4 t2FitFunc_fddc(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k) .	86
14.23t2fit_fitwaves.h File Reference	86
14.23.1 Function Documentation	86
14.23.1.1 t2FitFunc_fitwaves(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	86

14.23.1.2 t2UpdateFunc_fitwaves(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	86
14.24 t2fit_glitch.h File Reference	86
14.24.1 Function Documentation	87
14.24.1.1 t2FitFunc_stdGlitch(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	87
14.24.1.2 t2UpdateFunc_stdGlitch(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	87
14.25 t2fit_ifunc.h File Reference	87
14.25.1 Function Documentation	87
14.25.1.1 ifunc(const double *mjd, const double t, const int N, const int k)	87
14.25.1.2 sinfunc(const double *T, const double t, const int k)	87
14.25.1.3 t2FitFunc_ifunc(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	87
14.25.1.4 t2FitFunc_sifunc(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	87
14.25.1.5 t2UpdateFunc_ifunc(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	87
14.26 t2fit_position.h File Reference	87
14.26.1 Function Documentation	87
14.26.1.1 t2FitFunc_stdPosition(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	87
14.26.1.2 t2UpdateFunc_stdPosition(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	87
14.27 t2fit_stdFitFuncs.h File Reference	88
14.27.1 Function Documentation	88
14.27.1.1 t2FitFunc_binaryModels(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.2 t2FitFunc_ifunc(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.3 t2FitFunc_jump(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.4 t2FitFunc_notImplemented(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.5 t2FitFunc_planet(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.6 t2FitFunc_stdDm(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.7 t2FitFunc_stdFreq(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.8 t2FitFunc_stdGravWav(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.9 t2FitFunc_telPos(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	88
14.27.1.10 t2FitFunc_zero(pulsar *psr, int ipsr, double x, int ipos, param_label label, int k)	89
14.27.1.11 t2UpdateFunc_binaryModels(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.27.1.12 t2UpdateFunc_ifunc(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.27.1.13 t2UpdateFunc_jump(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.27.1.14 t2UpdateFunc_notImplemented(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89

14.27.1.152UpdateFunc_planet(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.27.1.162UpdateFunc_simpleAdd(pulsar *psr, int ipsr, param_label label, int k, double val, double error)	89
14.27.1.172UpdateFunc_simpleMinus(pulsar *psr, int ipsr, param_label label, int k, double val, double error)	89
14.27.1.182UpdateFunc_stdFreq(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.27.1.192UpdateFunc_stdGravWav(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.27.1.202UpdateFunc_telPos(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.27.1.212UpdateFunc_zero(pulsar *psr, int ipsr, param_label label, int k, double val, double err)	89
14.28T2toolkit.h File Reference	89
14.28.1 Detailed Description	90
14.28.2 Function Documentation	90
14.28.2.1 genrand_int32(void)	90
14.28.2.2 genrand_real1(void)	90
14.28.2.3 init_genrand(unsigned long s)	90
14.28.2.4 TKconvertFloat1(double *x, float *ox, int n)	90
14.28.2.5 TKconvertFloat2(double *x, double *y, float *ox, float *oy, int n)	90
14.28.2.6 TKfindMax_d(double *x, int n)	90
14.28.2.7 TKfindMax_f(float *x, int n)	90
14.28.2.8 TKfindMedian_d(double *val, int count)	90
14.28.2.9 TKfindMedian_f(float *val, int count)	90
14.28.2.10TKfindMin_d(double *x, int n)	90
14.28.2.11TKfindMin_f(float *x, int n)	90
14.28.2.12TKfindRMS_d(double *x, int n)	90
14.28.2.13TKfindRMS_f(float *x, int n)	90
14.28.2.14TKfindRMSweight_d(double *x, double *e, int n)	90
14.28.2.15TKgaussDev(long *seed)	90
14.28.2.16TKmean_d(double *x, int n)	90
14.28.2.17TKmean_f(float *x, int n)	90
14.28.2.18TKranDev(long *seed)	91
14.28.2.19TKrange_d(double *x, int n)	91
14.28.2.20TKrange_f(float *x, int n)	91
14.28.2.21TKretMax_d(double a, double b)	91
14.28.2.22TKretMax_f(float a, float b)	91
14.28.2.23TKretMin_d(double a, double b)	91
14.28.2.24TKretMin_f(float a, float b)	91
14.28.2.25TKretMin_i(int a, int b)	91

14.28.2.26TKsetSeed()	91
14.28.2.27TKsign_d(double a, double b)	91
14.28.2.28TKsort_2f(float *val, float *val2, int nobs)	91
14.28.2.29TKsort_3d(double *val, double *val2, double *val3, int nobs)	91
14.28.2.30TKsort_d(double *val, int nobs)	91
14.28.2.31TKsort_f(float *val, int nobs)	91
14.28.2.32TKvariance_d(double *x, int n)	91
14.28.2.33TKzeromean_d(int n, double *y)	91
14.29tabulatedfunction.h File Reference	91
14.29.1 Function Documentation	91
14.29.1.1 TabulatedFunction_getEndX(TabulatedFunction *func)	91
14.29.1.2 TabulatedFunction_getStartX(TabulatedFunction *func)	92
14.29.1.3 TabulatedFunction_getValue(TabulatedFunction *func, double x)	92
14.29.1.4 TabulatedFunction_load(TabulatedFunction *func, char *fileName)	92
14.30tempo2.h File Reference	92
14.30.1 Detailed Description	98
14.30.2 Macro Definition Documentation	98
14.30.2.1 AU_DIST	98
14.30.2.2 AULTSC	98
14.30.2.3 BIG_G	98
14.30.2.4 DM_CONST	98
14.30.2.5 DM_CONST_SI	98
14.30.2.6 ECLIPTIC_OBLIQUITY_VAL	98
14.30.2.7 FB90_TIMEEPH	98
14.30.2.8 GM	99
14.30.2.9 GM_C3	99
14.30.2.10GMJ_C3	99
14.30.2.11GMN_C3	99
14.30.2.12GMS_C3	99
14.30.2.13GMU_C3	99
14.30.2.14GMV_C3	99
14.30.2.15HAVE_GWSIM_H	99
14.30.2.16F99_TIMEEPH	99
14.30.2.17FTEPH_FILE	99
14.30.2.18LEAPSECOND_FILE	99
14.30.2.19MASYR2RADS	99
14.30.2.20MAX_BPJ_JUMPS	99
14.30.2.21MAX_CLK_CORR	100
14.30.2.22MAX_CLKCORR	100
14.30.2.23MAX_COEFF	100

14.30.2.24	MAX_COMPANIONS	100
14.30.2.25	MAX_DM_DERIVATIVES	100
14.30.2.26	MAX_DM_X	100
14.30.2.27	MAX_FILELEN	100
14.30.2.28	MAX_FIT	100
14.30.2.29	MAX_FLAG_LEN	100
14.30.2.30	MAX_FLAGS	100
14.30.2.31	MAX_FREQ_DERIVATIVES	100
14.30.2.32	MAX_IFUNC	100
14.30.2.33	MAX_JUMPS	101
14.30.2.34	MAX_LEAPSEC	101
14.30.2.35	MAX_MSG	101
14.30.2.36	MAX_OBSN_VAL	101
14.30.2.37	MAX_PARAMS	101
14.30.2.38	MAX_PSR_VAL	101
14.30.2.39	MAX_QUAD	101
14.30.2.40	MAX_SITE	101
14.30.2.41	MAX_STOREPRECISION	101
14.30.2.42	MAX_STRLEN	101
14.30.2.43	MAX_T2EFAC	101
14.30.2.44	MAX_T2EQUAD	101
14.30.2.45	MAX_TEL_CLK_OFFS	102
14.30.2.46	MAX_TEL_DX	102
14.30.2.47	MAX_TEL_DY	102
14.30.2.48	MAX_TEL_DZ	102
14.30.2.49	MAX_TNBN	102
14.30.2.50	MAX_TNDMEv	102
14.30.2.51	MAX_TNECORR	102
14.30.2.52	MAX_TNEF	102
14.30.2.53	MAX_TNEQ	102
14.30.2.54	MAX_TNGN	102
14.30.2.55	MAX_TNSQ	102
14.30.2.56	MAX_TOFFSET	102
14.30.2.57	MAX_WHITE	102
14.30.2.58	NE_SW_DEFAULT	103
14.30.2.59	OBLQ	103
14.30.2.60	OBSSYS_FILE	103
14.30.2.61	PCM	103
14.30.2.62	SECDAY	103
14.30.2.63	SECDAYI	103

14.30.2.64	SI_UNITS	103
14.30.2.65	SOLAR_MASS	103
14.30.2.66	SOLAR_RADIUS	103
14.30.2.67	SPEED_LIGHT	103
14.30.2.68	T2C_IAU2000B	103
14.30.2.69	T2C_TEMPO	103
14.30.2.70	TDB_UNITS	103
14.30.2.71	TDBTDT_FILE	104
14.30.2.72	TEMPO2_h_HASH	104
14.30.2.73	TEMPO2_h_MAJOR_VER	104
14.30.2.74	TEMPO2_h_MINOR_VER	104
14.30.2.75	TEMPO2_h_VER	104
14.30.2.76	TSUN	104
14.30.2.77	UT1_FILE	104
14.30.3	Typedef Documentation	104
14.30.3.1	constraint_label	104
14.30.3.2	constraintDerivFunc	104
14.30.3.3	FitInfo	104
14.30.3.4	observation	104
14.30.3.5	param_label	104
14.30.3.6	paramDerivFunc	104
14.30.3.7	parameter	105
14.30.3.8	paramUpdateFunc	105
14.30.3.9	pulsar	105
14.30.3.10	storePrecision	105
14.30.4	Enumeration Type Documentation	105
14.30.4.1	constraint	105
14.30.4.2	label	106
14.30.5	Function Documentation	109
14.30.5.1	allocateMemory(pulsar *psr, int realloc)	109
14.30.5.2	autoConstraints(pulsar *psr, int ipsr, int npsr)	109
14.30.5.3	bootstrap(pulsar *psr, int p, int npsr)	109
14.30.5.4	BTJmodel(pulsar *psr, int p, int obs, int param, int arr)	109
14.30.5.5	BTmodel(pulsar *psr, int p, int obs, int param)	109
14.30.5.6	BTXmodel(pulsar *psr, int p, int obs, int param, int arr)	109
14.30.5.7	calcRMS(pulsar *psr, int p)	109
14.30.5.8	calculate_bclt(pulsar *psr, int npsr)	109
14.30.5.9	compute_tropospheric_delays(pulsar *psr, int npsr)	109
14.30.5.10	copyParam(parameter p1, parameter *p2)	109
14.30.5.11	copyPSR(pulsar *p, int p1, int p2)	109

14.30.5.12	CVSdisplayVersion(const char *file, const char *func, const char *verNum)	110
14.30.5.13	DDGRmodel(pulsar *psr, int p, int obs, int param)	110
14.30.5.14	DDHmodel(pulsar *psr, int p, int obs, int param)	110
14.30.5.15	DDKmodel(pulsar *psr, int p, int obs, int param)	110
14.30.5.16	DDmodel(pulsar *psr, int p, int obs, int param)	110
14.30.5.17	DDSmodel(pulsar *psr, int p, int obs, int param)	110
14.30.5.18	defineClockCorrectionSequence(char *fileList, int dispWarnings)	110
14.30.5.19	destroyMemory(pulsar *psr)	110
14.30.5.20	destroyOne(pulsar *psr)	110
14.30.5.21	displayMsg(int type, const char *key, const char *searchStr, const char *variableStr, int noWarnings)	110
14.30.5.22	displayParameters(int pos, char timeFile[][MAX_FILELEN], char parFile[][MAX_FILELEN], pulsar *psr, int npsr)	110
14.30.5.23	dms_delays(pulsar *psr, int npsr, int p, int i, double delt, double dt_SSB)	110
14.30.5.24	dms_turn(char *line)	110
14.30.5.25	doFit(pulsar *psr, int npsr, int writeModel) DEPRECATED	110
14.30.5.26	doFitAll(pulsar *psr, int npsr, const char *covarFuncFile) DEPRECATED	110
14.30.5.27	doFitDCM(pulsar *psr, const char *dcmFile, const char *covarFuncFile, int npsr, int writeModel) DEPRECATED	110
14.30.5.28	doFitGlobal(pulsar *psr, int npsr, double *globalParameter, int nGlobal, int writeModel) DEPRECATED	110
14.30.5.29	dotproduct(double *v1, double *v2)	110
14.30.5.30	ELL1Hmodel(pulsar *psr, int p, int obs, int param)	110
14.30.5.31	ELL1model(pulsar *psr, int p, int obs, int param)	110
14.30.5.32	equ2ecl(double *x)	110
14.30.5.33	FITfuncs(double x, double afunc[], int ma, pulsar *psr, int ipos, int ipsr)	110
14.30.5.34	formBats(pulsar *psr, int npsr)	110
14.30.5.35	formBatsAll(pulsar *psr, int npsr)	110
14.30.5.36	formResiduals(pulsar *psr, int npsr, int removeMean)	110
14.30.5.37	fortran_mod(longdouble a, longdouble p)	110
14.30.5.38	fortran_nint(double x)	110
14.30.5.39	fortran_nlong(longdouble x)	110
14.30.5.40	get_EOP(double mjd, double *xp, double *yp, double *dut1, double *dut1dot, int dispWarnings, char *eopcFile)	111
14.30.5.41	get_obsCoord(pulsar *psr, int npsr)	111
14.30.5.42	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])	111
14.30.5.43	get_OneobsCoord(pulsar *psr, int npsr, int obs)	111
14.30.5.44	getCholeskyMatrix(double **uinv, const char *fname, pulsar *psr, double *resx, double *resy, double *rese, int np, int nc, int *ip)	111
14.30.5.45	getClockCorrections(observation *obs, const char *clockFrom, const char *clockTo, int warnings)	111

14.30.5.46	getCorrection(observation *obs, const char *clockFrom, const char *clockTo, int warnings)	111
14.30.5.47	getCorrectionTT(observation *obs)	111
14.30.5.48	getInputs(pulsar *psr, int argc, char *argv[], char timFile[][MAX_FILELEN], char parFile[][MAX_FILELEN], int *displayParams, int *npsr, int *nGlobal, int *out← Res, int *writeModel, char *outputSO, int *polyco, char *polyco_args, char *polyco_file, int *newpar, int *onlypre, char *dcmFile, char *covarFuncFile, char *newparname)	111
14.30.5.49	getObservatory(char *code)	111
14.30.5.50	getParamDeriv(pulsar *psr, int ipos, double x, int i, int k) DEPRECATED	111
14.30.5.51	getParameterValue(pulsar *psr, int param, int arr)	111
14.30.5.52	hms_turn(char *line)	111
14.30.5.53	d_residual(float xcurs, float ycurs)	111
14.30.5.54	initialise(pulsar *psr, int noWarnings)	111
14.30.5.55	initialiseOne(pulsar *psr, int noWarnings, int fullSetup)	111
14.30.5.56	JVmodel(pulsar *psr, int p, int obs, int param, int arr)	111
14.30.5.57	logicFlag(char *line, pulsar *psr, int npsr)	111
14.30.5.58	lookup_observatory_alias(char *incod, char *outcode)	111
14.30.5.59	MSSmodel(pulsar *psr, int p, int obs, int param)	111
14.30.5.60	polyco(pulsar *psr, int npsr, longdouble polyco_MJD1, longdouble polyco_MJ← D2, int nspan, int ncoeff, longdouble maxha, char *sitename, longdouble freq, longdouble coeff[MAX_COEFF], int trueDM, char *polyco_file)	111
14.30.5.61	preProcess(pulsar *psr, int npsr, int argc, char *argv[])	111
14.30.5.62	preProcessSimple(pulsar *psr)	111
14.30.5.63	preProcessSimple1(pulsar *psr, int tempo1, double thelast)	111
14.30.5.64	preProcessSimple2(pulsar *psr, float startdmmjd, int ndm, float *dmvals, int tri← monly)	112
14.30.5.65	preProcessSimple3(pulsar *psr)	112
14.30.5.66	processFlag(char *line, pulsar *psr, int npsr)	112
14.30.5.67	processSimultaneous(char *line, pulsar *psr, int npsr)	112
14.30.5.68	readEphemeris(pulsar *psr, int npsr, int addEphemNoise)	112
14.30.5.69	readEphemeris_calceph(pulsar *psr, int npsr)	112
14.30.5.70	readJBO_bat(char *fname, pulsar *psr, int p)	112
14.30.5.71	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])	112
14.30.5.72	readOneEphemeris(pulsar *psr, int npsr, int addEphemNoise, int obsNumber)	112
14.30.5.73	readParfile(pulsar *psr, char parFile[][MAX_FILELEN], char timFile[][MAX_FIL← ELEN], int npsr)	112
14.30.5.74	readParfileGlobal(pulsar *psr, int npsr, char tpar[MAX_STRLEN][MAX_FILELE← N], char ttim[MAX_STRLEN][MAX_FILELEN])	112
14.30.5.75	readSimpleParfile(FILE *fin, pulsar *p)	112
14.30.5.76	readTimfile(pulsar *psr, char timFile[][MAX_FILELEN], int npsr)	112

14.30.5.77	recordPrecision(pulsar *psr, longdouble prec, const char *routine, const char *comment)	112
14.30.5.78	secularMotion(pulsar *psr, int npsr)	112
14.30.5.79	setPlugPath()	112
14.30.5.80	setStart(float xcurs, float ycurs, int flag)	112
14.30.5.81	setupParameterFileDefaults(pulsar *p)	112
14.30.5.82	shapiro_delay(pulsar *psr, int npsr, int p, int i, double delt, double dt_SSB)	112
14.30.5.83	simplePlot(pulsar *psr, double unitFlag)	112
14.30.5.84	solarWindModel(pulsar psr, int iobs)	112
14.30.5.85	sortToAs(pulsar *psr)	112
14.30.5.86	T2_PTAmodel(pulsar *psr, int p, int obs, int param, int arr)	112
14.30.5.87	T2model(pulsar *psr, int p, int obs, int param, int arr)	112
14.30.5.88	tai2tt(pulsar *psr, int npsr)	112
14.30.5.89	tai2ut1(pulsar *psr, int npsr)	112
14.30.5.90	textOutput(pulsar *psr, int npsr, double globalParameter, int nGlobal, int outRes, int newpar, const char *fname)	113
14.30.5.91	toa2utc(pulsar *psr, int npsr)	113
14.30.5.92	transform_units(struct pulsar *psr, int from, int to)	113
14.30.5.93	tt2b(pulsar *psr, int npsr)	113
14.30.5.94	tt2b_calceph(pulsar *psr, int npsr)	113
14.30.5.95	turn_deg(double turn)	113
14.30.5.96	turn_dms(double turn, char *dms)	113
14.30.5.97	turn_hms(double turn, char *hms)	113
14.30.5.98	updateBatsAll(pulsar *psr, int npsr)	113
14.30.5.99	updateBT(pulsar *psr, double val, double err, int pos)	113
14.30.5.100	updateBTJ(pulsar *psr, double val, double err, int pos, int arr)	113
14.30.5.101	updateBTX(pulsar *psr, double val, double err, int pos, int arr)	113
14.30.5.102	updateDD(pulsar *psr, double val, double err, int pos)	113
14.30.5.103	updateDDGR(pulsar *psr, double val, double err, int pos)	113
14.30.5.104	updateDDH(pulsar *psr, double val, double err, int pos)	113
14.30.5.105	updateDDK(pulsar *psr, double val, double err, int pos)	113
14.30.5.106	updateDDS(pulsar *psr, double val, double err, int pos)	113
14.30.5.107	updateELL1(pulsar *psr, double val, double err, int pos)	113
14.30.5.108	updateELL1H(pulsar *psr, double val, double err, int pos)	113
14.30.5.109	updateJV(pulsar *psr, double val, double err, int pos, int arr)	113
14.30.5.110	updateMSS(pulsar *psr, double val, double err, int pos)	113
14.30.5.111	updateParameters(pulsar *psr, int p, double *val, double *error)	113
14.30.5.112	updateT2(pulsar *psr, double val, double err, int pos, int arr)	113
14.30.5.113	updateT2_PTA(pulsar *psr, double val, double err, int pos, int arr)	113
14.30.5.114	useSelectFile(char *fname, pulsar *psr, int npsr)	113

14.30.5.11	<code>tc2tai(pulsar *psr, int npsr)</code>	113
14.30.5.11	<code>VectorPulsar(pulsar *psr, int npsr)</code>	113
14.30.5.11	<code>Vectorscale(double *v, double k)</code>	113
14.30.5.11	<code>Vectorsum(double *res, double *v1, double *v2)</code>	114
14.30.5.11	<code>WriteTim(const char *timname, pulsar *psr, const char *fileFormat)</code>	114
14.30.5.12	<code>boom_graphics(float xcurs2, float ycurs2, int flag)</code>	114
14.30.6	Variable Documentation	114
14.30.6.1	<code>covarFuncFile</code>	114
14.30.6.2	<code>dcmFile</code>	114
14.30.6.3	<code>displayCVSversion</code>	114
14.30.6.4	<code>ECLIPTIC_OBLIQUITY</code>	114
14.30.6.5	<code>forceGlobalFit</code>	114
14.30.6.6	<code>MAX_OBSN</code>	114
14.30.6.7	<code>MAX_PSR</code>	114
14.30.6.8	<code>NEWFIT</code>	114
14.30.6.9	<code>TEMPO2_ENVIRON</code>	114
14.30.6.10	<code>TEMPO2_ERROR</code>	114
14.30.6.11	<code>tempo2_plug_path</code>	114
14.30.6.12	<code>tempo2_plug_path_len</code>	115
14.30.6.13	<code>tempo2MachineType</code>	115
14.30.6.14	<code>veryFast</code>	115
14.31	<code>tempo2pred.h</code> File Reference	115
14.31.1	Enumeration Type Documentation	116
14.31.1.1	<code>T2PredictorKind</code>	116
14.31.2	Function Documentation	116
14.31.2.1	<code>T2Predictor_Copy(T2Predictor *into_t2p, const T2Predictor *from_t2p)</code>	116
14.31.2.2	<code>T2Predictor_Destroy(T2Predictor *t2p)</code>	116
14.31.2.3	<code>T2Predictor_FRead(T2Predictor *t2p, FILE *f)</code>	116
14.31.2.4	<code>T2Predictor_FWrite(const T2Predictor *t2p, FILE *f)</code>	116
14.31.2.5	<code>T2Predictor_GetEndFreq(T2Predictor *t2p)</code>	116
14.31.2.6	<code>T2Predictor_GetEndMJD(T2Predictor *t2p)</code>	116
14.31.2.7	<code>T2Predictor_GetFrequency(const T2Predictor *t2p, long double mjd, long double freq)</code>	116
14.31.2.8	<code>T2Predictor_GetPhase(const T2Predictor *t2p, long double mjd, long double freq)</code>	116
14.31.2.9	<code>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)</code>	116
14.31.2.10	<code>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)</code>	116
14.31.2.11	<code>T2Predictor_GetPSRName(T2Predictor *t2p)</code>	116

14.31.2.12	T2Predictor_GetSiteName(T2Predictor *t2p)	116
14.31.2.13	T2Predictor_GetStartFreq(T2Predictor *t2p)	116
14.31.2.14	T2Predictor_GetStartMJD(T2Predictor *t2p)	116
14.31.2.15	T2Predictor_Init(T2Predictor *t2p)	116
14.31.2.16	T2Predictor_Insert(T2Predictor *into_t2p, const T2Predictor *from_t2p)	116
14.31.2.17	T2Predictor_Keep(T2Predictor *, unsigned nmjd, const long double *mjd)	116
14.31.2.18	T2Predictor_Kind(T2Predictor *t2p)	116
14.31.2.19	T2Predictor_Read(T2Predictor *t2p, char *fname)	117
14.31.2.20	T2Predictor_Write(const T2Predictor *t2p, char *fname)	117
14.31.3	Variable Documentation	117
14.31.3.1	ChebyModelSet_OutOfRange	117
14.32	tempo2pred_int.h File Reference	117
14.32.1	Function Documentation	118
14.32.1.1	Cheby2D_Construct(Cheby2D *cheby, void(*func)(long double *x, long double *y, int nx, int ny, long double *z, void *info), void *info)	118
14.32.1.2	Cheby2D_Construct_x_Derivative(Cheby2D *dcheby, const Cheby2D *cheby)	118
14.32.1.3	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)	118
14.32.1.4	ChebyModel_Construct(ChebyModel *cm, const pulsar *psr)	118
14.32.1.5	ChebyModel_Copy(ChebyModel *cm, ChebyModel *from)	118
14.32.1.6	ChebyModel_Destroy(ChebyModel *cm)	118
14.32.1.7	ChebyModel_GetFrequency(const ChebyModel *cm, long double mjd, long double freq)	118
14.32.1.8	ChebyModel_GetPhase(const ChebyModel *cm, long double mjd, long double freq)	118
14.32.1.9	ChebyModel_Init(ChebyModel *cm, int nmjdcoeff, int nfreqcoeff)	118
14.32.1.10	ChebyModel_Read(ChebyModel *cm, FILE *f)	118
14.32.1.11	ChebyModel_Test(ChebyModel *cm, const pulsar *psr, int nmjd, int nfreq, long double *residualRMS, long double *residualMAV)	118
14.32.1.12	ChebyModel_Write(const ChebyModel *cm, FILE *f)	118
14.32.1.13	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)	118
14.32.1.14	ChebyModelSet_Destroy(ChebyModelSet *cms)	118
14.32.1.15	ChebyModelSet_GetFrequency(const ChebyModelSet *cms, long double mjd, long double freq)	118
14.32.1.16	ChebyModelSet_GetNearest(const ChebyModelSet *cms, long double mjd)	118
14.32.1.17	ChebyModelSet_GetPhase(const ChebyModelSet *cms, long double mjd, long double freq)	118
14.32.1.18	ChebyModelSet_Init(ChebyModelSet *cms)	118
14.32.1.19	ChebyModelSet_Insert(ChebyModelSet *cms, const ChebyModelSet *from)	118

14.32.1.20	ChebyModelSet_Keep(ChebyModelSet *cms, unsigned nmjd, const long double *mjd)	118
14.32.1.21	ChebyModelSet_Read(ChebyModelSet *cms, FILE *f)	118
14.32.1.22	ChebyModelSet_Test(ChebyModelSet *cms, const pulsar *psr, int nmjd, int nfreq, long double *residualRMS, long double *residualMAV)	118
14.32.1.23	ChebyModelSet_Write(const ChebyModelSet *cms, FILE *f)	118
14.32.1.24	T1Polyco_GetFrequency(const T1Polyco *t1p, long double mjd, long double freq)	118
14.32.1.25	T1Polyco_GetPhase(const T1Polyco *t1p, long double mjd, long double freq)	119
14.32.1.26	T1Polyco_Read(T1Polyco *t1p, FILE *f)	119
14.32.1.27	T1Polyco_Write(const T1Polyco *t1p, FILE *f)	119
14.32.1.28	T1PolycoSet_Destroy(T1PolycoSet *t1ps)	119
14.32.1.29	T1PolycoSet_GetFrequency(const T1PolycoSet *t1ps, long double mjd, long double freq)	119
14.32.1.30	T1PolycoSet_GetNearest(long double mjd)	119
14.32.1.31	T1PolycoSet_GetPhase(const T1PolycoSet *t1ps, long double mjd, long double freq)	119
14.32.1.32	T1PolycoSet_Read(T1PolycoSet *t1ps, FILE *f)	119
14.32.1.33	T1PolycoSet_Write(const T1PolycoSet *t1ps, FILE *f)	119
14.33	tempo2Util.h File Reference	119
14.33.1	Function Documentation	119
14.33.1.1	dms_turn(char *line)	119
14.33.1.2	hms_turn(char *line)	119
14.33.1.3	turn_deg(double turn)	119
14.34	TKcholesky.h File Reference	119
14.34.1	Function Documentation	120
14.34.1.1	cholesky_covarFunc2matrix(double **m, double *covarFunc, int ndays, double *resx, double *resy, double *rese, int np, int nc)	120
14.34.1.2	cholesky_dmModel(double **m, double D, double d, double ref_freq, double *resx, double *resy, double *rese, int np, int nc)	120
14.34.1.3	cholesky_dmModelCovarParam(double **m, double alpha, double a, double b, double *resx, double *resy, double *rese, int np, int nc)	120
14.34.1.4	cholesky_ecm(double **m, char *fileName, double *resx, double *resy, double *rese, int np, int nc)	120
14.34.1.5	cholesky_formUinv(double **uinv, double **m, int np)	120
14.34.1.6	cholesky_powerlawModel(double **m, double modelAlpha, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)	120
14.34.1.7	cholesky_powerlawModel_withBeta(double **m, double modelAlpha, double beta, double modelFc, double modelA, double *resx, double *resy, double *rese, int np, int nc)	120
14.34.1.8	cholesky_readFromCovarianceFunction(double **m, const char *fname, double *resx, double *resy, double *rese, int np, int nc)	120
14.35	TKfit.h File Reference	120
14.35.1	Function Documentation	120

14.35.1.1 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)	121
14.35.1.2 TKfindPoly_d(double *px, double *py, int n, int m, double *p)	121
14.35.1.3 TKfitPoly(double x, double *v, int m)	121
14.35.1.4 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)	121
14.35.1.5 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)	121
14.35.1.6 TKleastSquares_svd_noErr(double *x, double *y, int n, double *p, int nf, void(*fitFuncs)(double, double[], int))	121
14.35.1.7 TKremovePoly_d(double *px, double *py, int n, int m)	121
14.35.1.8 TKremovePoly_f(float *px, float *py, int n, int m)	121
14.35.1.9 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)	121
14.35.1.10 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)	121
14.36 TKlog.h File Reference	121
14.36.1 Macro Definition Documentation	122
14.36.1.1 _LOG	122
14.36.1.2 BOLDCOLOR	122
14.36.1.3 DEPRECATED	122
14.36.1.4 ENDERR	122
14.36.1.5 ENDL	122
14.36.1.6 ERRORCOLOR	122
14.36.1.7 LOG_OUTFILE	122
14.36.1.8 logdbg	122
14.36.1.9 logerr	122
14.36.1.10 logmsg	122
14.36.1.11 logtchk	122
14.36.1.12 logwarn	123
14.36.1.13 RESETCOLOR	123
14.36.1.14 TK_MAX_ERROR_LEN	123
14.36.1.15 TK_MAX_ERRORS	123
14.36.1.16 TK_STORE_ERROR	123
14.36.1.17 TK_STORE_WARNING	123
14.36.1.18 WARNCOLOR	123
14.36.1.19 WHEREARG	123
14.36.1.20 WHEREERR	123

14.36.1.21	WHERESTR	123
14.36.1.22	WHERECHK	123
14.36.1.23	WHEREWARN	123
14.36.2	Function Documentation	123
14.36.2.1	_TKchklog(FILE *, const char *,...)	123
14.36.2.2	logerr_check()	123
14.36.3	Variable Documentation	123
14.36.3.1	debugFlag	123
14.36.3.2	tcheck	123
14.36.3.3	timer_clk	123
14.36.3.4	TK_errorCount	123
14.36.3.5	TK_errorlog	123
14.36.3.6	TK_warnCount	123
14.36.3.7	TK_warnlog	123
14.36.3.8	writeResiduals	123
14.37	TKlongdouble.float128.h File Reference	123
14.37.1	Macro Definition Documentation	124
14.37.1.1	cosl	124
14.37.1.2	fabsl	124
14.37.1.3	floorl	124
14.37.1.4	FMT_LD	124
14.37.1.5	LD_PI	124
14.37.1.6	longdouble	124
14.37.1.7	LONGDOUBLE_IS_FLOAT128	124
14.37.1.8	LONGDOUBLE_ONE	124
14.37.1.9	powl	124
14.37.1.10	sinl	124
14.37.1.11	USE_BUILTIN_LONGDOUBLE	124
14.37.2	Typedef Documentation	124
14.37.2.1	longdouble	125
14.37.3	Function Documentation	125
14.37.3.1	ld_fprintf(FILE *__stream, const char *__format,...)	125
14.37.3.2	ld_printf(const char *__format,...)	125
14.37.3.3	ld_sprintf(char *__str, const char *__format,...)	125
14.37.3.4	parse_longdouble(const char *str)	125
14.38	TKlongdouble.h File Reference	125
14.38.1	Macro Definition Documentation	125
14.38.1.1	ld_fprintf	125
14.38.1.2	LD_PI	125
14.38.1.3	ld_printf	125

14.38.1.4	<code>ld_sprintf</code>	125
14.38.1.5	<code>longdouble</code>	125
14.38.1.6	<code>LONGDOUBLE_IS_IEEE754</code>	125
14.38.1.7	<code>LONGDOUBLE_ONE</code>	125
14.38.1.8	<code>USE_BUILTIN_LONGDOUBLE</code>	126
14.38.2	Typedef Documentation	126
14.38.2.1	<code>longdouble</code>	126
14.38.3	Function Documentation	126
14.38.3.1	<code>parse_longdouble(const char *str)</code>	126
14.39TK	<code>longdouble.ld.h</code> File Reference	126
14.39.1	Macro Definition Documentation	126
14.39.1.1	<code>ld_fprintf</code>	126
14.39.1.2	<code>LD_PI</code>	126
14.39.1.3	<code>ld_printf</code>	126
14.39.1.4	<code>ld_sprintf</code>	126
14.39.1.5	<code>longdouble</code>	126
14.39.1.6	<code>LONGDOUBLE_IS_IEEE754</code>	126
14.39.1.7	<code>LONGDOUBLE_ONE</code>	126
14.39.1.8	<code>USE_BUILTIN_LONGDOUBLE</code>	126
14.39.2	Typedef Documentation	127
14.39.2.1	<code>longdouble</code>	127
14.39.3	Function Documentation	127
14.39.3.1	<code>parse_longdouble(const char *str)</code>	127
14.40TK	<code>matrix.h</code> File Reference	127
14.40.1	Function Documentation	127
14.40.1.1	<code>free_2df(float **uinv)</code>	127
14.40.1.2	<code>free_blas(double **matrix)</code>	127
14.40.1.3	<code>free_uinv(double **uinv)</code>	127
14.40.1.4	<code>get_blas_cols(double **uinv)</code>	127
14.40.1.5	<code>get_blas_rows(double **uinv)</code>	127
14.40.1.6	<code>malloc_2df(int rows, int cols)</code>	127
14.40.1.7	<code>malloc_blas(int n, int m)</code>	127
14.40.1.8	<code>malloc_uinv(int n)</code>	127
14.40.1.9	<code>TKmultMatrix(double **idcm, double **u, int ndata, int ndata2, int npol, double **uout)</code>	127
14.40.1.10	<code>TKmultMatrix_sq(double **idcm, double **u, int ndata, int npol, double **uout)</code>	127
14.40.1.11	<code>TKmultMatrixVec(double **idcm, double *b, int ndata, int ndata2, double *bout)</code>	127
14.40.1.12	<code>TKmultMatrixVec_sq(double **idcm, double *b, int ndata, double *bout)</code>	127
14.41TK	<code>spectrum.h</code> File Reference	128
14.41.1	Macro Definition Documentation	129

14.41.1.1 ABS	129
14.41.1.2 MAX	129
14.41.1.3 MIN	129
14.41.2 Typedef Documentation	129
14.41.2.1 complexVal	129
14.41.3 Function Documentation	129
14.41.3.1 calcSpectra(double **uinv, double *resx, double *resy, int nres, double *specX, double *specY, int nfit)	129
14.41.3.2 calcSpectraErr(double **uinv, double *resx, double *resy, int nres, double *specX, double *specY, double *specE, int nfit)	129
14.41.3.3 fit4(int *nfit, double *p4, double *cov4, int ndostats, double *chidf, double *avewt)	129
14.41.3.4 getprtj(int n)	129
14.41.3.5 getweights(int n, double *wt)	129
14.41.3.6 indexx8(int n, double *arrin, int *indx)	129
14.41.3.7 mat20(double sam[21][21], double a[21][21], int n, double *determ, int *nbad)	129
14.41.3.8 sineFunc(double x, double *v, int ma)	129
14.41.3.9 TK_dft(double *x, double *y, int n, double *outX, double *outY, int *outN, double *outY_re, double *outY_im)	129
14.41.3.10TK_fft(short int dir, long n, double *x, double *y)	129
14.41.3.11TK_fitSine(double *x, double *y, double *e, int n, int wErr, double *outX, double *outY, int *outN)	129
14.41.3.12TK_fitSinusoids(double *x, double *y, double *sig, int n, double *outX, double *outY, int *outN)	129
14.41.3.13TK_weightLS(double *x, double *y, double *sig, int n, double *outX, double *outY, int *outN, double *outY_re, double *outY_im)	129
14.41.3.14TKaveragePts(double *x, double *y, int n, int width, double *meanX, double *meanY, int *nMean)	129
14.41.3.15TKboxcar(double *x, double *y, int n, double *ox, double *oy, int *on, int width)	129
14.41.3.16TKcmonot(int n, double x[], double y[], double yd[][4])	129
14.41.3.17TKfirstDifference(double *x, double *y, int n)	129
14.41.3.18TKhann(double *x, double *y, int n, double *ox, double *oy, int *on, int width)	129
14.41.3.19TKinterpolateSplineSmoothFixedXPts(double *inX, double *inY, int inN, double *interpX, double *interpY, int nInterp)	129
14.41.3.20TKlomb_d(double *x, double *y, int n, double ofac, double hifac, double *ox, dou- ble *oy, int *outN, double *var)	130
14.41.3.21TKsortit(double *x, double *y, int n)	130
14.41.3.22TKspectrum(double *x, double *y, double *e, int n, int averageTime, int smooth← Width, int smoothType, int fitSpline, int preWhite, int specType, double ofac, dou- ble hifac, int specOut, double *outX, double *outY, int *nout, int calcWhite, int output, double *outY_re, double *outY_im)	130
14.41.3.23TKspline_interpolate(int n, double *x, double *y, double yd[][4], double *interpX, double *interpY, int nInterp)	130
14.41.4 Variable Documentation	130
14.41.4.1 verbose_calc_spectra	130

14.42TKsvd.h File Reference	130
14.42.1 Function Documentation	130
14.42.1.1 TKbacksubstitution_svd(longdouble **V, longdouble *w, longdouble **U, longdouble *b, longdouble *x, int n, int nf)	130
14.42.1.2 TKbidiagonal(longdouble **a, longdouble *anorm, int ndata, int nfit, longdouble **v, longdouble *w, longdouble **u, longdouble *rv1)	130
14.42.1.3 TKpythag(longdouble a, longdouble b)	130
14.42.1.4 TKsingularValueDecomposition_lsqr(longdouble **designMatrix, int n, int nf, longdouble **v, longdouble *w, longdouble **u)	130
Index	131

Chapter 1

Main Page

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

Chapter 2

User Guide

2.1 Tempo2 User Manual

2.1.1 About tempo2

Tempo2 is a pulsar timing package, based on the old fortran tempo code to address some shortcomings in that code for high precision pulsar timing. Over the years tempo2 has been expanded by many [developers](#), and has grown to become the premier package for all kinds of pulsar timing experiments.

For more details on pulsar timing in general, you may wish to read the Tempo2 paper series:

- I. An overview <http://adsabs.harvard.edu/abs/2006MNRAS.369..655H>
- II. The timing model and precision estimates <http://adsabs.harvard.edu/abs/2006MNRAS.372.1549E>
- III. Gravitational wave simulation <http://adsabs.harvard.edu/abs/2009MNRAS.394.1945H>

There is also a lot of useful information on the tempo2 wiki, <http://www.atnf.csiro.au/research/pulsar/tempo2/index.php?n=Main.HomePage>. Some of the details are outdated as of 2015, but the general principles are sound.

The wiki is the best place for tutorials and basic introduction to tempo2.

2.1.2 Terminology and basic usage

This documentation will focus on providing some basic overview of the many functions of tempo2 and is mostly intended as a reference for those who have mastered the basics. However, for completeness, here we will cover the most basic functions of tempo2.

Tempo2 brings together time-of-arrival measurements (ToAs), stored in a `.tim` file, and a pulsar ephemeris stored in a `.par` file to produce the difference between the pulsar ephemeris model and the actual arrival times. This step is generally known as "forming residuals", and depends on having accurate models of the Earth ephemeris and of the clocks used to measure the ToAs. If all is well, these differences will be consistent with the uncertainty in the measurements. This is not generally the case, therefore the second part of tempo2 is a fitting routine that attempts to update the model parameters to get the best-fit model.

The basic usage of tempo2 is to feed in a `.par` and a `.tim` file, form residuals, do the fit and write out the best-fit parameters.

```
tempo2 -f example1.par example1.tim -newpar
```

This will write out `new.par` file with the updated parameters, as well as printing to the console the pre and post-fit parameters. Note any warnings that are printed. One of the side-effects of tempo2 is that it sometimes prints a lot of warnings, some you can ignore and some that you can't, so you have to read them!

If you compiled the `pgplot` plugins, you can run the graphical interface `plk`

```
tempo2 -gr plk -f example1.par example1.tim
```

Running plugins

There are many, many plugins. Some plugins are better supported than others. To get a list of the plugins you have installed try `tempo2 -h`. The majority of plugins are "graphical" plugins, even if they do not use graphics. This is to do with the way that the plugin is called, rather than anything to do with it being graphical. Graphical plugins are run with the `-gr` option. A few other types of plugins exist:

- `-gr <plugin_name>` for so-called "graphical" plugins. This is most plugins.
- `-output <plugin_name>` for "output" plugins, like `general` and `general2`
- `-fitfunc <plugin_name>` for alternative fit routines. This is likely to be removed in a future release.
- `-select <plugin_name>` for ToA filtering plugins.

You may have to review the source code if you can't find documentation for the plugin you desire. See the [Plugin Documentation](#) for more details on the available plugins.

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

Developer Guide

4.1 Tempo2 Developer Guide

4.1.1 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.

4.1.2 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.

4.1.3 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.

4.1.4 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 libraries 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.

    // usestdint types where possible to avoid confusion on 32-bit vs 64-bit machines.
    // use unsigned types where suitable
    // 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++) {
    // ...
}
}

```

Note

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

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.

```

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

/*!
 * @brief A brief description of the function
 * @param anInt[in] description of this parameter
 * @param str[in] description of this parameter
 * @param matrix[out] description, note if it is an "output" parameter!
 *
 * More description if required
 */

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

```


Chapter 5

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 \$TEMP← PO2 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 6

Plugin Documentation

6.1 Tempo2 Plugins

[TOC]

Chapter 7

README

! [Build Status] (<https://drone.io/bitbucket.org/psrsoft/tempo2/status.png>)

Git INSTALLATION README

0. Contents

1. What this package is
2. Quick Guide
3. Requirements
4. Detailed instalation guide
5. Plugins
6. Changes from old makefile
7. Installation troubleshooting
8. Bugs and feature requests

1. What this package is

You (or someone else) have checked out tempo2 from the Git (<https://bitbucket.org/psrsoft/tempo2>)

This is the best way to get the latest/cutting edge version, and develop your own additions to the tempo2 code or via plugins.

For more information on tempo2 see: <http://www.atnf.csiro.au/research/pulsar/tempo2/>

This requires the gnu autotools. If you don't have or don't want to install autotools, we recommend you install the latest distributed release from <http://www.atnf.csiro.au/research/pulsar/tempo2/> or use PSRSoft to install tempo2: <http://www.pulsarastronomy.net/wiki/Software/PSRSoft>

2. Quick Guide

Bootstrap the build system:

```
./bootstrap
```

setup the tempo2 runtime dir

```
cp -r T2runtime /usr/share/tempo2/
export TEMPO2=/usr/share/tempo2/
```

Configure:

```
./configure [--prefix=/your/install/path]]
```

use `--prefix` to set the path you want to install the binaries and libraries

Make and install...

```
make && make install
```

You will probably want to build the default plugins (plk, etc). Do this with:

```
make plugins && make plugins-install
```

And you're done.

3. Requirements

Tempo2 requires the following:

- A fortran 77 compiler (tested with gfortran).
- A C compiler (tested with gcc).

Plugins may have other requirements, notably PGPLOT.

5. Plugins

The bootstrap command will create suitable makefiles for the default set of plugins. This is controlled by the contents of the files in `./plugin/plugin_lists/`

- `vanilla.plugins` lists plugins to install which have no dependancies.
- `pgplot.plugins` lists plugins to install that are dependant on PGPLOT.
- `gsl.plugins` lists plugins to install that are dependant on the GSL.

5.1 Building your own plugin

The easiest way to compile your own plugins is:

```
g++ ${CFLAGS} ${LDFLAGS} -fPIC -shared -o ${TEMPO2}/plugins/${PLG_NAME}_${LOGIN_ARCH}_plug.t2 ${SRCLIST}
```

where:

- `{ $PLG_NAME }` is the name of your plugin
- `{ $SRCLIST }` is your plugin's source code.
- `{ $LOGIN_ARCH }` is the result of ``uname`` (usually Linux).
- `{ $CFLAGS }` are the compiler flags your plugin needs... remeber to add a `-I` option to point to the location of [tempo2.h](#)
- `{ $LDFLAGS }` are any linking options you need, e.g. `pgplot`, etc.
- `{ $TEMPO2 }` is the tempo2 runtime dir

For example, to compile a basic plugin called 'foo' on linux, you might do

```
g++ -I/usr/src/tempo2 -fPIC -shared -o $TEMPO2/plugins/foo_${LOGIN_ARCH}_plug.t2 foo_plug.C
```


5.2 Adding a new plugin to the default build list

If your plugin has dependences that are already covered by the lists above, just add the name to the appropriate list, and name your plugin source file as:

name_plug.C

6. Changes from the old Make system.

At the start of 2010, tempo2 moved over to an autotools based make system, replacing the old hand written make-files. This may confuse some people!

Important notes:

- Tempo2 plugins now have a .t2 extension, rather than the old .so This is to ensure reduce confusion on MacOSx and to allow the old make system and the new make system to co-exist for a while.
- Any 3rd party plugins will still work as before. Indeed, to update a plugin, just change the .so extension to a .t2 extension. e.g. mv general_Linux_plug.so general_Linux_plug.t2

7. Installation Troubleshooting

7.1 Can't find PGPLOT

Download pgplot from: <http://www.astro.caltech.edu/~tjp/pgplot/>

Or use PSRSOFT to manage the installation. <http://www.pulsarastronomy.net/wiki/Software/PSRSoft>

If you have pgplot installed, but it is not detected by the configure script, check:

- You have got at least libpgplot.a and libcpgplot.a in your LDFLAGS
- Check you have \$PGPLOT_DIR pointing to the folder with grfont.dat and rgb.txt
- Check that you have \$F77 set to the same compiler that compiled PGPLOT (e.g. setenv F77 gfortran, if you used gfortran for PGPLOT)

7.2 Incompatible C and Fortran compilers

Check that you are using the same build of gcc and gfortran (or whatever compiler you are using).

Note that on MacOSX there is often an issue where the default compiler is incompatible with gfortran. The gfortran compatible version is often called gcc-4 and gxx-4 or similar. Use this with:

```
export CC=gcc-4
export CXX=g++-4
```

and reconfigure.

8. Bugs and feature requests

Please submit bug reports here: <https://bitbucket.org/psrsoft/tempo2/issues/new>

Note that it is very helpful if you can upload a small example demonstrating the bug!

Chapter 8

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 9

Module Index

9.1 Modules

Here is a list of all modules:

libt2toolkit API	27
libtempo2 External API	28

Chapter 10

Class Index

10.1 Class List

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

Cheby2D	29
ChebyModel	29
ChebyModelSet	30
clock_correction	30
complexVal	31
DynamicArray	31
FitInfo	
Details of the fit	32
gwgeneralSrc	33
gwgenSpec	34
gwSrc	35
interpolation_info	36
jpl_eph_data	37
observation	
A struct containing the details of a single obesrvation	38
observatory	46
parameter	
Holds the values for a parameter	46
pulsar	
Details for a single pulsar	48
storePrecision	67
T1Polyco	67
T1PolycoSet	68
T2Predictor	69
TabulatedFunction	69
TabulatedFunctionSample	70

Chapter 11

File Index

11.1 File List

Here is a list of all files with brief descriptions:

cholesky.h	71
choleskyRoutines.h	72
config.h	72
constraints.h	74
dynarr.h	75
enum_str.h	76
GWsim.h	76
ifteph.h	78
jpl_int.h	79
jpleph.h	80
read_fortran.h	82
read_fortran2.h	83
T2accel.h	84
t2fit.h	85
t2fit_dmmodel.h	85
t2fit_dmother.h	86
t2fit_fitwaves.h	86
t2fit_glitch.h	86
t2fit_ifunc.h	87
t2fit_position.h	87
t2fit_stdFitFuncs.h	88
T2toolkit.h	
Set of routines that are commonly used in tempo2 and/or its plugins	89
tabulatedfunction.h	91
tempo2.h	
Main interface to libtempo2	92
tempo2pred.h	115
tempo2pred_int.h	117
tempo2Util.h	119
TKcholesky.h	119
TKfit.h	120
TKlog.h	121
TKlongdouble.float128.h	123
TKlongdouble.h	125
TKlongdouble.ld.h	126
TKmatrix.h	127
TKspectrum.h	128
TKsvd.h	130

Chapter 12

Module Documentation

12.1 libt2toolkit API

Files

- file [T2toolkit.h](#)

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

12.1.1 Detailed Description

12.2 libtempo2 External API

Files

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

12.2.1 Detailed Description

Chapter 13

Class Documentation

13.1 Cheby2D Struct Reference

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

Public Attributes

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

13.1.1 Member Data Documentation

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

13.1.1.2 [int Cheby2D::nx](#)

13.1.1.3 [int Cheby2D::ny](#)

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

- [tempo2pred.h](#)

13.2 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](#)

13.2.1 Member Data Documentation

13.2.1.1 Cheby2D ChebyModel::cheby

13.2.1.2 long double ChebyModel::dispersion_constant

13.2.1.3 long double ChebyModel::freq_end

13.2.1.4 long double ChebyModel::freq_start

13.2.1.5 Cheby2D ChebyModel::frequency_cheby

13.2.1.6 long double ChebyModel::mjd_end

13.2.1.7 long double ChebyModel::mjd_start

13.2.1.8 char ChebyModel::psrname[64]

13.2.1.9 char ChebyModel::sitename[64]

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

- [tempo2pred.h](#)

13.3 ChebyModelSet Struct Reference

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

Collaboration diagram for ChebyModelSet:

Public Attributes

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

13.3.1 Member Data Documentation

13.3.1.1 int ChebyModelSet::nsegments

13.3.1.2 ChebyModel* ChebyModelSet::segments

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

- [tempo2pred.h](#)

13.4 clock_correction Struct Reference

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

Public Attributes

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

13.4.1 Detailed Description

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

13.4.2 Member Data Documentation

13.4.2.1 double clock_correction::correction

13.4.2.2 char clock_correction::corrects_to[32]

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

- [tempo2.h](#)

13.5 complexVal Struct Reference

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

Public Attributes

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

13.5.1 Member Data Documentation

13.5.1.1 double complexVal::imag

13.5.1.2 double complexVal::real

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

- [TKspectrum.h](#)

13.6 DynamicArray Struct Reference

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

Public Attributes

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

13.6.1 Member Data Documentation

13.6.1.1 `void* DynamicArray::data`

13.6.1.2 `size_t DynamicArray::elem_size`

13.6.1.3 `size_t DynamicArray::nallocated`

13.6.1.4 `size_t DynamicArray::nelem`

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

- [dynarr.h](#)

13.7 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](#)]

13.7.1 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.

13.7.2 Member Data Documentation

13.7.2.1 `int FitInfo::constraintCounters[MAX_FIT]`

13.7.2.2 `constraintDerivFunc FitInfo::constraintDerivs[MAX_FIT]`

13.7.2.3 `constraint_label FitInfo::constraintIndex[MAX_FIT]`

13.7.2.4 `unsigned FitInfo::nConstraints`

13.7.2.5 `unsigned FitInfo::nParams`

13.7.2.6 `int FitInfo::paramCounters[MAX_FIT]`

13.7.2.7 **paramDerivFunc** FitInfo::paramDerivs[MAX_FIT]

13.7.2.8 **param_label** FitInfo::paramIndex[MAX_FIT]

13.7.2.9 **paramUpdateFunc** FitInfo::updateFunctions[MAX_FIT]

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

- [tempo2.h](#)

13.8 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 \[3\]](#)

13.8.1 Member Data Documentation

13.8.1.1 **longdouble** gwgeneralSrc::across_g

13.8.1.2 **longdouble** gwgeneralSrc::across_im_g

13.8.1.3 **longdouble** gwgeneralSrc::aplus_g

13.8.1.4 **longdouble** gwgeneralSrc::aplus_im_g

13.8.1.5 **longdouble** gwgeneralSrc::asl_g

- 13.8.1.6 `longdouble gwgeneralSrc::asl_im_g`
- 13.8.1.7 `longdouble gwgeneralSrc::ast_g`
- 13.8.1.8 `longdouble gwgeneralSrc::ast_im_g`
- 13.8.1.9 `longdouble gwgeneralSrc::avx_g`
- 13.8.1.10 `longdouble gwgeneralSrc::avx_im_g`
- 13.8.1.11 `longdouble gwgeneralSrc::avy_g`
- 13.8.1.12 `longdouble gwgeneralSrc::avy_im_g`
- 13.8.1.13 `longdouble gwgeneralSrc::dist_bin`
- 13.8.1.14 `longdouble gwgeneralSrc::h[3][3]`
- 13.8.1.15 `longdouble gwgeneralSrc::h_im[3][3]`
- 13.8.1.16 `longdouble gwgeneralSrc::inc_bin`
- 13.8.1.17 `longdouble gwgeneralSrc::kg[3]`
- 13.8.1.18 `longdouble gwgeneralSrc::omega_g`
- 13.8.1.19 `longdouble gwgeneralSrc::phase_g`
- 13.8.1.20 `longdouble gwgeneralSrc::phi_bin`
- 13.8.1.21 `longdouble gwgeneralSrc::phi_g`
- 13.8.1.22 `longdouble gwgeneralSrc::phi_polar_g`
- 13.8.1.23 `longdouble gwgeneralSrc::theta_bin`
- 13.8.1.24 `longdouble gwgeneralSrc::theta_g`

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

- [GWsim.h](#)

13.9 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](#)

13.9.1 Member Data Documentation

13.9.1.1 double `gwgenSpec::sl_alpha`

13.9.1.2 double `gwgenSpec::sl_amp`

13.9.1.3 double `gwgenSpec::st_alpha`

13.9.1.4 double `gwgenSpec::st_amp`

13.9.1.5 double `gwgenSpec::tensor_alpha`

13.9.1.6 double `gwgenSpec::tensor_amp`

13.9.1.7 double `gwgenSpec::vl_alpha`

13.9.1.8 double `gwgenSpec::vl_amp`

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

- [GWsim.h](#)

13.10 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 \[3\]](#)

13.10.1 Member Data Documentation

- 13.10.1.1 `longdouble gwSrc::across_g`
- 13.10.1.2 `longdouble gwSrc::across_im_g`
- 13.10.1.3 `longdouble gwSrc::aplus_g`
- 13.10.1.4 `longdouble gwSrc::aplus_im_g`
- 13.10.1.5 `longdouble gwSrc::dist_bin`
- 13.10.1.6 `longdouble gwSrc::h[3][3]`
- 13.10.1.7 `longdouble gwSrc::h_im[3][3]`
- 13.10.1.8 `longdouble gwSrc::inc_bin`
- 13.10.1.9 `longdouble gwSrc::kg[3]`
- 13.10.1.10 `longdouble gwSrc::omega_g`
- 13.10.1.11 `longdouble gwSrc::phase_g`
- 13.10.1.12 `longdouble gwSrc::phi_bin`
- 13.10.1.13 `longdouble gwSrc::phi_g`
- 13.10.1.14 `longdouble gwSrc::phi_polar_g`
- 13.10.1.15 `longdouble gwSrc::theta_bin`
- 13.10.1.16 `longdouble gwSrc::theta_g`

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

- [GWSim.h](#)

13.11 interpolation_info Struct Reference

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

Public Attributes

- double `posn_coeff` [`MAX_CHEBY`]
- double `vel_coeff` [`MAX_CHEBY`]
- double `twot`
- unsigned `n_posn_avail`
- unsigned `n_vel_avail`

13.11.1 Member Data Documentation

- 13.11.1.1 `unsigned interpolation_info::n_posn_avail`

13.11.1.2 unsigned interpolation_info::n_vel_avail

13.11.1.3 double interpolation_info::posn_coeff[MAX_CHEBY]

13.11.1.4 double interpolation_info::twot

13.11.1.5 double interpolation_info::vel_coeff[MAX_CHEBY]

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

- [jpl_int.h](#)

13.12 jpl_eph_data Struct Reference

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

Collaboration diagram for jpl_eph_data:

Public Attributes

- double [ephem_start](#)
- double [ephem_end](#)
- double [ephem_step](#)
- uint32_t [ncon](#)
- double [au](#)
- double [emrat](#)
- uint32_t [ipt](#) [15][3]
- uint32_t [ephemeris_version](#)
- uint32_t [kernel_size](#)
- uint32_t [recsize](#)
- uint32_t [ncoeff](#)
- uint32_t [swap_bytes](#)
- uint32_t [curr_cache_loc](#)
- double [pvsun](#) [9]
- double [pvsun_t](#)
- double * [cache](#)
- struct [interpolation_info](#) [iinfo](#)
- FILE * [ifile](#)

13.12.1 Member Data Documentation

13.12.1.1 double jpl_eph_data::au

13.12.1.2 double* jpl_eph_data::cache

13.12.1.3 uint32_t jpl_eph_data::curr_cache_loc

13.12.1.4 double jpl_eph_data::emrat

13.12.1.5 double jpl_eph_data::ephem_end

13.12.1.6 double jpl_eph_data::ephem_start

- 13.12.1.7 double `jpl_eph_data::ephem_step`
- 13.12.1.8 uint32_t `jpl_eph_data::ephemeris_version`
- 13.12.1.9 FILE* `jpl_eph_data::ifile`
- 13.12.1.10 struct `interpolation_info` `jpl_eph_data::iinfo`
- 13.12.1.11 uint32_t `jpl_eph_data::ipt[15][3]`
- 13.12.1.12 uint32_t `jpl_eph_data::kernel_size`
- 13.12.1.13 uint32_t `jpl_eph_data::ncoeff`
- 13.12.1.14 uint32_t `jpl_eph_data::ncon`
- 13.12.1.15 double `jpl_eph_data::pvsun[9]`
- 13.12.1.16 double `jpl_eph_data::pvsun_t`
- 13.12.1.17 uint32_t `jpl_eph_data::recsize`
- 13.12.1.18 uint32_t `jpl_eph_data::swap_bytes`

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

- [jpl_int.h](#)

13.13 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](#)
- double [averagedmbat](#)
- double [averagedmres](#)
- double [averagedmerr](#)
- char [fname](#) [MAX_FILELEN]
- char [telID](#) [100]
- [clock_correction](#) [correctionsTT](#) [MAX_CLK_CORR]
- int [nclock_correction](#)
- [longdouble](#) [correctionTT_TB](#)
- double [einsteinRate](#)
- [longdouble](#) [correctionTT_calcEph](#)
- [longdouble](#) [correctionTT_Teph](#)
- [longdouble](#) [correctionUT1](#)
- double [sun_ssb](#) [6]
- double [sun_earth](#) [6]
- double [planet_ssb](#) [9][6]
- double [planet_ssb_tmr](#) [9][6]
- double [planet_ssb_derv](#) [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](#)

13.13.1 Detailed Description

A struct containing the details of a single obseration.

13.13.2 Member Data Documentation

13.13.2.1 double `observation::addedNoise`

13.13.2.2 double `observation::averagebat`

13.13.2.3 double `observation::averagedmbat`

13.13.2.4 double `observation::averagedmerr`

13.13.2.5 double `observation::averagedmres`

13.13.2.6 double `observation::averageerr`

13.13.2.7 double `observation::averageres`

13.13.2.8 **longdouble** `observation::bat`

Infinite frequency barycentric arrival time

13.13.2.9 **longdouble** `observation::batCorr`

13.13.2.10 **longdouble** `observation::bbat`

Arrival time at binary barycentre

13.13.2.11 int `observation::clockCorr`

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

13.13.2.12 **clock_correction** `observation::correctionsTT[MAX_CLK_CORR]`

chain of corrections from site TOA to chosen realisation of TT

13.13.2.13 **longdouble** `observation::correctionTT_calcEph`

13.13.2.14 **longdouble** `observation::correctionTT_TB`

Correction to TDB/TCB

13.13.2.15 longdouble observation::correctionTT_Teph

Correction to Teph

13.13.2.16 longdouble observation::correctionUT1

Correction from site TOA to UT1

13.13.2.17 int observation::delayCorr

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

13.13.2.18 int observation::deleted

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

13.13.2.19 double observation::earth_ssb[6]

Centre of Earth w.r.t. SSB

13.13.2.20 double observation::earthMoonBary_earth[6]

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

13.13.2.21 double observation::earthMoonBary_ssb[6]

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

13.13.2.22 double observation::efac

Error multiplication factor

13.13.2.23 double observation::einsteinRate

Derivative of correctionTT_TB

13.13.2.24 double observation::equad

Value to add in quadrature

13.13.2.25 char observation::flagID[MAX_FLAGS][MAX_FLAG_LEN]

Flags in .tim file

13.13.2.26 char observation::flagVal[MAX_FLAGS][MAX_FLAG_LEN]**13.13.2.27 char observation::fname[MAX_FILELEN]**

Name of data file giving TOA

13.13.2.28 `double observation::freq`

Frequency of observation (in MHz)

13.13.2.29 `double observation::freqSSB`

Frequency of observation in barycentric frame (in Hz)

13.13.2.30 `int observation::jump[MAX_FLAGS]`

Jump region

13.13.2.31 `double observation::jupiter_earth[6]`

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

13.13.2.32 `int observation::nclock_correction`

13.13.2.33 `double observation::neptune_earth[6]`

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

13.13.2.34 `int observation::nFlags`

13.13.2.35 `longdouble observation::nphase`

allows the pulse number to be determined

13.13.2.36 `double observation::nutations[6]`

13.13.2.37 `double observation::observatory_earth[6]`

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

13.13.2.38 `int observation::obsNjump`

Number of jumps for this observation

13.13.2.39 `double observation::origErr`

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

13.13.2.40 `longdouble observation::origsat`

13.13.2.41 `longdouble observation::pet`

Pulsar emission time

13.13.2.42 **longdouble** observation::phase

13.13.2.43 **double** observation::phaseOffset

Phase offset

13.13.2.44 **double** observation::planet_ssb[9][6]

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

13.13.2.45 **double** observation::planet_ssb_derv[9][6]

13.13.2.46 **double** observation::planet_ssb_tmr[9][6]

13.13.2.47 **longdouble** observation::prefitResidual

Pre-fit residual

13.13.2.48 **double** observation::psrPos[3]

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

13.13.2.49 **long long** observation::pulseN

Pulse number

13.13.2.50 **longdouble** observation::residual

residual

13.13.2.51 **longdouble** observation::roemer

Roemer delay

13.13.2.52 **longdouble** observation::sat

Site arrival time

13.13.2.53 **longdouble** observation::sat_day

13.13.2.54 **longdouble** observation::sat_sec

13.13.2.55 **double** observation::saturn_earth[6]

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

13.13.2.56 **double** observation::shapiroDelayJupiter

Shapiro Delay due to Jupiter

13.13.2.57 `double observation::shapiroDelayNeptune`

Shapiro Delay due to Neptune

13.13.2.58 `double observation::shapiroDelaySaturn`

Shapiro Delay due to Saturn

13.13.2.59 `double observation::shapiroDelaySun`

Shapiro Delay due to the Sun

13.13.2.60 `double observation::shapiroDelayUranus`

Shapiro Delay due to Uranus

13.13.2.61 `double observation::shapiroDelayVenus`

Shapiro Delay due to Venus

13.13.2.62 `longdouble observation::shklovskii`

Shklovskii delay term

13.13.2.63 `double observation::siteVel[3]`

Observatory velocity w.r.t. geocentre

13.13.2.64 `double observation::sun_earth[6]`

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

13.13.2.65 `double observation::sun_ssb[6]`

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

13.13.2.66 `double observation::tdis1`

Interstellar dispersion measure delay

13.13.2.67 `double observation::tdis2`

Dispersion measure delay due to solar system

13.13.2.68 `char observation::telID[100]`

Telescope ID

13.13.2.69 double observation::TNDMErr

Error on Model DM signal from temponest fit

13.13.2.70 double observation::TNDMSignal

Model DM signal from temponest fit

13.13.2.71 double observation::TNGroupErr

Error on Model Group Noise signal from temponest fit

13.13.2.72 double observation::TNGroupSignal

Model Group Noise signal from temponest fit

13.13.2.73 double observation::TNRedErr

Error on Model red noise signal from temponest fit

13.13.2.74 double observation::TNRedSignal

Model red noise signal from temponest fit

13.13.2.75 double observation::toaDMErr

Error on TOA due to DM (in us)

13.13.2.76 double observation::toaErr

Error on TOA (in us)

13.13.2.77 longdouble observation::torb

Combined binary delays

13.13.2.78 double observation::troposphericDelay

Delay due to neutral refraction in atmosphere

13.13.2.79 double observation::uranus_earth[6]

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

13.13.2.80 double observation::venus_earth[6]

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

13.13.2.81 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](#)

13.14 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]

13.14.1 Member Data Documentation

13.14.1.1 char observatory::clock_name[16]

13.14.1.2 char observatory::code[16]

13.14.1.3 double observatory::height_grs80

13.14.1.4 double observatory::latitude_grs80

13.14.1.5 double observatory::longitude_grs80

13.14.1.6 char observatory::name[32]

13.14.1.7 double observatory::x

13.14.1.8 double observatory::y

13.14.1.9 double observatory::z

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

- [tempo2.h](#)

13.15 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](#)

13.15.1 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

13.15.2 Member Data Documentation

13.15.2.1 int parameter::aSize

Number of elements in the array for this parameter

13.15.2.2 [longdouble](#)* parameter::err

Uncertainty on parameter value

13.15.2.3 int* parameter::fitFlag

= 1 if fitting required, = 2 for global fit

13.15.2.4 char** parameter::label

Label about this parameter

13.15.2.5 int parameter::linkFrom[5]

13.15.2.6 int parameter::linkTo[5]

13.15.2.7 int parameter::nLinkFrom

13.15.2.8 int parameter::nLinkTo

13.15.2.9 int* parameter::paramSet

= 1 if parameter has been set

13.15.2.10 `longdouble* parameter::prefit`

Pre-fit value of the parameter

13.15.2.11 `longdouble* parameter::prefitErr`

Pre-fit value of the uncertainty

13.15.2.12 `char parameter::shortlabel`**

Label about this parameter without units

13.15.2.13 `longdouble* parameter::val`

Value of parameter

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

- [tempo2.h](#)

13.16 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 [gwsrc_ra](#)
- double [gwsrc_dec](#)
- double [gwsrc_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](#)

- int [AverageDMResiduals](#)
- 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](#)

13.16.1 Detailed Description

contains the details for a single pulsar.

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

13.16.2 Member Data Documentation

13.16.2.1 `double pulsar::addTNGlobalEQ`

13.16.2.2 `char pulsar::auto_constraints`

13.16.2.3 `int pulsar::AverageDMResiduals`

13.16.2.4 `float pulsar::AverageEpochWidth`

13.16.2.5 `char pulsar::AverageFlag[MAX_FLAG_LEN]`

13.16.2.6 `int pulsar::AverageResiduals`

13.16.2.7 `char pulsar::binaryModel[100]`

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

13.16.2.8 `int pulsar::bootStrap`

0 if calculating errors using bootstrap Monte-Carlo method

13.16.2.9 `int pulsar::calcShapiro`

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

13.16.2.10 `double pulsar::cgw_angpol`

13.16.2.11 `double pulsar::cgw_cosinc`

13.16.2.12 `double pulsar::cgw_h0`

13.16.2.13 `double pulsar::cgw_mc`

13.16.2.14 `double pulsar::clk_offsE[MAX_TEL_CLK_OFFS]`

13.16.2.15 `double pulsar::clk_offsT[MAX_TEL_CLK_OFFS]`

13.16.2.16 `double pulsar::clk_offsV[MAX_TEL_CLK_OFFS]`

13.16.2.17 `int pulsar::clkOffsN`

13.16.2.18 `char pulsar::clock[16]`

Clock standard to use as "UTC"

13.16.2.19 `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)

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

Which constraints are specified

13.16.2.21 `int pulsar::correctTroposphere`

whether or not do correct for tropospheric delay

13.16.2.22 `double** pulsar::covar`

13.16.2.23 `char pulsar::decjStrPost[100]`

String containing RAJ and DECJ (postfit)

13.16.2.24 `char pulsar::decjStrPre[100]`

String containing RAJ and DECJ (prefit)

13.16.2.25 `double pulsar::decsim`

13.16.2.26 `char pulsar::deleteFileName[100]`

File name containing deleted points

13.16.2.27 `int pulsar::dilateFreq`

whether or not to apply SS time dilation to RFs

13.16.2.28 `double pulsar::dmoffsCM[MAX_IFUNC]`

13.16.2.29 `double pulsar::dmoffsCM_error[MAX_IFUNC]`

13.16.2.30 `double pulsar::dmoffsCM_mjd[MAX_IFUNC]`

13.16.2.31 `double pulsar::dmoffsCM_weight[MAX_IFUNC]`

13.16.2.32 `int pulsar::dmoffsCMnum`

13.16.2.33 `double pulsar::dmoffsDM[MAX_IFUNC]`

13.16.2.34 `double pulsar::dmoffsDM_error[MAX_IFUNC]`

13.16.2.35 `double pulsar::dmoffsDM_mjd[MAX_IFUNC]`

13.16.2.36 `double pulsar::dmoffsDM_weight[MAX_IFUNC]`

13.16.2.37 `int pulsar::dmoffsDMnum`

13.16.2.38 `double pulsar::dmOffset`

Value to add to DM flags

13.16.2.39 `int pulsar::eclCoord`

= 1 for ecliptic coords otherwise celestial coords

13.16.2.40 `char pulsar::eopc04_file[MAX_FILELEN]`

13.16.2.41 `char pulsar::ephemeris[MAX_FILELEN]`

13.16.2.42 `char pulsar::filterStr[MAX_STRLEN]`

String describing filters

13.16.2.43 `double pulsar::fitChisq`

Chisq value from the fit

13.16.2.44 `char pulsar::fitFunc[MAX_FILELEN]`

13.16.2.45 `FitInfo pulsar::fitinfo`

13.16.2.46 `int pulsar::fitJump[MAX_JUMPS]`

= 1 if fit for jump

13.16.2.47 `int pulsar::fitMode`

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

13.16.2.48 `int pulsar::fitNfree`

Number of degrees of freedom in fit

13.16.2.49 `int pulsar::fitParamGlobal[MAX_FIT]`

13.16.2.50 `int pulsar::fitParamGlobalK[MAX_FIT]`

13.16.2.51 `int pulsar::fitParamI[MAX_FIT]`

13.16.2.52 `int pulsar::fitParamK[MAX_FIT]`

13.16.2.53 `int pulsar::fixedFormat`

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

13.16.2.54 `char pulsar::fjumpID[16]`

13.16.2.55 `int pulsar::globalNfit`

Total number of parameters in the fit

13.16.2.56 int pulsar::globalNoConstrain

Total number of points without constraints

13.16.2.57 double pulsar::gwb_decj

13.16.2.58 double pulsar::gwb_epoch

13.16.2.59 double pulsar::gwb_geom_c

13.16.2.60 double pulsar::gwb_geom_p

13.16.2.61 double pulsar::gwb_raj

13.16.2.62 double pulsar::gwb_width

13.16.2.63 double pulsar::gwecc_dec

13.16.2.64 double pulsar::gwecc_distance

13.16.2.65 double pulsar::gwecc_e

13.16.2.66 double pulsar::gwecc_epoch

13.16.2.67 double pulsar::gwecc_inc

13.16.2.68 double pulsar::gwecc_m1

13.16.2.69 double pulsar::gwecc_m2

13.16.2.70 double pulsar::gwecc_nodes_orientation

13.16.2.71 double pulsar::gwecc_orbital_period

13.16.2.72 double pulsar::gwecc_psrdist

13.16.2.73 int pulsar::gwecc_pulsarTermOn

13.16.2.74 double pulsar::gwecc_ra

13.16.2.75 double pulsar::gwecc_redshift

13.16.2.76 double pulsar::gwecc_theta_0

13.16.2.77 double pulsar::gwecc_theta_nodes

13.16.2.78 double pulsar::gwm_decj

13.16.2.79 double pulsar::gwm_dphase

13.16.2.80 double pulsar::gwm_epoch

13.16.2.81 double pulsar::gwm_phi

13.16.2.82 double pulsar::gwm_raj

13.16.2.83 double pulsar::gwsrc_across_i

13.16.2.84 double pulsar::gwsrc_across_i_e

13.16.2.85 double pulsar::gwsrc_across_r

13.16.2.86 double pulsar::gwsrc_across_r_e

13.16.2.87 double pulsar::gwsrc_aplus_i

13.16.2.88 double pulsar::gwsrc_aplus_i_e

13.16.2.89 double pulsar::gwsrc_aplus_r

13.16.2.90 double pulsar::gwsrc_aplus_r_e

13.16.2.91 double pulsar::gwsrc_dec

13.16.2.92 double pulsar::gwsrc_epoch

13.16.2.93 double pulsar::gwsrc_psrdist

13.16.2.94 double pulsar::gwsrc_ra

13.16.2.95 double pulsar::ifunc_weights[MAX_IFUNC]

13.16.2.96 double pulsar::ifuncE[MAX_IFUNC]

13.16.2.97 int pulsar::ifuncN

13.16.2.98 double pulsar::ifuncT[MAX_IFUNC]

13.16.2.99 double pulsar::ifuncV[MAX_IFUNC]

13.16.2.100 int pulsar::ipm

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

13.16.2.101 int pulsar::jboFormat

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

13.16.2.102 char pulsar::JPL_EPHEMERIS[MAX_FILELEN]

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

String describing jump

13.16.2.104 double pulsar::jumpVal[MAX_JUMPS]

Value of jump

13.16.2.105 double pulsar::jumpValErr[MAX_JUMPS]

Error on jump

13.16.2.106 char pulsar::name[100]

13.16.2.107 int pulsar::nCompanion

Number of binary companions

13.16.2.108 int pulsar::nconstraints

Number of fit constraints specified

13.16.2.109 int pulsar::nDMEvents

13.16.2.110 int pulsar::ndmx

Number of DM steps

13.16.2.111 double pulsar::ne_sw

Electron density at 1AU due to the solar wind

13.16.2.112 int pulsar::nFit

Number of points in the fit

13.16.2.113 int pulsar::nGlobal

Number of global parameters in the fit

13.16.2.114 int pulsar::nits

Number of iterations for the fit

13.16.2.115 int pulsar::nJumps

Number of jumps

13.16.2.116 int pulsar::nobs

Number of observations in .tim file

13.16.2.117 int pulsar::noWarnings

= 1, do not display warning messages

13.16.2.118 `int pulsar::nParam`

Number of parameters in the fit

13.16.2.119 `int pulsar::nPhaseJump`

Number of phase jumps

13.16.2.120 `int pulsar::nQuad`

13.16.2.121 `int pulsar::nStorePrecision`

13.16.2.122 `int pulsar::nT2efac`

13.16.2.123 `int pulsar::nT2equad`

13.16.2.124 `int pulsar::nTelDX`

13.16.2.125 `int pulsar::nTelDY`

13.16.2.126 `int pulsar::nTelDZ`

13.16.2.127 `int pulsar::nTNBandNoise`

13.16.2.128 `int pulsar::nTNECORR`

13.16.2.129 `int pulsar::nTNEF`

13.16.2.130 `int pulsar::nTNEQ`

13.16.2.131 `int pulsar::nTNGroupNoise`

13.16.2.132 `int pulsar::nTNShapeletEvents`

13.16.2.133 `int pulsar::nTNSQ`

13.16.2.134 `int pulsar::nToffset`

13.16.2.135 `int pulsar::nWhite`

13.16.2.136 `int pulsar::nWhite_dm`

13.16.2.137 `observation* pulsar::obsn`

[MAX_OBSN_VAL];

13.16.2.138 `double pulsar::offset`

Offset, always fitted for

13.16.2.139 `double pulsar::offset_e`

Error in the offset

13.16.2.140 `int pulsar::outputTMatrix`

13.16.2.141 `parameter pulsar::param[MAX_PARAMS]`

13.16.2.142 `char pulsar::passStr[MAX_STRLEN]`

String describing filters

13.16.2.143 `longdouble pulsar::phaseJump[MAX_JUMPS]`

Time of phase jump

13.16.2.144 `int pulsar::phaseJumpDir[MAX_JUMPS]`

Size and direction of phase jump

13.16.2.145 `int pulsar::phaseJumpID[MAX_JUMPS]`

ID of closest point to the phase jump

13.16.2.146 `int pulsar::planetShapiro`

= 1 if included otherwise 0

13.16.2.147 `double pulsar::posPulsar[3]`

3-vector pointing at pulsar

13.16.2.148 `double pulsar::quad_across_i[MAX_QUAD]`

13.16.2.149 `double pulsar::quad_across_i_e[MAX_QUAD]`

13.16.2.150 `double pulsar::quad_across_r[MAX_QUAD]`

13.16.2.151 `double pulsar::quad_across_r_e[MAX_QUAD]`

13.16.2.152 `double pulsar::quad_aplus_i[MAX_QUAD]`

13.16.2.153 `double pulsar::quad_aplus_i_e[MAX_QUAD]`

13.16.2.154 `double pulsar::quad_aplus_r[MAX_QUAD]`

13.16.2.155 `double pulsar::quad_aplus_r_e[MAX_QUAD]`

13.16.2.156 `double pulsar::quad_ifunc_c_DEC`

13.16.2.157 `double pulsar::quad_ifunc_c_RA`

13.16.2.158 `double pulsar::quad_ifunc_geom_c`

13.16.2.159 `double pulsar::quad_ifunc_geom_p`

13.16.2.160 double pulsar::quad_ifunc_p_DEC

13.16.2.161 double pulsar::quad_ifunc_p_RA

13.16.2.162 double pulsar::quad_ifuncE_c[MAX_IFUNC]

13.16.2.163 double pulsar::quad_ifuncE_p[MAX_IFUNC]

13.16.2.164 int pulsar::quad_ifuncN_c

13.16.2.165 int pulsar::quad_ifuncN_p

13.16.2.166 double pulsar::quad_ifuncT_c[MAX_IFUNC]

13.16.2.167 double pulsar::quad_ifuncT_p[MAX_IFUNC]

13.16.2.168 double pulsar::quad_ifuncV_c[MAX_IFUNC]

13.16.2.169 double pulsar::quad_ifuncV_p[MAX_IFUNC]

13.16.2.170 double pulsar::quadDEC

13.16.2.171 double pulsar::quadEpoch

13.16.2.172 double pulsar::quadRA

13.16.2.173 char pulsar::rajStrPost[100]

13.16.2.174 char pulsar::rajStrPre[100]

13.16.2.175 double pulsar::rasim

13.16.2.176 int pulsar::rescaleErrChisq

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

13.16.2.177 double pulsar::rmsPost

13.16.2.178 double pulsar::rmsPre

13.16.2.179 char pulsar::robust

13.16.2.180 int pulsar::setTelVelX

13.16.2.181 int pulsar::setTelVelY

13.16.2.182 int pulsar::setTelVelZ

13.16.2.183 int pulsar::setUnits

13.16.2.184 int pulsar::simflag

Which fit function are we using

13.16.2.185 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];

13.16.2.186 storePrecision pulsar::storePrec[MAX_STOREPRECISION]

13.16.2.187 int pulsar::swm

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

13.16.2.188 int pulsar::t2cMethod

How to transform from terrestrial to celestial coords

13.16.2.189 char pulsar::T2efacFlagID[MAX_T2EFAC][MAX_FLAG_LEN]

13.16.2.190 char pulsar::T2efacFlagVal[MAX_T2EFAC][MAX_FLAG_LEN]

13.16.2.191 double pulsar::T2efacVal[MAX_T2EFAC]

13.16.2.192 char pulsar::T2equadFlagID[MAX_T2EQUAD][MAX_FLAG_LEN]

13.16.2.193 char pulsar::T2equadFlagVal[MAX_T2EQUAD][MAX_FLAG_LEN]

13.16.2.194 double pulsar::T2equadVal[MAX_T2EQUAD]

13.16.2.195 double pulsar::T2globalEfac

13.16.2.196 double pulsar::telDX_e[MAX_TEL_DX]

13.16.2.197 double pulsar::telDX_t[MAX_TEL_DX]

13.16.2.198 double pulsar::telDX_v[MAX_TEL_DX]

13.16.2.199 double pulsar::telDX_vel[MAX_TEL_DX]

13.16.2.200 double pulsar::telDX_vel_e[MAX_TEL_DX]

13.16.2.201 double pulsar::telDY_e[MAX_TEL_DY]

13.16.2.202 double pulsar::telDY_t[MAX_TEL_DY]

13.16.2.203 double pulsar::telDY_v[MAX_TEL_DY]

13.16.2.204 double pulsar::telDY_vel[MAX_TEL_DY]

13.16.2.205 double pulsar::telDY_vel_e[MAX_TEL_DY]

13.16.2.206 double pulsar::telDZ_e[MAX_TEL_DZ]

13.16.2.207 double pulsar::telDZ_t[MAX_TEL_DZ]

13.16.2.208 double pulsar::telDZ_v[MAX_TEL_DZ]

13.16.2.209 double pulsar::telDZ_vel[MAX_TEL_DZ]

13.16.2.210 double pulsar::telDZ_vel_e[MAX_TEL_DZ]

13.16.2.211 int pulsar::tempo1

= 1 if tempo1 is emulated

13.16.2.212 int pulsar::timeEphemeris

Which code to use for Einstein delay

13.16.2.213 double pulsar::TNBandDMAmp

13.16.2.214 int pulsar::TNBandDMC

13.16.2.215 double pulsar::TNBandDMGam

13.16.2.216 double pulsar::TNBandNoiseAmp[MAX_TNBN]

13.16.2.217 int pulsar::TNBandNoiseC[MAX_TNBN]

13.16.2.218 double pulsar::TNBandNoiseGam[MAX_TNBN]

13.16.2.219 double pulsar::TNBandNoiseHF[MAX_TNBN]

13.16.2.220 double pulsar::TNBandNoiseLF[MAX_TNBN]

13.16.2.221 double pulsar::TNDMAmp

13.16.2.222 int pulsar::TNDMC

13.16.2.223 double pulsar::TNDMCoeffs[200]

13.16.2.224 double pulsar::TNDMEvAmp[MAX_TNDMEv]

13.16.2.225 double pulsar::TNDMEvGam[MAX_TNDMEv]

13.16.2.226 double pulsar::TNDMEvLength[MAX_TNDMEv]

13.16.2.227 int pulsar::TNDMEvLin[MAX_TNDMEv]

13.16.2.228 int pulsar::TNDMEvOff[MAX_TNDMEv]

13.16.2.229 int pulsar::TNDMEvQuad[MAX_TNDMEv]

13.16.2.230 double pulsar::TNDMEvStart[MAX_TNDMEv]

13.16.2.231 double pulsar::TNDMGam

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

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

13.16.2.234 double pulsar::TNECORRVal[MAX_TNECORR]

13.16.2.235 char pulsar::TNEFFlagID[MAX_TNEF][MAX_FLAG_LEN]
13.16.2.236 char pulsar::TNEFFlagVal[MAX_TNEF][MAX_FLAG_LEN]
13.16.2.237 double pulsar::TNEFVal[MAX_TNEF]
13.16.2.238 char pulsar::TNEQFlagID[MAX_TNEQ][MAX_FLAG_LEN]
13.16.2.239 char pulsar::TNEQFlagVal[MAX_TNEQ][MAX_FLAG_LEN]
13.16.2.240 double pulsar::TNEQVal[MAX_TNEQ]
13.16.2.241 double pulsar::TNGlobalEF
13.16.2.242 double pulsar::TNGlobalEQ
13.16.2.243 double pulsar::TNGroupNoiseAmp[MAX_TNGN]
13.16.2.244 int pulsar::TNGroupNoiseC[MAX_TNGN]
13.16.2.245 char pulsar::TNGroupNoiseFlagID[MAX_TNGN][MAX_FLAG_LEN]
13.16.2.246 char pulsar::TNGroupNoiseFlagVal[MAX_TNGN][MAX_FLAG_LEN]
13.16.2.247 double pulsar::TNGroupNoiseGam[MAX_TNGN]
13.16.2.248 double pulsar::TNRedAmp
13.16.2.249 int pulsar::TNRedC
13.16.2.250 double pulsar::TNRedCoeffs[200]
13.16.2.251 double pulsar::TNRedCorner
13.16.2.252 double pulsar::TNRedFlow
13.16.2.253 double pulsar::TNRedGam
13.16.2.254 double pulsar::TNShapeletEvFScale[MAX_TNDMEv]
13.16.2.255 int pulsar::TNShapeletEvN[MAX_TNDMEv]
13.16.2.256 double pulsar::TNShapeletEvPos[MAX_TNDMEv]
13.16.2.257 double pulsar::TNShapeletEvWidth[MAX_TNDMEv]
13.16.2.258 char pulsar::TNSQFlagID[MAX_TNSQ][MAX_FLAG_LEN]
13.16.2.259 char pulsar::TNSQFlagVal[MAX_TNSQ][MAX_FLAG_LEN]
13.16.2.260 double pulsar::TNSQVal[MAX_TNSQ]
13.16.2.261 int pulsar::TNsubtractDM
13.16.2.262 int pulsar::TNsubtractRed

13.16.2.263 `double** pulsar::ToAextraCovar`

13.16.2.264 `double pulsar::tOffset[MAX_TOFFSET]`

Offsets in TOAs in seconds

13.16.2.265 `double pulsar::tOffset_f1[MAX_TOFFSET]`

13.16.2.266 `double pulsar::tOffset_f2[MAX_TOFFSET]`

Range for offset to be applied

13.16.2.267 `double pulsar::tOffset_t1[MAX_TOFFSET]`

13.16.2.268 `double pulsar::tOffset_t2[MAX_TOFFSET]`

13.16.2.269 `char pulsar::tOffsetFlags[MAX_TOFFSET][1000]`

13.16.2.270 `char pulsar::tOffsetSite[MAX_TOFFSET][100]`

13.16.2.271 `char pulsar::tzrsite[100]`

Site-code for polyco

13.16.2.272 `int pulsar::units`

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

13.16.2.273 `int pulsar::useCalceph`

13.16.2.274 `int pulsar::useTNorth`

13.16.2.275 `double pulsar::velPulsar[3]`

3-vector giving pulsar's velocity

13.16.2.276 `double pulsar::wave_cos[MAX_WHITE]`

13.16.2.277 `double pulsar::wave_cos_dm[MAX_WHITE]`

13.16.2.278 `double pulsar::wave_cos_dm_err[MAX_WHITE]`

13.16.2.279 `double pulsar::wave_cos_err[MAX_WHITE]`

13.16.2.280 `double pulsar::wave_sine[MAX_WHITE]`

13.16.2.281 `double pulsar::wave_sine_dm[MAX_WHITE]`

13.16.2.282 `double pulsar::wave_sine_dm_err[MAX_WHITE]`

13.16.2.283 `double pulsar::wave_sine_err[MAX_WHITE]`

13.16.2.284 `double pulsar::waveScale`

13.16.2.285 char pulsar::whiteNoiseModelFile[MAX_STRLEN]

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

- [tempo2.h](#)

13.17 storePrecision Struct Reference

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

Public Attributes

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

13.17.1 Member Data Documentation

13.17.1.1 char storePrecision::comment[MAX_STRLEN]

13.17.1.2 longdouble storePrecision::minPrec

13.17.1.3 char storePrecision::routine[100]

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

- [tempo2.h](#)

13.18 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]

13.18.1 Member Data Documentation

- 13.18.1.1 double T1Polyco::binary_frequency
- 13.18.1.2 double T1Polyco::binary_phase
- 13.18.1.3 long double T1Polyco::coeff[32]
- 13.18.1.4 char T1Polyco::date_string[10]
- 13.18.1.5 double T1Polyco::dm
- 13.18.1.6 double T1Polyco::doppler
- 13.18.1.7 double T1Polyco::frequency_obs
- 13.18.1.8 long double T1Polyco::frequency_psr_0
- 13.18.1.9 double T1Polyco::log10rms
- 13.18.1.10 long double T1Polyco::mjd_mid
- 13.18.1.11 int T1Polyco::ncoeff
- 13.18.1.12 char T1Polyco::psrname[64]
- 13.18.1.13 long double T1Polyco::reference_phase
- 13.18.1.14 char T1Polyco::sitename[5]
- 13.18.1.15 int T1Polyco::span
- 13.18.1.16 char T1Polyco::utc_string[13]

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

- [tempo2pred.h](#)

13.19 T1PolycoSet Struct Reference

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

Collaboration diagram for T1PolycoSet:

Public Attributes

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

13.19.1 Member Data Documentation

- 13.19.1.1 int T1PolycoSet::nsegments

13.19.1.2 T1Polyco* T1PolycoSet::segments

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

- [tempo2pred.h](#)

13.20 T2Predictor Struct Reference

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

Collaboration diagram for T2Predictor:

Public Attributes

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

13.20.1 Member Data Documentation

13.20.1.1 ChebyModelSet T2Predictor::cheby

13.20.1.2 T2PredictorKind T2Predictor::kind

13.20.1.3 union { ... } T2Predictor::modelset

13.20.1.4 T1PolycoSet T2Predictor::t1

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

- [tempo2pred.h](#)

13.21 TabulatedFunction Struct Reference

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

Collaboration diagram for TabulatedFunction:

Public Attributes

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

13.21.1 Member Data Documentation

13.21.1.1 char TabulatedFunction::fileName[256]

13.21.1.2 char TabulatedFunction::header_line[256]

13.21.1.3 **DynamicArray** TabulatedFunction::samples

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

- [tabulatedfunction.h](#)

13.22 TabulatedFunctionSample Struct Reference

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

Public Attributes

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

13.22.1 Member Data Documentation

13.22.1.1 double TabulatedFunctionSample::x

13.22.1.2 double TabulatedFunctionSample::y

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

- [tabulatedfunction.h](#)

Chapter 14

File Documentation

14.1 cholesky.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)

14.1.1 Function Documentation

- 14.1.1.1 void [cholesky_covarFunc2matrix](#) (double ** *m*, double * *covarFunc*, int *ndays*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 14.1.1.2 void [cholesky_dmModel](#) (double ** *m*, double *D*, double *d*, double *ref_freq*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 14.1.1.3 void [cholesky_dmModelCovarParam](#) (double ** *m*, double *alpha*, double *a*, double *b*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 14.1.1.4 void [cholesky_ecm](#) (double ** *m*, char * *fileName*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 14.1.1.5 int [cholesky_formUinv](#) (double ** *uinv*, double ** *m*, int *np*)
- 14.1.1.6 void [cholesky_powerlawModel](#) (double ** *m*, double *modelAlpha*, double *modelFc*, double *modelA*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)
- 14.1.1.7 void [cholesky_powerlawModel_withBeta](#) (double ** *m*, double *modelAlpha*, double *beta*, double *modelFc*, double *modelA*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)

14.1.1.8 void cholesky_readFromCovarianceFunction (double ** *m*, const char * *fname*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*)

14.2 choleskyRoutines.h File Reference

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

Include dependency graph for choleskyRoutines.h:

14.3 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_CFITSIO](#) 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 2016.08.0"
- #define [PACKAGE_TARNAME](#) "tempo2"
- #define [PACKAGE_URL](#) "http://www.bitbucket.org/psrsoft/tempo2"
- #define [PACKAGE_VERSION](#) "2016.08.0"
- #define [STDC_HEADERS](#) 1
- #define [TEMPO2_ARCH](#) "darwin15"
- #define [VERSION](#) "2016.08.0"
- #define [_DARWIN_USE_64_BIT_INODE](#) 1

14.3.1 Macro Definition Documentation

14.3.1.1 `#define _DARWIN_USE_64_BIT_INODE 1`

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

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

14.3.1.4 `#define HAVE_BLAS 1`

14.3.1.5 `#define HAVE_CFITSIO 1`

14.3.1.6 `#define HAVE_DLERROR 1`

14.3.1.7 `#define HAVE_DLFCN_H 1`

14.3.1.8 `#define HAVE_FFTW3 1`

14.3.1.9 `#define HAVE_INTTYPES_H 1`

14.3.1.10 `#define HAVE_LAPACK 1`

14.3.1.11 `#define HAVE_LIBDL 1`

14.3.1.12 `#define HAVE_LIBDLLOADER 1`

14.3.1.13 `#define HAVE_LIBM 1`

14.3.1.14 `#define HAVE_MEMORY_H 1`

14.3.1.15 `#define HAVE_PGPLOT 1`

14.3.1.16 `#define HAVE_PTHREAD 1`

14.3.1.17 `#define HAVE_STDINT_H 1`

14.3.1.18 `#define HAVE_STDLIB_H 1`

14.3.1.19 `#define HAVE_STRING_H 1`

14.3.1.20 `#define HAVE_STRINGS_H 1`

14.3.1.21 `#define HAVE_SYS_STAT_H 1`

14.3.1.22 `#define HAVE_SYS_TYPES_H 1`

14.3.1.23 `#define HAVE_UNISTD_H 1`

14.3.1.24 `#define LT_OBJDIR ".libs/"`

14.3.1.25 `#define PACKAGE "tempo2"`

14.3.1.26 `#define PACKAGE_BUGREPORT "george.hobbs@csiro.au"`

14.3.1.27 `#define PACKAGE_NAME "Tempo2"`

```

14.3.1.28 #define PACKAGE_STRING "Tempo2 2016.08.0"

14.3.1.29 #define PACKAGE_TARNAME "tempo2"

14.3.1.30 #define PACKAGE_URL "http://www.bitbucket.org/psrsoft/tempo2"

14.3.1.31 #define PACKAGE_VERSION "2016.08.0"

14.3.1.32 #define STDC_HEADERS 1

14.3.1.33 #define TEMPO2_ARCH "darwin15"

14.3.1.34 #define VERSION "2016.08.0"

```

14.4 constraints.h File Reference

```

#include <string.h>
#include "tempo2.h"
Include dependency graph for constraints.h:

```

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)

14.4.1 Function Documentation

```

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

14.4.1.2 void computeConstraintWeights ( pulsar * psr )

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

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

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

```

- 14.4.1.6 `double consFunc_dmmodel_mean (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.7 `double consFunc_ifunc (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.8 `double consFunc_ifunc_year (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.9 `double consFunc_qifunc_c_year (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.10 `double consFunc_qifunc_p_year (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.11 `double consFunc_quad_ifunc_c (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.12 `double consFunc_quad_ifunc_p (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.13 `double consFunc_tel_dx (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.14 `double consFunc_tel_dy (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.15 `double consFunc_tel_dz (pulsar * psr, int ipsr, int i, int k, int order)`
- 14.4.1.16 `void CONSTRAINTfuncs (pulsar * psr, int ipsr, int nparams, int iconstraint, double * OUT)`
- 14.4.1.17 `std::string get_constraint_name (enum constraint c)`
- 14.4.1.18 `double standardConstraintFunctions (pulsar * psr, int ipsr, int iconstraint, int iparam, int constraintk, int k)`

14.5 documentation/1_USER_GUIDE.md File Reference

14.6 documentation/2_developers.md File Reference

14.7 documentation/3_DEVELOPER_GUIDE.md File Reference

14.8 documentation/4_directories.md File Reference

14.9 documentation/5_plugins.md File Reference

14.10 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](#) *)

14.10.1 Function Documentation

14.10.1.1 void `DynamicArray_free` (`DynamicArray *`)

14.10.1.2 void `DynamicArray_init` (`DynamicArray *`, `size_t elemSize`)

14.10.1.3 void* `DynamicArray_push_back` (`DynamicArray *`, void * *elem*)

14.10.1.4 void `DynamicArray_resize` (`DynamicArray *`, `size_t nelem`)

14.11 enum_str.h File Reference

Variables

- const char * `label_str` []
- const char * `constraint_str` []

14.11.1 Variable Documentation

14.11.1.1 const char* `constraint_str` []

14.11.1.2 const char* `label_str` []

14.12 GWsim.h File Reference

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

Include dependency graph for GWsim.h:

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 `dt dt` (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)

14.12.1 Typedef Documentation

14.12.1.1 typedef struct `gwgeneralSrc` `gwgeneralSrc`

14.12.1.2 typedef struct `gwgenSpec` `gwgenSpec`

14.12.1.3 typedef struct `gwSrc` `gwSrc`

14.12.2 Function Documentation

14.12.2.1 `longdouble` `calculateResidualgeneralGW` (`longdouble` * *kp*, `gwgeneralSrc` * *gw*, `longdouble` *time*, `longdouble` *dist*)

14.12.2.2 `longdouble` `calculateResidualGW` (`longdouble` * *kp*, `gwSrc` * *gw*, `longdouble` *time*, `longdouble` *dist*)

14.12.2.3 double `dadt` (double *ec*, double *a*, double *m1*, double *m2*)

14.12.2.4 double `dedt` (double *ec*, double *a*, double *m1*, double *m2*)

14.12.2.5 `longdouble` `dotProduct` (`longdouble` * *m1*, `longdouble` * *m2*)

14.12.2.6 double `dt dt` (double *ec*, double *t*, double *p*)

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

14.12.2.8 `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*)

14.12.2.9 double `Fe` (double *ec*)

- 14.12.2.10 `double Findphi (double prob, double amp, double phase)`
- 14.12.2.11 `void GWanisotropicbackground (gwSrc * gw, int numberGW, long * idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double ** harmlist, int nharms)`
- 14.12.2.12 `void GWbackground (gwSrc * gw, int numberGW, long * idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin)`
- 14.12.2.13 `int GWbackground_read (gwSrc * gw, FILE * file, int ireal)`
- 14.12.2.14 `void GWbackground_write (gwSrc * gw, FILE * file, int ngw, int ireal)`
- 14.12.2.15 `void GWdipolebackground (gwSrc * gw, int numberGW, long * idum, longdouble flo, longdouble fhi, double gwAmp, double alpha, int loglin, double * dipoleamps)`
- 14.12.2.16 `void GWgeneralanisotropicbackground (gwgeneralSrc * gw, int * numberGW, long * idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin, double *** harmlist, int * nharms)`
- 14.12.2.17 `void GWgeneralbackground (gwgeneralSrc * gw, int * numberGW, long * idum, longdouble flo, longdouble fhi, gwgenSpec gwAmps, int loglin)`
- 14.12.2.18 `int GWgeneralbackground_read (gwgeneralSrc * gw, FILE * file, int ireal)`
- 14.12.2.19 `void GWgeneralbackground_write (gwgeneralSrc * gw, FILE * file, int ngw, int ireal)`
- 14.12.2.20 `void matrixMult (longdouble m1[3][3], longdouble m2[3][3], longdouble out[3][3])`
- 14.12.2.21 `double psrangle (double centre_long, double centre_lat, double psr_long, double psr_lat)`
- 14.12.2.22 `double Rs (double m1)`
- 14.12.2.23 `void setupgeneralGW (gwgeneralSrc * gw)`
- 14.12.2.24 `void setupGW (gwSrc * gw)`
- 14.12.2.25 `void setupPulsar_GWsim (longdouble ra_p, longdouble dec_p, longdouble * kp)`
- 14.12.2.26 `double sphharm (int l, int m, double x)`

14.13 ifteph.h File Reference

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

Include dependency graph for ifteph.h:

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)

14.13.1 Macro Definition Documentation

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

14.13.2 Function Documentation

- 14.13.2.1 void [IFTE_close_file](#) ()
- 14.13.2.2 double [IFTE_DeltaT](#) (double *Teph0*, double *Teph1*)
- 14.13.2.3 double [IFTE_DeltaTDot](#) (double *Teph0*, double *Teph1*)
- 14.13.2.4 void [IFTE_get_DeltaT_DeltaTDot](#) (double *Teph0*, double *Teph1*, double * *DeltaT*, double * *DeltaTDot*)
- 14.13.2.5 void [IFTE_get_vE](#) (double *Teph0*, double *Teph1*, double * *vE*)
- 14.13.2.6 void [IFTE_get_vE_vEDot](#) (double *Teph0*, double *Teph1*, double * *ve*, double * *vEDot*)
- 14.13.2.7 void [IFTE_get_vEDot](#) (double *Teph0*, double *Teph1*, double * *vEDot*)
- 14.13.2.8 void [IFTE_init](#) (const char * *fname*)

14.14 jpl_int.h File Reference

Classes

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

Macros

- `#define JPL_HEADER_SIZE (5 * sizeof(double) + 41 * sizeof(int32_t))`
- `#define MAX_CHEBY 18`

14.14.1 Macro Definition Documentation

14.14.1.1 `#define JPL_HEADER_SIZE (5 * sizeof(double) + 41 * sizeof(int32_t))`

14.14.1.2 `#define MAX_CHEBY 18`

14.15 jpleph.h File Reference

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_IPT_ARRAY 44`
- `#define JPL_EPHEM_EPHEMERIS_VERSION 224`
- `#define JPL_EPHEM_KERNEL_SIZE 228`
- `#define JPL_EPHEM_KERNEL_RECORD_SIZE 232`
- `#define JPL_EPHEM_KERNEL_NCOEFF 236`
- `#define JPL_EPHEM_KERNEL_SWAP_BYTES 240`
- `#define JPL_EPH_OUTSIDE_RANGE (-1)`
- `#define JPL_EPH_READ_ERROR (-2)`
- `#define JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS (-3)`
- `#define JPL_EPH_INVALID_INDEX (-5)`
- `#define JPL_EPH_FSEEK_ERROR (-6)`
- `#define JPL_INIT_NO_ERROR 0`
- `#define JPL_INIT_FILE_NOT_FOUND -1`
- `#define JPL_INIT_FSEEK_FAILED -2`
- `#define JPL_INIT_FREAD_FAILED -3`
- `#define JPL_INIT_FREAD2_FAILED -4`
- `#define JPL_INIT_FREAD5_FAILED -10`
- `#define JPL_INIT_FILE_CORRUPT -5`
- `#define JPL_INIT_MEMORY_FAILURE -6`
- `#define JPL_INIT_FREAD3_FAILED -7`
- `#define JPL_INIT_FREAD4_FAILED -8`
- `#define JPL_INIT_NOT_CALLED -9`
- `#define jpl_get_pvsun(ephem) ((double *))((char *)ephem + 248)`

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, const int list[14], double pv[][6], double nut[4], const int bary)`
- `int DLL_FUNC jpl_pleph (void *ephem, const double et, const int ntarg, const int ncent, double rrd[], const int calc_velocity)`
- `double DLL_FUNC jpl_get_double (const void *ephem, const int value)`
- `long DLL_FUNC jpl_get_long (const void *ephem, const int value)`
- `int DLL_FUNC make_sub_ephem (void *ephem, const char *sub_filename, const double start_jd, const double end_jd)`
- `double DLL_FUNC jpl_get_constant (const int idx, void *ephem, char *constant_name)`
- `int DLL_FUNC jpl_init_error_code (void)`

14.15.1 Macro Definition Documentation

- 14.15.1.1 `#define DLL_FUNC`
- 14.15.1.2 `#define JPL_EPH_FSEEK_ERROR (-6)`
- 14.15.1.3 `#define JPL_EPH_INVALID_INDEX (-5)`
- 14.15.1.4 `#define JPL_EPH_OUTSIDE_RANGE (-1)`
- 14.15.1.5 `#define JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS (-3)`
- 14.15.1.6 `#define JPL_EPH_READ_ERROR (-2)`
- 14.15.1.7 `#define JPL_EPHEM_AU_IN_KM 28`
- 14.15.1.8 `#define JPL_EPHEM_EARTH_MOON_RATIO 36`
- 14.15.1.9 `#define JPL_EPHEM_END_JD 8`
- 14.15.1.10 `#define JPL_EPHEM_EPHEMERIS_VERSION 224`
- 14.15.1.11 `#define JPL_EPHEM_IPT_ARRAY 44`
- 14.15.1.12 `#define JPL_EPHEM_KERNEL_NCOEFF 236`
- 14.15.1.13 `#define JPL_EPHEM_KERNEL_RECORD_SIZE 232`
- 14.15.1.14 `#define JPL_EPHEM_KERNEL_SIZE 228`
- 14.15.1.15 `#define JPL_EPHEM_KERNEL_SWAP_BYTES 240`
- 14.15.1.16 `#define JPL_EPHEM_N_CONSTANTS 24`
- 14.15.1.17 `#define JPL_EPHEM_START_JD 0`
- 14.15.1.18 `#define JPL_EPHEM_STEP 16`
- 14.15.1.19 `#define jpl_get_pvsun(ephem) ((double *))((char *)ephem + 248))`
- 14.15.1.20 `#define JPL_INIT_FILE_CORRUPT -5`
- 14.15.1.21 `#define JPL_INIT_FILE_NOT_FOUND -1`
- 14.15.1.22 `#define JPL_INIT_FREAD2_FAILED -4`
- 14.15.1.23 `#define JPL_INIT_FREAD3_FAILED -7`
- 14.15.1.24 `#define JPL_INIT_FREAD4_FAILED -8`
- 14.15.1.25 `#define JPL_INIT_FREAD5_FAILED -10`
- 14.15.1.26 `#define JPL_INIT_FREAD_FAILED -3`
- 14.15.1.27 `#define JPL_INIT_FSEEK_FAILED -2`

14.15.1.28 `#define JPL_INIT_MEMORY_FAILURE -6`

14.15.1.29 `#define JPL_INIT_NO_ERROR 0`

14.15.1.30 `#define JPL_INIT_NOT_CALLED -9`

14.15.2 Function Documentation

14.15.2.1 `void DLL_FUNC jpl_close_ephemeris (void * ephem)`

14.15.2.2 `double DLL_FUNC jpl_get_constant (const int idx, void * ephem, char * constant_name)`

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

14.15.2.4 `long DLL_FUNC jpl_get_long (const void * ephem, const int value)`

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

14.15.2.6 `int DLL_FUNC jpl_init_error_code (void)`

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

14.15.2.8 `int DLL_FUNC jpl_state (void * ephem, const double et, const int list[14], double pv[[6], double nut[4], const int bary)`

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

14.16 read_fortran.h File Reference

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

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

Include dependency graph for read_fortran.h:

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](#)

14.16.1 Function Documentation

14.16.1.1 void close_file ()

14.16.1.2 int open_file (char * *fname*)

14.16.1.3 char read_char ()

14.16.1.4 void read_character (int *len*, char * *str*)

14.16.1.5 double read_double ()

14.16.1.6 float read_float ()

14.16.1.7 int read_int ()

14.16.1.8 int read_record_int ()

14.16.2 Variable Documentation

14.16.2.1 FILE* c_fileptr

14.16.2.2 int swapByte

14.17 read_fortran2.h File Reference

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

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

Include dependency graph for read_fortran2.h:

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](#)

14.17.1 Function Documentation

14.17.1.1 void close_file2 ()

14.17.1.2 void open_file2 (char * *fname*, int * *swap*)

14.17.1.3 void read_character2 (int *len*, char * *str*)

14.17.1.4 `double read_double2 ()`

14.17.1.5 `float read_float2 ()`

14.17.1.6 `int read_int2 ()`

14.17.1.7 `int read_record_int2 ()`

14.17.2 Variable Documentation

14.17.2.1 `FILE* c_fileptr2`

14.17.2.2 `int swapByte2`

14.18 README.md File Reference

14.19 T2accel.h File Reference

`#include "config.h"`
Include dependency graph for T2accel.h:

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 **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`

14.19.1 Macro Definition Documentation

14.19.1.1 `#define ACCEL_LSQ`

14.19.1.2 `#define ACCEL_MULTMATRIX`

14.19.1.3 `#define ACCEL_UINV`

14.19.2 Function Documentation

14.19.2.1 `double accel_lsq_qr (double ** dm, double * data, double * oparm, int ndata, int nparam, double ** Ocvn)`

14.19.2.2 `void accel_multMatrix (double * m1, double * m2, int ndata, int ndata2, int npol, double * out)`

14.19.2.3 void accel_multMatrixVec (double * *m1*, double * *v*, int *ndata*, int *npol*, double * *out*)

14.19.2.4 int accel_uinv (double * *_m*, int *n*)

14.19.3 Variable Documentation

14.19.3.1 char useT2accel

14.20 t2fit.h File Reference

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

Include dependency graph for t2fit.h:

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, const [FitInfo](#) &globals)
- 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)

14.20.1 Function Documentation

14.20.1.1 void t2Fit (pulsar * *psr*, unsigned int *npsr*, const char * *covarFuncFile*)

14.20.1.2 void t2Fit_buildConstraintsMatrix (pulsar * *psr*, int *ipsr*, int *iconstraint*, double * *afunc*)

14.20.1.3 void t2Fit_buildDesignMatrix (pulsar * *psr*, int *ipsr*, double *x*, int *ipos*, double * *afunc*)

14.20.1.4 void t2Fit_fillFitInfo (pulsar * *psr*, [FitInfo](#) & *OUT*, const [FitInfo](#) & *globals*)

14.20.1.5 void t2Fit_fillGlobalFitInfo (pulsar * *psr*, unsigned int *npsr*, [FitInfo](#) & *OUT*)

14.20.1.6 unsigned int t2Fit_getFitData (pulsar * *psr*, double * *x*, double * *y*, double * *e*, int * *ip*)

14.20.1.7 void t2Fit_updateParameters (pulsar * *psr*, int *ipsr*, double * *val*, double * *error*)

14.21 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)

14.21.1 Function Documentation

14.21.1.1 double t2FitFunc_dmmodelCM (pulsar * *psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.21.1.2 double t2FitFunc_dmmodelDM (pulsar * *psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.21.1.3 void t2UpdateFunc_dmmodelCM (pulsar * *psr*, int *ip*sr, param_label *label*, int *k*, double *val*, double *err*)

14.21.1.4 void t2UpdateFunc_dmmodelDM (pulsar * *psr*, int *ip*sr, param_label *label*, int *k*, double *val*, double *err*)

14.22 t2fit_dmother.h File Reference

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

Include dependency graph for t2fit_dmother.h: This graph shows which files directly or indirectly include this file:

Functions

- double t2FitFunc_dmx (pulsar **psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)
- double t2FitFunc_dmsinusoids (pulsar **psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)
- double t2FitFunc_fd (pulsar **psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)
- double t2FitFunc_fddc (pulsar **psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.22.1 Function Documentation

14.22.1.1 double t2FitFunc_dmsinusoids (pulsar * *psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.22.1.2 double t2FitFunc_dmx (pulsar * *psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.22.1.3 double t2FitFunc_fd (pulsar * *psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.22.1.4 double t2FitFunc_fddc (pulsar * *psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.23 t2fit_fitwaves.h File Reference

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

Functions

- double t2FitFunc_fitwaves (pulsar **psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)
- void t2UpdateFunc_fitwaves (pulsar **psr*, int *ip*sr, param_label *label*, int *k*, double *val*, double *err*)

14.23.1 Function Documentation

14.23.1.1 double t2FitFunc_fitwaves (pulsar * *psr*, int *ip*sr, double *x*, int *ip*os, param_label *label*, int *k*)

14.23.1.2 void t2UpdateFunc_fitwaves (pulsar * *psr*, int *ip*sr, param_label *label*, int *k*, double *val*, double *err*)

14.24 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)

14.24.1 Function Documentation

14.24.1.1 double [t2FitFunc_stdGlitch](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label label](#), int *k*)

14.24.1.2 void [t2UpdateFunc_stdGlitch](#) ([pulsar](#) * *psr*, int *ipsr*, [param_label label](#), int *k*, double *val*, double *err*)

14.25 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 [sinfunc](#) (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)

14.25.1 Function Documentation

14.25.1.1 double [ifunc](#) (const double * *mjd*, const double *t*, const int *N*, const int *k*)

14.25.1.2 double [sinfunc](#) (const double * *T*, const double *t*, const int *k*)

14.25.1.3 double [t2FitFunc_ifunc](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label label](#), int *k*)

14.25.1.4 double [t2FitFunc_sifunc](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label label](#), int *k*)

14.25.1.5 void [t2UpdateFunc_ifunc](#) ([pulsar](#) * *psr*, int *ipsr*, [param_label label](#), int *k*, double *val*, double *err*)

14.26 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)

14.26.1 Function Documentation

14.26.1.1 double [t2FitFunc_stdPosition](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label label](#), int *k*)

14.26.1.2 void [t2UpdateFunc_stdPosition](#) ([pulsar](#) * *psr*, int *ipsr*, [param_label label](#), int *k*, double *val*, double *err*)

14.27 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_dmmmodel.h"
#include "t2fit_dmother.h"
Include dependency graph for t2fit_stdFitFuncs.h:
```

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 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)
- double [t2FitFunc_notImplemented](#) ([pulsar](#) *psr, int ipsr, double x, int ipos, [param_label](#) label, int k)
- void [t2UpdateFunc_notImplemented](#) ([pulsar](#) *psr, int ipsr, [param_label](#) label, int k, double val, double err)

14.27.1 Function Documentation

- 14.27.1.1 double [t2FitFunc_binaryModels](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.2 double [t2FitFunc_ifunc](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.3 double [t2FitFunc_jump](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.4 double [t2FitFunc_notImplemented](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.5 double [t2FitFunc_planet](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.6 double [t2FitFunc_stdDm](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.7 double [t2FitFunc_stdFreq](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.8 double [t2FitFunc_stdGravWav](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)
- 14.27.1.9 double [t2FitFunc_telPos](#) ([pulsar](#) * *psr*, int *ipsr*, double *x*, int *ipos*, [param_label](#) *label*, int *k*)

- 14.27.1.10 `double t2FitFunc_zero (pulsar * psr, int ipsr, double x, int ipos, param_label label, int k)`
- 14.27.1.11 `void t2UpdateFunc_binaryModels (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.12 `void t2UpdateFunc_ifunc (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.13 `void t2UpdateFunc_jump (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.14 `void t2UpdateFunc_notImplemented (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.15 `void t2UpdateFunc_planet (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.16 `void t2UpdateFunc_simpleAdd (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`
- 14.27.1.17 `void t2UpdateFunc_simpleMinus (pulsar * psr, int ipsr, param_label label, int k, double val, double error)`
- 14.27.1.18 `void t2UpdateFunc_stdFreq (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.19 `void t2UpdateFunc_stdGravWav (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.20 `void t2UpdateFunc_telPos (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`
- 14.27.1.21 `void t2UpdateFunc_zero (pulsar * psr, int ipsr, param_label label, int k, double val, double err)`

14.28 T2toolkit.h File Reference

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

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 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)

14.28.1 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

14.28.2 Function Documentation

- 14.28.2.1 unsigned long [genrand_int32](#) (void)
- 14.28.2.2 double [genrand_real1](#) (void)
- 14.28.2.3 void [init_genrand](#) (unsigned long s)
- 14.28.2.4 void [TKconvertFloat1](#) (double * x, float * ox, int n)
- 14.28.2.5 void [TKconvertFloat2](#) (double * x, double * y, float * ox, float * oy, int n)
- 14.28.2.6 double [TKfindMax_d](#) (double * x, int n)
- 14.28.2.7 float [TKfindMax_f](#) (float * x, int n)
- 14.28.2.8 double [TKfindMedian_d](#) (double * val, int count)
- 14.28.2.9 float [TKfindMedian_f](#) (float * val, int count)
- 14.28.2.10 double [TKfindMin_d](#) (double * x, int n)
- 14.28.2.11 float [TKfindMin_f](#) (float * x, int n)
- 14.28.2.12 double [TKfindRMS_d](#) (double * x, int n)
- 14.28.2.13 float [TKfindRMS_f](#) (float * x, int n)
- 14.28.2.14 float [TKfindRMSweight_d](#) (double * x, double * e, int n)
- 14.28.2.15 double [TKgaussDev](#) (long * seed)
- 14.28.2.16 double [TKmean_d](#) (double * x, int n)
- 14.28.2.17 float [TKmean_f](#) (float * x, int n)

14.28.2.18 double TKranDev (long * *seed*)

14.28.2.19 double TKrange_d (double * *x*, int *n*)

14.28.2.20 float TKrange_f (float * *x*, int *n*)

14.28.2.21 double TKretMax_d (double *a*, double *b*)

14.28.2.22 float TKretMax_f (float *a*, float *b*)

14.28.2.23 double TKretMin_d (double *a*, double *b*)

14.28.2.24 float TKretMin_f (float *a*, float *b*)

14.28.2.25 int TKretMin_i (int *a*, int *b*)

14.28.2.26 long TKsetSeed ()

14.28.2.27 double TKsign_d (double *a*, double *b*)

14.28.2.28 void TKsort_2f (float * *val*, float * *val2*, int *nobs*)

14.28.2.29 void TKsort_3d (double * *val*, double * *val2*, double * *val3*, int *nobs*)

14.28.2.30 void TKsort_d (double * *val*, int *nobs*)

14.28.2.31 void TKsort_f (float * *val*, int *nobs*)

14.28.2.32 double TKvariance_d (double * *x*, int *n*)

14.28.2.33 void TKzeromean_d (int *n*, double * *y*)

14.29 tabulatedfunction.h File Reference

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

Include dependency graph for tabulatedfunction.h:

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)

14.29.1 Function Documentation

14.29.1.1 double [TabulatedFunction_getEndX](#) ([TabulatedFunction](#) * *func*)

14.29.1.2 `double TabulatedFunction_getStartX (TabulatedFunction * func)`

14.29.1.3 `double TabulatedFunction_getValue (TabulatedFunction * func, double x)`

14.29.1.4 `void TabulatedFunction_load (TabulatedFunction * func, char * fileName)`

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

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: e7d4e3eed480ba2cff605f4ca7e53d2d6368dede $"`
- `#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 40
- #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 observation.
- 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_glt](#),
[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_dphaseplanet](#), [param_waveepoch](#),
[param_ifunc](#), [param_clk_offs](#), [param_dmx](#), [param_dmxr1](#),
[param_dmxr2](#), [param_dmmmodel](#), [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 [tt2tb_calceph](#) ([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_PTAmode](#) (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`

14.30.1 Detailed Description

contains the main interface to libtempo2.

Note

some parts of this to be moved to an internal interface

14.30.2 Macro Definition Documentation

14.30.2.1 `#define AU_DIST 1.49598e11`

1 AU in m

14.30.2.2 `#define AULTSC 499.00478364`

Number of light seconds in 1 AU

14.30.2.3 `#define BIG_G 6.673e-11`

Gravitational constant

14.30.2.4 `#define DM_CONST 2.41e-4`

14.30.2.5 `#define DM_CONST_SI 7.436e6`

Dispersion constant in SI units

14.30.2.6 `#define ECLIPTIC_OBLIQUITY_VAL 84381.4059`

mean obliquity of ecliptic in arcsec

14.30.2.7 `#define FB90_TIMEEPH 2`

Fairhead & Bretagnon time ephemeris

14.30.2.8 `#define GM 1.3271243999e20`

Gravitational constant * mass sun

14.30.2.9 `#define GM_C3 4.925490947e-6`

GM_{\odot}/c^3 (in seconds)

14.30.2.10 `#define GMJ_C3 4.70255e-9`

GM_{jupiter}/c^3 (in seconds)

14.30.2.11 `#define GMN_C3 2.54488e-10`

GM_{neptune}/c^3 (in seconds)

14.30.2.12 `#define GMS_C3 1.40797e-9`

GM_{saturn}/c^3 (in seconds)

14.30.2.13 `#define GMU_C3 2.14539e-10`

GM_{uranus}/c^3 (in seconds)

14.30.2.14 `#define GMV_C3 1.2061e-11`

GM_{venus}/c^3 (in seconds)

14.30.2.15 `#define HAVE_GWSIM_H`

14.30.2.16 `#define IF99_TIMEEPH 1`

Irwin & Fukushima time ephemeris

14.30.2.17 `#define IFTEPH_FILE "/ephemeris/TIMEEPH_short.te405"`

14.30.2.18 `#define LEAPSECOND_FILE "/clock/leap.sec"`

Path for the file containing dates when leap seconds should be added

14.30.2.19 `#define MASR2RADS 1.53628185e-16`

Converts from mas/yr to rad/s

14.30.2.20 `#define MAX_BPJ_JUMPS 5`

Maximum number of jumps in binary params - for BPJ model

14.30.2.21 `#define MAX_CLK_CORR 30`

Maximum number of steps in the correction to TT

14.30.2.22 `#define MAX_CLKCORR 5000`

Maximum number of lines in time.dat file

14.30.2.23 `#define MAX_COEFF 5000`

Maximum number of coefficients in polyco

14.30.2.24 `#define MAX_COMPANIONS 4`

Maximum number of binary companions

14.30.2.25 `#define MAX_DM_DERIVATIVES 10`

DM0 -> DMn where n=10

14.30.2.26 `#define MAX_DMx 512`

Max number of DM steps allowed

14.30.2.27 `#define MAX_FILELEN 500`

Maximum filename length

14.30.2.28 `#define MAX_FIT 10000`

Maximum number of parameters to fit for

14.30.2.29 `#define MAX_FLAG_LEN 32`

Maximum number of characters in each flag

14.30.2.30 `#define MAX_FLAGS 40`

Maximum number of flags in .tim file/observation

14.30.2.31 `#define MAX_FREQ_DERIVATIVES 13`

F0 -> Fn where n=10

14.30.2.32 `#define MAX_IFUNC 1000`

Maximum number of parameters for interpolation function

14.30.2.33 `#define MAX_JUMPS 2000`

Maximum number of phase jumps

14.30.2.34 `#define MAX_LEAPSEC 100`

Maximum number of line in the leap second file

14.30.2.35 `#define MAX_MSG 50`

Maximum number of different warnings

14.30.2.36 `#define MAX_OBSN_VAL 20000`

Maximum number of TOAs

14.30.2.37 `#define MAX_PARAMS 2000`

Maximum number of parameters

14.30.2.38 `#define MAX_PSR_VAL 40`

Maximum number of pulsars

14.30.2.39 `#define MAX_QUAD 150`

Maximum number of frequency channels in quadrupolar function

14.30.2.40 `#define MAX_SITE 100`

Maximum number of observatory sites

14.30.2.41 `#define MAX_STOREPRECISION 50`

How many routines in TEMPO2 store precision information

14.30.2.42 `#define MAX_STRLEN 1000`

Maximum length for strings

14.30.2.43 `#define MAX_T2EFAC 100`

Maximum number of T2EFACs allowed

14.30.2.44 `#define MAX_T2EQUAD 100`

Maximum number of T2EQUADs allowed

14.30.2.45 `#define MAX_TEL_CLK_OFFS 500`

Maximum number of parameters for telescope clock offset

14.30.2.46 `#define MAX_TEL_DX 500`

Maximum number of parameters for interpolation function

14.30.2.47 `#define MAX_TEL_DY 500`

Maximum number of parameters for interpolation function

14.30.2.48 `#define MAX_TEL_DZ 500`

Maximum number of parameters for interpolation function

14.30.2.49 `#define MAX_TNBN 50 /*maximum number of TNBandNoise parameters allowed*/`

14.30.2.50 `#define MAX_TNDMEv 10 /*Maximum number of TNDMEvents allowed */`

14.30.2.51 `#define MAX_TNECORR 50`

Maximum number of TNECORRss allowed

14.30.2.52 `#define MAX_TNEF 50`

Maximum number of TNEFACs allowed

14.30.2.53 `#define MAX_TNEQ 50`

Maximum number of TNEQUADs allowed

14.30.2.54 `#define MAX_TNGN 50`

maximum number of TNGroupNoise parameters allowed

14.30.2.55 `#define MAX_TNSQ 50`

Maximum number of TNEQUADs allowed

14.30.2.56 `#define MAX_TOFFSET 10`

Number of time jumps allowed in .par file

14.30.2.57 `#define MAX_WHITE 100`

Maximum number of parameters for whitening

14.30.2.58 `#define NE_SW_DEFAULT 4`

Default value for electron density (cm⁻³) at 1 AU due to solar wind

14.30.2.59 `#define OBLQ 23.445833333333333`

Obliquity of the ecliptic

14.30.2.60 `#define OBSSYS_FILE "/observatory/newobsys.dat"`

Path for file containing Observatory data (obsys.dat)

14.30.2.61 `#define PCM 3.08568025e16`

one parsec in meters

14.30.2.62 `#define SECDAY 86400.0`

Number of seconds in 1 day

14.30.2.63 `#define SECDAYI longdouble(86400.0)`

Number of seconds in 1 day

14.30.2.64 `#define SI_UNITS 1`

New tempo2 mode

14.30.2.65 `#define SOLAR_MASS 1.98892e30`

Mass of Sun (kg)

14.30.2.66 `#define SOLAR_RADIUS 6.96e8`

Radius of the Sun (in meters)

14.30.2.67 `#define SPEED_LIGHT 299792458.0`

Speed of light (m/s)

14.30.2.68 `#define T2C_IAU2000B 1`

14.30.2.69 `#define T2C_TEMPO 2`

14.30.2.70 `#define TDB_UNITS 2`

original tempo mode

14.30.2.71 `#define TDBTDT_FILE "/ephemeris/TDB.1950.2050"`

Path for file containing TDB-TDT ephemeris

14.30.2.72 `#define TEMPO2_h_HASH "$Id: e7d4e3eed480ba2cff605f4ca7e53d2d6368dede $"`

14.30.2.73 `#define TEMPO2_h_MAJOR_VER 2015.09`

14.30.2.74 `#define TEMPO2_h_MINOR_VER 0`

14.30.2.75 `#define TEMPO2_h_VER "2015.09.0"`

14.30.2.76 `#define TSUN longdouble(4.925490947e-6)`

Solar constant for mass calculations.

14.30.2.77 `#define UT1_FILE "/clock/ut1.dat"`

Path for the file containing TAI-UT1

14.30.3 Typedef Documentation

14.30.3.1 `typedef int constraint_label`

for 'strong typing' - type for enum constraint

14.30.3.2 `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.

14.30.3.3 `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.

14.30.3.4 `typedef struct observation observation`

A struct containing the details of a single observation.

14.30.3.5 `typedef int param_label`

for 'strong typing' - type for enum label

14.30.3.6 `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.

14.30.3.7 `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

14.30.3.8 `typedef void(* paramUpdateFunc) (struct pulsar *, int, param_label, int, double, double)`

a function used to update the parameters after a fit.

14.30.3.9 `typedef struct pulsar pulsar`

contains the details for a single pulsar.

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

14.30.3.10 `typedef struct storePrecision storePrecision`

14.30.4 Enumeration Type Documentation

14.30.4.1 `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

14.30.4.2 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_dphaseplanet
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

14.30.5 Function Documentation

- 14.30.5.1 void allocateMemory (*pulsar* * *psr*, int *realloc*)
- 14.30.5.2 void autoConstraints (*pulsar* * *psr*, int *ipsr*, int *npsr*)
- 14.30.5.3 int bootstrap (*pulsar* * *psr*, int *p*, int *npsr*)
- 14.30.5.4 double BTJmodel (*pulsar* * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 14.30.5.5 double BTmodel (*pulsar* * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.6 double BTXmodel (*pulsar* * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 14.30.5.7 double calcRMS (*pulsar* * *psr*, int *p*)
- 14.30.5.8 void calculate_bclt (*pulsar* * *psr*, int *npsr*)
- 14.30.5.9 void compute_tropospheric_delays (*pulsar* * *psr*, int *npsr*)
- 14.30.5.10 void copyParam (*parameter* *p1*, *parameter* * *p2*)
- 14.30.5.11 void copyPSR (*pulsar* * *p*, int *p1*, int *p2*)

- 14.30.5.12 void CVSdisplayVersion (const char * *file*, const char * *func*, const char * *verNum*)
- 14.30.5.13 double DDGRmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.14 double DDHmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.15 double DDKmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.16 longdouble DDmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.17 double DDSmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.18 void defineClockCorrectionSequence (char * *fileList*, int *dispWarnings*)
- 14.30.5.19 void destroyMemory (pulsar * *psr*)
- 14.30.5.20 void destroyOne (pulsar * *psr*)
- 14.30.5.21 void displayMsg (int *type*, const char * *key*, const char * *searchStr*, const char * *variableStr*, int *noWarnings*)
- 14.30.5.22 void displayParameters (int *pos*, char *timeFile*[][MAX_FILELEN], char *parFile*[][MAX_FILELEN], pulsar * *psr*, int *npsr*)
- 14.30.5.23 void dm_delays (pulsar * *psr*, int *npsr*, int *p*, int *i*, double *delt*, double *dt_SSB*)
- 14.30.5.24 double dms_turn (char * *line*)
- 14.30.5.25 void doFit (pulsar * *psr*, int *npsr*, int *writeModel*)
- 14.30.5.26 void doFitAll (pulsar * *psr*, int *npsr*, const char * *covarFuncFile*)
- 14.30.5.27 void doFitDCM (pulsar * *psr*, const char * *dcmFile*, const char * *covarFuncFile*, int *npsr*, int *writeModel*)
- 14.30.5.28 void doFitGlobal (pulsar * *psr*, int *npsr*, double * *globalParameter*, int *nGlobal*, int *writeModel*)
- 14.30.5.29 double dotproduct (double * *v1*, double * *v2*)
- 14.30.5.30 double ELL1Hmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.31 double ELL1model (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.32 void equ2ecl (double * *x*)
- 14.30.5.33 void FITfuncs (double *x*, double *afunc*[], int *ma*, pulsar * *psr*, int *ipos*, int *ipsr*)
- 14.30.5.34 void formBats (pulsar * *psr*, int *npsr*)
- 14.30.5.35 void formBatsAll (pulsar * *psr*, int *npsr*)
- 14.30.5.36 void formResiduals (pulsar * *psr*, int *npsr*, int *removeMean*)
- 14.30.5.37 longdouble fortran_mod (longdouble *a*, longdouble *p*)
- 14.30.5.38 int fortran_nint (double *x*)
- 14.30.5.39 long fortran_nlong (longdouble *x*)

- 14.30.5.40 void get_EOP (double *mjd*, double * *xp*, double * *yp*, double * *dut1*, double * *dut1dot*, int *dispWarnings*, char * *eopcFile*)
- 14.30.5.41 void get_obsCoord (pulsar * *psr*, int *npsr*)
- 14.30.5.42 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])
- 14.30.5.43 void get_OneobsCoord (pulsar * *psr*, int *npsr*, int *obs*)
- 14.30.5.44 void getCholeskyMatrix (double ** *uinv*, const char * *fname*, pulsar * *psr*, double * *resx*, double * *resy*, double * *rese*, int *np*, int *nc*, int * *ip*)
- 14.30.5.45 void getClockCorrections (observation * *obs*, const char * *clockFrom*, const char * *clockTo*, int *warnings*)
- 14.30.5.46 double getCorrection (observation * *obs*, const char * *clockFrom*, const char * *clockTo*, int *warnings*)
- 14.30.5.47 double getCorrectionTT (observation * *obs*)
- 14.30.5.48 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*)
- 14.30.5.49 observatory* getObservatory (char * *code*)
- 14.30.5.50 double getParamDeriv (pulsar * *psr*, int *ipos*, double *x*, int *i*, int *k*)
- 14.30.5.51 longdouble getParameterValue (pulsar * *psr*, int *param*, int *arr*)
- 14.30.5.52 double hms_turn (char * *line*)
- 14.30.5.53 int id_residual (float *xcurs*, float *ycurs*)
- 14.30.5.54 void initialise (pulsar * *psr*, int *noWarnings*)
- 14.30.5.55 void initialiseOne (pulsar * *psr*, int *noWarnings*, int *fullSetup*)
- 14.30.5.56 double JVmodel (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 14.30.5.57 void logicFlag (char * *line*, pulsar * *psr*, int *npsr*)
- 14.30.5.58 void lookup_observatory_alias (char * *incode*, char * *outcode*)
- 14.30.5.59 double MSSmodel (pulsar * *psr*, int *p*, int *obs*, int *param*)
- 14.30.5.60 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*)
- 14.30.5.61 void preProcess (pulsar * *psr*, int *npsr*, int *argc*, char * *argv*[])
- 14.30.5.62 void preProcessSimple (pulsar * *psr*)
- 14.30.5.63 void preProcessSimple1 (pulsar * *psr*, int *tempo1*, double *thelast*)

- 14.30.5.64 void preProcessSimple2 (pulsar * *psr*, float *startdmmjd*, int *ndm*, float * *dmvals*, int *trimonly*)
- 14.30.5.65 void preProcessSimple3 (pulsar * *psr*)
- 14.30.5.66 void processFlag (char * *line*, pulsar * *psr*, int *npsr*)
- 14.30.5.67 void processSimultaneous (char * *line*, pulsar * *psr*, int *npsr*)
- 14.30.5.68 void readEphemeris (pulsar * *psr*, int *npsr*, int *addEphemNoise*)
- 14.30.5.69 void readEphemeris_calceph (pulsar * *psr*, int *npsr*)
- 14.30.5.70 void readJBO_bat (char * *fname*, pulsar * *psr*, int *p*)
- 14.30.5.71 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])
- 14.30.5.72 void readOneEphemeris (pulsar * *psr*, int *npsr*, int *addEphemNoise*, int *obsNumber*)
- 14.30.5.73 void readParfile (pulsar * *psr*, char *parFile*[][MAX_FILELEN], char *timFile*[][MAX_FILELEN], int *npsr*)
- 14.30.5.74 void readParfileGlobal (pulsar * *psr*, int *npsr*, char *tpar*[MAX_STRLEN][MAX_FILELEN], char *ttim*[MAX_STRLEN][MAX_FILELEN])
- 14.30.5.75 int readSimpleParfile (FILE * *fin*, pulsar * *p*)
- 14.30.5.76 void readTimfile (pulsar * *psr*, char *timFile*[][MAX_FILELEN], int *npsr*)
- 14.30.5.77 void recordPrecision (pulsar * *psr*, longdouble *prec*, const char * *routine*, const char * *comment*)
- 14.30.5.78 void secularMotion (pulsar * *psr*, int *npsr*)
- 14.30.5.79 void setPlugPath ()
- 14.30.5.80 float setStart (float *xcurs*, float *ycurs*, int *flag*)
- 14.30.5.81 int setupParameterFileDefaults (pulsar * *p*)
- 14.30.5.82 void shapiro_delay (pulsar * *psr*, int *npsr*, int *p*, int *i*, double *delt*, double *dt_SSB*)
- 14.30.5.83 void simplePlot (pulsar * *psr*, double *unitFlag*)
- 14.30.5.84 double solarWindModel (pulsar *psr*, int *iobs*)
- 14.30.5.85 void sortToAs (pulsar * *psr*)
- 14.30.5.86 double T2_PTAmode1 (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 14.30.5.87 double T2model (pulsar * *psr*, int *p*, int *obs*, int *param*, int *arr*)
- 14.30.5.88 void tai2tt (pulsar * *psr*, int *npsr*)
- 14.30.5.89 void tai2ut1 (pulsar * *psr*, int *npsr*)

- 14.30.5.90 void textOutput (pulsar * *psr*, int *npsr*, double *globalParameter*, int *nGlobal*, int *outRes*, int *newpar*, const char * *fname*)
- 14.30.5.91 void toa2utc (pulsar * *psr*, int *npsr*)
- 14.30.5.92 void transform_units (struct pulsar * *psr*, int *from*, int *to*)
- 14.30.5.93 void tt2tb (pulsar * *psr*, int *npsr*)
- 14.30.5.94 void tt2tb_calceph (pulsar * *psr*, int *npsr*)
- 14.30.5.95 double turn_deg (double *turn*)
- 14.30.5.96 int turn_dms (double *turn*, char * *dms*)
- 14.30.5.97 int turn_hms (double *turn*, char * *hms*)
- 14.30.5.98 void updateBatsAll (pulsar * *psr*, int *npsr*)
- 14.30.5.99 void updateBT (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.100 void updateBTJ (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)
- 14.30.5.101 void updateBTX (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)
- 14.30.5.102 void updateDD (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.103 void updateDDGR (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.104 void updateDDH (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.105 void updateDDK (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.106 void updateDDS (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.107 void updateELL1 (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.108 void updateELL1H (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.109 void updateJV (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)
- 14.30.5.110 void updateMSS (pulsar * *psr*, double *val*, double *err*, int *pos*)
- 14.30.5.111 void updateParameters (pulsar * *psr*, int *p*, double * *val*, double * *error*)
- 14.30.5.112 void updateT2 (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)
- 14.30.5.113 void updateT2_PTA (pulsar * *psr*, double *val*, double *err*, int *pos*, int *arr*)
- 14.30.5.114 void useSelectFile (char * *fname*, pulsar * *psr*, int *npsr*)
- 14.30.5.115 void utc2tai (pulsar * *psr*, int *npsr*)
- 14.30.5.116 void vectorPulsar (pulsar * *psr*, int *npsr*)
- 14.30.5.117 void vectorscale (double * *v*, double *k*)

14.30.5.118 void vectorsum (double * *res*, double * *v1*, double * *v2*)

14.30.5.119 void writeTim (const char * *timname*, pulsar * *psr*, const char * *fileFormat*)

14.30.5.120 int zoom_graphics (float *xcurs2*, float *ycurs2*, int *flag*)

14.30.6 Variable Documentation

14.30.6.1 char covarFuncFile[MAX_FILELEN]

14.30.6.2 char dcmFile[MAX_FILELEN]

14.30.6.3 int displayCVSversion

Display CVS version

14.30.6.4 double ECLIPTIC_OBLIQUITY

14.30.6.5 int forceGlobalFit

Global = 1 if we are forcing a global fit

14.30.6.6 int MAX_OBSN

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

14.30.6.7 int MAX_PSR

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

14.30.6.8 char NEWFIT

global boolean used to enable new fit.

Warning

this will be removed in future.

14.30.6.9 char TEMPO2_ENVIRON[]

TEMPO2 environment variable

14.30.6.10 char TEMPO2_ERROR[]

TEMPO2 error messages

14.30.6.11 char tempo2_plug_path[32][MAX_STRLEN]

paths to search for plugins

14.30.6.12 int tempo2_plug_path_len

14.30.6.13 char tempo2MachineType[MAX_FILELEN]

14.30.6.14 int veryFast

Global to run the code fast

14.31 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](#)

14.31.1 Enumeration Type Documentation

14.31.1.1 enum T2PredictorKind

Enumerator

NonePredType

Cheby

T1

14.31.2 Function Documentation

14.31.2.1 void T2Predictor_Copy (T2Predictor * *into_t2p*, const T2Predictor * *from_t2p*)

14.31.2.2 void T2Predictor_Destroy (T2Predictor * *t2p*)

14.31.2.3 int T2Predictor_FRead (T2Predictor * *t2p*, FILE * *f*)

14.31.2.4 void T2Predictor_FWrite (const T2Predictor * *t2p*, FILE * *f*)

14.31.2.5 long double T2Predictor_GetEndFreq (T2Predictor * *t2p*)

14.31.2.6 long double T2Predictor_GetEndMJD (T2Predictor * *t2p*)

14.31.2.7 long double T2Predictor_GetFrequency (const T2Predictor * *t2p*, long double *mjd*, long double *freq*)

14.31.2.8 long double T2Predictor_GetPhase (const T2Predictor * *t2p*, long double *mjd*, long double *freq*)

14.31.2.9 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*)

14.31.2.10 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*)

14.31.2.11 char* T2Predictor_GetPSRName (T2Predictor * *t2p*)

14.31.2.12 char* T2Predictor_GetSiteName (T2Predictor * *t2p*)

14.31.2.13 long double T2Predictor_GetStartFreq (T2Predictor * *t2p*)

14.31.2.14 long double T2Predictor_GetStartMJD (T2Predictor * *t2p*)

14.31.2.15 void T2Predictor_Init (T2Predictor * *t2p*)

14.31.2.16 int T2Predictor_Insert (T2Predictor * *into_t2p*, const T2Predictor * *from_t2p*)

14.31.2.17 void T2Predictor_Keep (T2Predictor * , unsigned *nmjd*, const long double * *mjd*)

14.31.2.18 T2PredictorKind T2Predictor_Kind (T2Predictor * *t2p*)

14.31.2.19 `int T2Predictor_Read (T2Predictor * t2p, char * fname)`

14.31.2.20 `void T2Predictor_Write (const T2Predictor * t2p, char * fname)`

14.31.3 Variable Documentation

14.31.3.1 `int ChebyModelSet_OutOfRange`

14.32 tempo2pred_int.h File Reference

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

```
#include "tempo2pred.h"
```

Include dependency graph for tempo2pred_int.h:

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)

14.32.1 Function Documentation

- 14.32.1.1 void Cheby2D_Construct (Cheby2D * *cheby*, void(*) (long double **x*, long double **y*, int *nx*, int *ny*, long double **z*, void **info*) *func*, void * *info*)
- 14.32.1.2 void Cheby2D_Construct_x_Derivative (Cheby2D * *dcheby*, const Cheby2D * *cheby*)
- 14.32.1.3 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*)
- 14.32.1.4 void ChebyModel_Construct (ChebyModel * *cm*, const pulsar * *psr*)
- 14.32.1.5 void ChebyModel_Copy (ChebyModel * *cm*, ChebyModel * *from*)
- 14.32.1.6 void ChebyModel_Destroy (ChebyModel * *cm*)
- 14.32.1.7 long double ChebyModel_GetFrequency (const ChebyModel * *cm*, long double *mjd*, long double *freq*)
- 14.32.1.8 long double ChebyModel_GetPhase (const ChebyModel * *cm*, long double *mjd*, long double *freq*)
- 14.32.1.9 void ChebyModel_Init (ChebyModel * *cmodel*, int *nmjdcoeff*, int *nfreqcoeff*)
- 14.32.1.10 int ChebyModel_Read (ChebyModel * *cm*, FILE * *f*)
- 14.32.1.11 void ChebyModel_Test (ChebyModel * *cm*, const pulsar * *psr*, int *nmjd*, int *nfreq*, long double * *residualRMS*, long double * *residualMAV*)
- 14.32.1.12 void ChebyModel_Write (const ChebyModel * *cm*, FILE * *f*)
- 14.32.1.13 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*)
- 14.32.1.14 void ChebyModelSet_Destroy (ChebyModelSet * *cms*)
- 14.32.1.15 long double ChebyModelSet_GetFrequency (const ChebyModelSet * *cms*, long double *mjd*, long double *freq*)
- 14.32.1.16 ChebyModel* ChebyModelSet_GetNearest (const ChebyModelSet * *cms*, long double *mjd*)
- 14.32.1.17 long double ChebyModelSet_GetPhase (const ChebyModelSet * *cms*, long double *mjd*, long double *freq*)
- 14.32.1.18 void ChebyModelSet_Init (ChebyModelSet * *cms*)
- 14.32.1.19 int ChebyModelSet_Insert (ChebyModelSet * *cms*, const ChebyModelSet * *from*)
- 14.32.1.20 void ChebyModelSet_Keep (ChebyModelSet * *cms*, unsigned *nmjd*, const long double * *mjd*)
- 14.32.1.21 int ChebyModelSet_Read (ChebyModelSet * *cms*, FILE * *f*)
- 14.32.1.22 void ChebyModelSet_Test (ChebyModelSet * *cms*, const pulsar * *psr*, int *nmjd*, int *nfreq*, long double * *residualRMS*, long double * *residualMAV*)
- 14.32.1.23 void ChebyModelSet_Write (const ChebyModelSet * *cms*, FILE * *f*)
- 14.32.1.24 long double T1Polyco_GetFrequency (const T1Polyco * *t1p*, long double *mjd*, long double *freq*)

- 14.32.1.25 long double T1Polyco_GetPhase (const T1Polyco * *t1p*, long double *mjd*, long double *freq*)
- 14.32.1.26 int T1Polyco_Read (T1Polyco * *t1p*, FILE * *f*)
- 14.32.1.27 void T1Polyco_Write (const T1Polyco * *t1p*, FILE * *f*)
- 14.32.1.28 void T1PolycoSet_Destroy (T1PolycoSet * *t1ps*)
- 14.32.1.29 long double T1PolycoSet_GetFrequency (const T1PolycoSet * *t1ps*, long double *mjd*, long double *freq*)
- 14.32.1.30 T1Polyco* T1PolycoSet_GetNearest (long double *mjd*)
- 14.32.1.31 long double T1PolycoSet_GetPhase (const T1PolycoSet * *t1ps*, long double *mjd*, long double *freq*)
- 14.32.1.32 int T1PolycoSet_Read (T1PolycoSet * *t1ps*, FILE * *f*)
- 14.32.1.33 void T1PolycoSet_Write (const T1PolycoSet * *t1ps*, FILE * *f*)

14.33 tempo2Util.h File Reference

Functions

- double [turn_deg](#) (double *turn*)
- double [dms_turn](#) (char **line*)
- double [hms_turn](#) (char **line*)

14.33.1 Function Documentation

- 14.33.1.1 double [dms_turn](#) (char * *line*)
- 14.33.1.2 double [hms_turn](#) (char * *line*)
- 14.33.1.3 double [turn_deg](#) (double *turn*)

14.34 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*)

14.34.1 Function Documentation

- 14.34.1.1 `void cholesky_covarFunc2matrix (double ** m, double * covarFunc, int ndays, double * resx, double * resy, double * rese, int np, int nc)`
- 14.34.1.2 `void cholesky_dmModel (double ** m, double D, double d, double ref_freq, double * resx, double * resy, double * rese, int np, int nc)`
- 14.34.1.3 `void cholesky_dmModelCovarParam (double ** m, double alpha, double a, double b, double * resx, double * resy, double * rese, int np, int nc)`
- 14.34.1.4 `void cholesky_ecm (double ** m, char * fileName, double * resx, double * resy, double * rese, int np, int nc)`
- 14.34.1.5 `int cholesky_formUinv (double ** uinv, double ** m, int np)`
- 14.34.1.6 `void cholesky_powerlawModel (double ** m, double modelAlpha, double modelFc, double modelA, double * resx, double * resy, double * rese, int np, int nc)`
- 14.34.1.7 `void cholesky_powerlawModel_withBeta (double ** m, double modelAlpha, double beta, double modelFc, double modelA, double * resx, double * resy, double * rese, int np, int nc)`
- 14.34.1.8 `void cholesky_readFromCovarianceFunction (double ** m, const char * fname, double * resx, double * resy, double * rese, int np, int nc)`

14.35 TKfit.h File Reference

```
#include "TKmatrix.h"
#include "TKlongdouble.h"
Include dependency graph for TKfit.h:
```

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*)

14.35.1 Function Documentation

- 14.35.1.1 `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)`
- 14.35.1.2 `void TKfindPoly_d (double * px, double * py, int n, int m, double * p)`
- 14.35.1.3 `void TKfitPoly (double x, double * v, int m)`
- 14.35.1.4 `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)`
- 14.35.1.5 `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)`
- 14.35.1.6 `void TKleastSquares_svd_noErr (double * x, double * y, int n, double * p, int nf, void(*) (double, double[], int) fitFuncs)`
- 14.35.1.7 `void TKremovePoly_d (double * px, double * py, int n, int m)`
- 14.35.1.8 `void TKremovePoly_f (float * px, float * py, int n, int m)`
- 14.35.1.9 `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)`
- 14.35.1.10 `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)`

14.36 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]`

14.36.1 Macro Definition Documentation

14.36.1.1 `#define _LOG(_fmt, ...) _TKchklog(LOG_OUTFILE,_fmt,##__VA_ARGS__)`

14.36.1.2 `#define BOLDCOLOR RESETCOLOR "\033[1m"`

14.36.1.3 `#define DEPRECATED`

14.36.1.4 `#define ENDERR "\n***!!!!***"`

14.36.1.5 `#define ENDL "\n"`

14.36.1.6 `#define ERRORCOLOR RESETCOLOR "\033[1;31m"`

14.36.1.7 `#define LOG_OUTFILE stdout`

14.36.1.8 `#define logdbg(_fmt, ...) if(debugFlag)logmsg(_fmt,##__VA_ARGS__)`

14.36.1.9 `#define logerr(_fmt, ...) do{TK_STORE_ERROR(_fmt,##__VA_ARGS__); _LOG(WHEREERR _fmt
ENDERR ENDL, WHEREARG,##__VA_ARGS__);while(0)`

14.36.1.10 `#define logmsg(_fmt, ...) _LOG(WHERESTR _fmt ENDL, WHEREARG,##__VA_ARGS__)`

14.36.1.11 `#define logtchk(_fmt, ...) if(tcheck) _LOG(WHERETCHK _fmt ENDL,
WHEREARG,(clock()-timer_clk)/(float)CLOCKS_PER_SEC,##__VA_ARGS__)`

```

14.36.1.12 #define logwarn( _fmt, ... ) do{TK_STORE_WARNING(_fmt,##__VA_ARGS__); _LOG(WHEREWARN _fmt
        ENDL, WHEREARG,##__VA_ARGS__);}while(0)

14.36.1.13 #define RESETCOLOR "\033[0m"

14.36.1.14 #define TK_MAX_ERROR_LEN 128

14.36.1.15 #define TK_MAX_ERRORS 16

14.36.1.16 #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

14.36.1.17 #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

14.36.1.18 #define WARNCOLOR RESETCOLOR "\033[0;35m"

14.36.1.19 #define WHEREARG __FILE__, __LINE__

14.36.1.20 #define WHEREERR ERRORCOLOR "***ERROR***\n [%s:%d] " RESETCOLOR

14.36.1.21 #define WHERESTR "[%s:%d] "

14.36.1.22 #define WHERECHK "[%s:%d] T=%.2f s: "

14.36.1.23 #define WHEREWARN BOLDCOLOR "[%s:%d] " WARNCOLOR "Warning: " RESETCOLOR

```

14.36.2 Function Documentation

```

14.36.2.1 void _TKchklog ( FILE *, const char *, ... )

14.36.2.2 int logerr_check ( )

```

14.36.3 Variable Documentation

```

14.36.3.1 int debugFlag

14.36.3.2 int tcheck

14.36.3.3 clock_t timer_clk

14.36.3.4 unsigned TK_errorCount

14.36.3.5 char TK_errorlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

14.36.3.6 unsigned TK_warnCount

14.36.3.7 char TK_warnlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]

14.36.3.8 int writeResiduals

```

14.37 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`
- `#define powl powq`

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,...)

14.37.1 Macro Definition Documentation

14.37.1.1 `#define cosl cosq`

14.37.1.2 `#define fabsl fabsq`

14.37.1.3 `#define floorl floorq`

14.37.1.4 `#define FMT_LD "Q"`

14.37.1.5 `#define LD_PI M_PIq`

14.37.1.6 `#define longdouble(a) a##Q`

14.37.1.7 `#define LONGDOUBLE_IS_FLOAT128`

14.37.1.8 `#define LONGDOUBLE_ONE 1.0Q`

14.37.1.9 `#define powl powq`

14.37.1.10 `#define sinl sinq`

14.37.1.11 `#define USE_BUILTIN_LONGDOUBLE`

14.37.2 Typedef Documentation

14.37.2.1 `typedef __float128 longdouble`

14.37.3 Function Documentation

14.37.3.1 `int ld_fprintf (FILE * __stream, const char * __format, ...)`

14.37.3.2 `int ld_printf (const char * __format, ...)`

14.37.3.3 `int ld_sprintf (char * __str, const char * __format, ...)`

14.37.3.4 `longdouble parse_longdouble (const char * str)`

14.38 TKlongdouble.h File Reference

```
#include <math.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(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)`

14.38.1 Macro Definition Documentation

14.38.1.1 `#define ld_fprintf fprintf`

14.38.1.2 `#define LD_PI M_PI`

14.38.1.3 `#define ld_printf printf`

14.38.1.4 `#define ld_sprintf sprintf`

14.38.1.5 `#define longdouble(a) a##L`

14.38.1.6 `#define LONGDOUBLE_IS_IEEE754`

14.38.1.7 `#define LONGDOUBLE_ONE 1.0L`

14.38.1.8 `#define USE_BUILTIN_LONGDOUBLE`

14.38.2 Typedef Documentation

14.38.2.1 `typedef long double longdouble`

14.38.3 Function Documentation

14.38.3.1 `longdouble parse_longdouble (const char * str)`

14.39 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)`

14.39.1 Macro Definition Documentation

14.39.1.1 `#define ld_fprintf fprintf`

14.39.1.2 `#define LD_PI M_PI`

14.39.1.3 `#define ld_printf printf`

14.39.1.4 `#define ld_sprintf sprintf`

14.39.1.5 `#define longdouble(a) a##L`

14.39.1.6 `#define LONGDOUBLE_IS_IEEE754`

14.39.1.7 `#define LONGDOUBLE_ONE 1.0L`

14.39.1.8 `#define USE_BUILTIN_LONGDOUBLE`

14.39.2 Typedef Documentation

14.39.2.1 typedef long double longdouble

14.39.3 Function Documentation

14.39.3.1 longdouble parse_longdouble (const char * *str*)

14.40 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*)

14.40.1 Function Documentation

14.40.1.1 void free_2df (float ** *uinv*)

14.40.1.2 void free_blas (double ** *matrix*)

14.40.1.3 void free_uinv (double ** *uinv*)

14.40.1.4 int get_blas_cols (double ** *uinv*)

14.40.1.5 int get_blas_rows (double ** *uinv*)

14.40.1.6 float** malloc_2df (int *rows*, int *cols*)

14.40.1.7 double** malloc_blas (int *n*, int *m*)

14.40.1.8 double** malloc_uinv (int *n*)

14.40.1.9 void TKmultMatrix (double ** *idcm*, double ** *u*, int *ndata*, int *ndata2*, int *npol*, double ** *uout*)

14.40.1.10 void TKmultMatrix_sq (double ** *idcm*, double ** *u*, int *ndata*, int *npol*, double ** *uout*)

14.40.1.11 void TKmultMatrixVec (double ** *idcm*, double * *b*, int *ndata*, int *ndata2*, double * *bout*)

14.40.1.12 void TKmultMatrixVec_sq (double ** *idcm*, double * *b*, int *ndata*, double * *bout*)

14.41 TKspectrum.h File Reference

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 [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)
- 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)
- int [calcSpectra](#) (double **uinv, double *resx, double *resy, int nres, double *specX, double *specY, int nfit)

Variables

- bool [verbose_calc_spectra](#)

14.41.1 Macro Definition Documentation

14.41.1.1 `#define ABS(x) ((x) < 0 ? -(x) : (x))`

14.41.1.2 `#define MAX(x, y) ((x) > (y) ? (x) : (y))`

14.41.1.3 `#define MIN(x, y) ((x) < (y) ? (x) : (y))`

14.41.2 Typedef Documentation

14.41.2.1 `typedef struct complexVal complexVal`

14.41.3 Function Documentation

14.41.3.1 `int calcSpectra (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY, int nfit)`

14.41.3.2 `int calcSpectraErr (double ** uinv, double * resx, double * resy, int nres, double * specX, double * specY, double * specE, int nfit)`

14.41.3.3 `void fit4 (int * nfit, double * p4, double * cov4, int ndostats, double * chidf, double * avewt)`

14.41.3.4 `void getprtj (int n)`

14.41.3.5 `void getweights (int n, double * wt)`

14.41.3.6 `void indexx8 (int n, double * arrin, int * indx)`

14.41.3.7 `void mat20 (double sam[21][21], double a[21][21], int n, double * determ, int * nbad)`

14.41.3.8 `void sineFunc (double x, double * v, int ma)`

14.41.3.9 `void TK_dft (double * x, double * y, int n, double * outX, double * outY, int * outN, double * outY_re, double * outY_im)`

14.41.3.10 `int TK_fft (short int dir, long n, double * x, double * y)`

14.41.3.11 `void TK_fitSine (double * x, double * y, double * e, int n, int wErr, double * outX, double * outY, int * outN)`

14.41.3.12 `void TK_fitSinusoids (double * x, double * y, double * sig, int n, double * outX, double * outY, int * outN)`

14.41.3.13 `void TK_weightLS (double * x, double * y, double * sig, int n, double * outX, double * outY, int * outN, double * outY_re, double * outY_im)`

14.41.3.14 `void TKaveragePts (double * x, double * y, int n, int width, double * meanX, double * meanY, int * nMean)`

14.41.3.15 `void TKboxcar (double * x, double * y, int n, double * ox, double * oy, int * on, int width)`

14.41.3.16 `void TKcmonot (int n, double x[], double y[], double yd[][4])`

14.41.3.17 `void TKfirstDifference (double * x, double * y, int n)`

14.41.3.18 `void TKhann (double * x, double * y, int n, double * ox, double * oy, int * on, int width)`

14.41.3.19 `void TKinterpolateSplineSmoothFixedXPts (double * inX, double * inY, int inN, double * interpX, double * interpY, int nInterp)`

- 14.41.3.20 `void TKlomb_d (double * x, double * y, int n, double ofac, double hifac, double * ox, double * oy, int * outN, double * var)`
- 14.41.3.21 `void TKsortit (double * x, double * y, int n)`
- 14.41.3.22 `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)`
- 14.41.3.23 `void TKSpline_interpolate (int n, double * x, double * y, double yd[][4], double * interpX, double * interpY, int nInterp)`

14.41.4 Variable Documentation

- 14.41.4.1 `bool verbose_calc_spectra`

14.42 TKsvd.h File Reference

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)`

14.42.1 Function Documentation

- 14.42.1.1 `void TKbacksubstitution_svd (longdouble ** V, longdouble * w, longdouble ** U, longdouble * b, longdouble * x, int n, int nf)`
- 14.42.1.2 `void TKbidiagonal (longdouble ** a, longdouble * anorm, int ndata, int nfit, longdouble ** v, longdouble * w, longdouble ** u, longdouble * rv1)`
- 14.42.1.3 `longdouble TKpythag (longdouble a, longdouble b)`
- 14.42.1.4 `void TKsingularValueDecomposition_Isq (longdouble ** designMatrix, int n, int nf, longdouble ** v, longdouble * w, longdouble ** u)`

Index

`_DARWIN_USE_64_BIT_INODE`
 `config.h`, 73

`_LOG`
 `TKlog.h`, 122
`_TKchklog`
 `TKlog.h`, 123

`ABS`
 `TKspectrum.h`, 129

`ACCEL_LSQ`
 `T2accel.h`, 84
`ACCEL_MULTMATRIX`
 `T2accel.h`, 84

`ACCEL_UINV`
 `T2accel.h`, 84

`aSize`
 parameter, 47

`AU_DIST`
 `tempo2.h`, 98

`AULTSC`
 `tempo2.h`, 98

`accel_lsqr`
 `T2accel.h`, 84

`accel_multMatrix`
 `T2accel.h`, 84

`accel_multMatrixVec`
 `T2accel.h`, 84

`accel_uinv`
 `T2accel.h`, 85

`across_g`
 `gwSrc`, 36
 `gwgeneralSrc`, 33

`across_im_g`
 `gwSrc`, 36
 `gwgeneralSrc`, 33

`addTNGlobalEQ`
 pulsar, 54

`addedNoise`
 observation, 40

`allocateMemory`
 `tempo2.h`, 109

`apls_g`
 `gwSrc`, 36
 `gwgeneralSrc`, 33

`apls_im_g`
 `gwSrc`, 36
 `gwgeneralSrc`, 33

`asl_g`
 `gwgeneralSrc`, 33

`asl_im_g`

`gwgeneralSrc`, 33

`ast_g`
 `gwgeneralSrc`, 34

`ast_im_g`
 `gwgeneralSrc`, 34

`au`
 `jpl_eph_data`, 37

`auto_constraints`
 pulsar, 54

`autoConstraints`
 `tempo2.h`, 109

`autosetDMCM`
 `constraints.h`, 74

`AverageDMResiduals`
 pulsar, 54

`AverageEpochWidth`
 pulsar, 54

`AverageFlag`
 pulsar, 54

`AverageResiduals`
 pulsar, 54

`averagebat`
 observation, 40

`averagedmbat`
 observation, 40

`averagedmerr`
 observation, 40

`averagedmres`
 observation, 40

`averageerr`
 observation, 40

`averageres`
 observation, 40

`avx_g`
 `gwgeneralSrc`, 34

`avx_im_g`
 `gwgeneralSrc`, 34

`avy_g`
 `gwgeneralSrc`, 34

`avy_im_g`
 `gwgeneralSrc`, 34

`BIG_G`
 `tempo2.h`, 98

`BOLDCOLOR`
 `TKlog.h`, 122

`BTJmodel`
 `tempo2.h`, 109

`BTXmodel`
 `tempo2.h`, 109

- BTmodel
 - tempo2.h, 109
- bat
 - observation, 40
- batCorr
 - observation, 40
- bbat
 - observation, 40
- binary_frequency
 - T1Polyco, 68
- binary_phase
 - T1Polyco, 68
- binaryModel
 - pulsar, 54
- bootStrap
 - pulsar, 54
- bootstrap
 - tempo2.h, 109
- c_fileptr
 - read_fortran.h, 83
- c_fileptr2
 - read_fortran2.h, 84
- CONSTRAINTfuncs
 - constraints.h, 75
- CVSdisplayVersion
 - tempo2.h, 109
- cache
 - jpl_eph_data, 37
- calcRMS
 - tempo2.h, 109
- calcShapiro
 - pulsar, 54
- calcSpectra
 - TKspectrum.h, 129
- calcSpectraErr
 - TKspectrum.h, 129
- calculate_bclt
 - tempo2.h, 109
- calculateResidualGW
 - GWsim.h, 77
- calculateResidualgeneralGW
 - GWsim.h, 77
- cgw_angpol
 - pulsar, 54
- cgw_cosinc
 - pulsar, 54
- cgw_h0
 - pulsar, 54
- cgw_mc
 - pulsar, 54
- Cheby
 - tempo2pred.h, 116
- cheby
 - ChebyModel, 30
 - T2Predictor, 69
- Cheby2D, 29
 - coeff, 29
 - nx, 29
 - ny, 29
- Cheby2D_Construct
 - tempo2pred_int.h, 118
- Cheby2D_Construct_x_Derivative
 - tempo2pred_int.h, 118
- Cheby2D_Test
 - tempo2pred_int.h, 118
- ChebyModel, 29
 - cheby, 30
 - dispersion_constant, 30
 - freq_end, 30
 - freq_start, 30
 - frequency_cheby, 30
 - mjd_end, 30
 - mjd_start, 30
 - psrname, 30
 - sitename, 30
- ChebyModel_Construct
 - tempo2pred_int.h, 118
- ChebyModel_Copy
 - tempo2pred_int.h, 118
- ChebyModel_Destroy
 - tempo2pred_int.h, 118
- ChebyModel_GetFrequency
 - tempo2pred_int.h, 118
- ChebyModel_GetPhase
 - tempo2pred_int.h, 118
- ChebyModel_Init
 - tempo2pred_int.h, 118
- ChebyModel_Read
 - tempo2pred_int.h, 118
- ChebyModel_Test
 - tempo2pred_int.h, 118
- ChebyModel_Write
 - tempo2pred_int.h, 118
- ChebyModelSet, 30
 - nsegments, 30
 - segments, 30
- ChebyModelSet_Construct
 - tempo2pred_int.h, 118
- ChebyModelSet_Destroy
 - tempo2pred_int.h, 118
- ChebyModelSet_GetFrequency
 - tempo2pred_int.h, 118
- ChebyModelSet_GetNearest
 - tempo2pred_int.h, 118
- ChebyModelSet_GetPhase
 - tempo2pred_int.h, 118
- ChebyModelSet_Init
 - tempo2pred_int.h, 118
- ChebyModelSet_Insert
 - tempo2pred_int.h, 118
- ChebyModelSet_Keep
 - tempo2pred_int.h, 118
- ChebyModelSet_OutOfRange
 - tempo2pred.h, 117
- ChebyModelSet_Read
 - tempo2pred_int.h, 118

- ChebyModelSet_Test
 - tempo2pred_int.h, [118](#)
- ChebyModelSet_Write
 - tempo2pred_int.h, [118](#)
- cholesky.h, [71](#)
 - cholesky_covarFunc2matrix, [71](#)
 - cholesky_dmModel, [71](#)
 - cholesky_dmModelCovarParam, [71](#)
 - cholesky_ecm, [71](#)
 - cholesky_formUinv, [71](#)
 - cholesky_powerlawModel, [71](#)
 - cholesky_powerlawModel_withBeta, [71](#)
 - cholesky_readFromCovarianceFunction, [71](#)
- cholesky_covarFunc2matrix
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- cholesky_dmModel
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- cholesky_dmModelCovarParam
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- cholesky_ecm
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- cholesky_formUinv
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- cholesky_powerlawModel
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- cholesky_powerlawModel_withBeta
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- cholesky_readFromCovarianceFunction
 - cholesky.h, [71](#)
 - TKcholesky.h, [120](#)
- choleskyRoutines.h, [72](#)
- clk_offsE
 - pulsar, [54](#)
- clk_offsT
 - pulsar, [54](#)
- clk_offsV
 - pulsar, [54](#)
- clkOffsN
 - pulsar, [54](#)
- clock
 - pulsar, [54](#)
- clock_correction, [30](#)
 - correction, [31](#)
 - corrects_to, [31](#)
- clock_name
 - observatory, [46](#)
- clockCorr
 - observation, [40](#)
- clockFromOverride
 - pulsar, [54](#)
- close_file
 - read_fortran.h, [83](#)
- close_file2
 - read_fortran2.h, [83](#)
- code
 - observatory, [46](#)
- coeff
 - Cheby2D, [29](#)
 - T1Polyco, [68](#)
- comment
 - storePrecision, [67](#)
- complexVal, [31](#)
 - imag, [31](#)
 - real, [31](#)
 - TKspectrum.h, [129](#)
- compute_tropospheric_delays
 - tempo2.h, [109](#)
- computeConstraintWeights
 - constraints.h, [74](#)
- config.h, [72](#)
 - _DARWIN_USE_64_BIT_INODE, [73](#)
 - F77_FUNC, [73](#)
 - F77_FUNC_, [73](#)
 - HAVE_BLAS, [73](#)
 - HAVE_CFITSIO, [73](#)
 - HAVE_DLERROR, [73](#)
 - HAVE_DLFCN_H, [73](#)
 - HAVE_FFTW3, [73](#)
 - HAVE_INTPYPES_H, [73](#)
 - HAVE_LAPACK, [73](#)
 - HAVE_LIBDL, [73](#)
 - HAVE_LIBDLLOADER, [73](#)
 - HAVE_LIBM, [73](#)
 - HAVE_MEMORY_H, [73](#)
 - HAVE_PGPLOT, [73](#)
 - HAVE_PTHREAD, [73](#)
 - HAVE_STDINT_H, [73](#)
 - HAVE_STDLIB_H, [73](#)
 - HAVE_STRING_H, [73](#)
 - HAVE_STRINGS_H, [73](#)
 - HAVE_SYS_STAT_H, [73](#)
 - HAVE_SYS_TYPES_H, [73](#)
 - HAVE_UNISTD_H, [73](#)
 - LT_OBJDIR, [73](#)
 - PACKAGE, [73](#)
 - PACKAGE_BUGREPORT, [73](#)
 - PACKAGE_NAME, [73](#)
 - PACKAGE_STRING, [73](#)
 - PACKAGE_TARNAME, [74](#)
 - PACKAGE_URL, [74](#)
 - PACKAGE_VERSION, [74](#)
 - STDC_HEADERS, [74](#)
 - TEMPO2_ARCH, [74](#)
 - VERSION, [74](#)
- consFunc_dmmodel_cw
 - constraints.h, [74](#)
- consFunc_dmmodel_cw_year
 - constraints.h, [74](#)
- consFunc_dmmodel_dm1

constraints.h, [74](#)
 consFunc_dmmodel_mean
 constraints.h, [74](#)
 consFunc_ifunc
 constraints.h, [75](#)
 consFunc_ifunc_year
 constraints.h, [75](#)
 consFunc_qifunc_c_year
 constraints.h, [75](#)
 consFunc_qifunc_p_year
 constraints.h, [75](#)
 consFunc_quad_ifunc_c
 constraints.h, [75](#)
 consFunc_quad_ifunc_p
 constraints.h, [75](#)
 consFunc_tel_dx
 constraints.h, [75](#)
 consFunc_tel_dy
 constraints.h, [75](#)
 consFunc_tel_dz
 constraints.h, [75](#)
 constraint
 tempo2.h, [105](#)
 constraint_LAST
 tempo2.h, [106](#)
 constraint_dmmodel_cw_0
 tempo2.h, [105](#)
 constraint_dmmodel_cw_1
 tempo2.h, [105](#)
 constraint_dmmodel_cw_2
 tempo2.h, [105](#)
 constraint_dmmodel_cw_3
 tempo2.h, [105](#)
 constraint_dmmodel_cw_px
 tempo2.h, [106](#)
 constraint_dmmodel_cw_year_cos
 tempo2.h, [106](#)
 constraint_dmmodel_cw_year_cos2
 tempo2.h, [106](#)
 constraint_dmmodel_cw_year_sin
 tempo2.h, [106](#)
 constraint_dmmodel_cw_year_sin2
 tempo2.h, [106](#)
 constraint_dmmodel_cw_year_xcos
 tempo2.h, [106](#)
 constraint_dmmodel_cw_year_xsin
 tempo2.h, [106](#)
 constraint_dmmodel_dm1
 tempo2.h, [105](#)
 constraint_dmmodel_mean
 tempo2.h, [105](#)
 constraint_ifunc_0
 tempo2.h, [105](#)
 constraint_ifunc_1
 tempo2.h, [105](#)
 constraint_ifunc_2
 tempo2.h, [105](#)
 constraint_ifunc_year_cos
 tempo2.h, [106](#)
 constraint_ifunc_year_cos2
 tempo2.h, [106](#)
 constraint_ifunc_year_sin
 tempo2.h, [106](#)
 constraint_ifunc_year_sin2
 tempo2.h, [106](#)
 constraint_ifunc_year_xcos
 tempo2.h, [106](#)
 constraint_ifunc_year_xsin
 tempo2.h, [106](#)
 constraint_label
 tempo2.h, [104](#)
 constraint_qifunc_c_year_cos
 tempo2.h, [106](#)
 constraint_qifunc_c_year_cos2
 tempo2.h, [106](#)
 constraint_qifunc_c_year_sin
 tempo2.h, [106](#)
 constraint_qifunc_c_year_sin2
 tempo2.h, [106](#)
 constraint_qifunc_c_year_xcos
 tempo2.h, [106](#)
 constraint_qifunc_c_year_xsin
 tempo2.h, [106](#)
 constraint_qifunc_p_year_cos
 tempo2.h, [106](#)
 constraint_qifunc_p_year_cos2
 tempo2.h, [106](#)
 constraint_qifunc_p_year_sin
 tempo2.h, [106](#)
 constraint_qifunc_p_year_sin2
 tempo2.h, [106](#)
 constraint_qifunc_p_year_xcos
 tempo2.h, [106](#)
 constraint_qifunc_p_year_xsin
 tempo2.h, [106](#)
 constraint_quad_ifunc_c_0
 tempo2.h, [105](#)
 constraint_quad_ifunc_c_1
 tempo2.h, [106](#)
 constraint_quad_ifunc_c_2
 tempo2.h, [106](#)
 constraint_quad_ifunc_p_0
 tempo2.h, [105](#)
 constraint_quad_ifunc_p_1
 tempo2.h, [105](#)
 constraint_quad_ifunc_p_2
 tempo2.h, [105](#)
 constraint_str
 enum_str.h, [76](#)
 constraint_tel_dx_0
 tempo2.h, [105](#)
 constraint_tel_dx_1
 tempo2.h, [105](#)
 constraint_tel_dx_2
 tempo2.h, [105](#)
 constraint_tel_dy_0

- tempo2.h, 105
- constraint_tel_dy_1
 - tempo2.h, 105
- constraint_tel_dy_2
 - tempo2.h, 105
- constraint_tel_dz_0
 - tempo2.h, 105
- constraint_tel_dz_1
 - tempo2.h, 105
- constraint_tel_dz_2
 - tempo2.h, 105
- constraintCounters
 - FitInfo, 32
- constraintDerivFunc
 - tempo2.h, 104
- constraintDerivs
 - FitInfo, 32
- constraintIndex
 - FitInfo, 32
- constraints
 - pulsar, 54
- constraints.h, 74
 - autosetDMCM, 74
 - CONSTRAINTfuncs, 75
 - computeConstraintWeights, 74
 - consFunc_dmmodel_cw, 74
 - consFunc_dmmodel_cw_year, 74
 - consFunc_dmmodel_dm1, 74
 - consFunc_dmmodel_mean, 74
 - consFunc_ifunc, 75
 - consFunc_ifunc_year, 75
 - consFunc_qifunc_c_year, 75
 - consFunc_qifunc_p_year, 75
 - consFunc_quad_ifunc_c, 75
 - consFunc_quad_ifunc_p, 75
 - consFunc_tel_dx, 75
 - consFunc_tel_dy, 75
 - consFunc_tel_dz, 75
 - get_constraint_name, 75
 - standardConstraintFunctions, 75
- copyPSR
 - tempo2.h, 109
- copyParam
 - tempo2.h, 109
- correctTroposphere
 - pulsar, 55
- correction
 - clock_correction, 31
- correctionTT_TB
 - observation, 40
- correctionTT_Teph
 - observation, 40
- correctionTT_calcEph
 - observation, 40
- correctionUT1
 - observation, 41
- correctionsTT
 - observation, 40
- corrects_to
 - clock_correction, 31
- cosl
 - TKlongdouble.float128.h, 124
- covar
 - pulsar, 55
- covarFuncFile
 - tempo2.h, 114
- curr_cache_loc
 - jpl_eph_data, 37
- DDGRmodel
 - tempo2.h, 110
- DDHmodel
 - tempo2.h, 110
- DDKmodel
 - tempo2.h, 110
- DDSmodel
 - tempo2.h, 110
- DDmodel
 - tempo2.h, 110
- DEPRECATED
 - TKlog.h, 122
- DLL_FUNC
 - jpleph.h, 81
- DM_CONST
 - tempo2.h, 98
- DM_CONST_SI
 - tempo2.h, 98
- dadt
 - GWsim.h, 77
- data
 - DynamicArray, 32
- date_string
 - T1Polyco, 68
- dcmFile
 - tempo2.h, 114
- debugFlag
 - TKlog.h, 123
- decjStrPost
 - pulsar, 55
- decjStrPre
 - pulsar, 55
- decsim
 - pulsar, 55
- dedt
 - GWsim.h, 77
- defineClockCorrectionSequence
 - tempo2.h, 110
- delayCorr
 - observation, 41
- deleteFileName
 - pulsar, 55
- deleted
 - observation, 41
- destroyMemory
 - tempo2.h, 110
- destroyOne
 - tempo2.h, 110

- dilateFreq
 - pulsar, 55
- dispersion_constant
 - ChebyModel, 30
- displayCVSversion
 - tempo2.h, 114
- displayMsg
 - tempo2.h, 110
- displayParameters
 - tempo2.h, 110
- dist_bin
 - gwSrc, 36
 - gwgeneralSrc, 34
- dm
 - T1Polyco, 68
- dm_delays
 - tempo2.h, 110
- dmOffset
 - pulsar, 55
- dmoftsCM
 - pulsar, 55
- dmoftsCM_error
 - pulsar, 55
- dmoftsCM_mjd
 - pulsar, 55
- dmoftsCM_weight
 - pulsar, 55
- dmoftsCMnum
 - pulsar, 55
- dmoftsDM
 - pulsar, 55
- dmoftsDM_error
 - pulsar, 55
- dmoftsDM_mjd
 - pulsar, 55
- dmoftsDM_weight
 - pulsar, 55
- dmoftsDMnum
 - pulsar, 55
- dms_turn
 - tempo2.h, 110
 - tempo2Util.h, 119
- doFit
 - tempo2.h, 110
- doFitAll
 - tempo2.h, 110
- doFitDCM
 - tempo2.h, 110
- doFitGlobal
 - tempo2.h, 110
- documentation/1_USER_GUIDE.md, 75
- documentation/2_developers.md, 75
- documentation/3_DEVELOPER_GUIDE.md, 75
- documentation/4_directories.md, 75
- documentation/5_plugins.md, 75
- doppler
 - T1Polyco, 68
- dotProduct
 - GWsim.h, 77
- dotproduct
 - tempo2.h, 110
- dttd
 - GWsim.h, 77
- DynamicArray, 31
 - data, 32
 - elem_size, 32
 - nallocated, 32
 - nelem, 32
- DynamicArray_free
 - dynarr.h, 76
- DynamicArray_init
 - dynarr.h, 76
- DynamicArray_push_back
 - dynarr.h, 76
- DynamicArray_resize
 - dynarr.h, 76
- dynarr.h, 75
 - DynamicArray_free, 76
 - DynamicArray_init, 76
 - DynamicArray_push_back, 76
 - DynamicArray_resize, 76
- ECLIPTIC_OBLIQUITY
 - tempo2.h, 114
- ECLIPTIC_OBLIQUITY_VAL
 - tempo2.h, 98
- ELL1Hmodel
 - tempo2.h, 110
- ELL1model
 - tempo2.h, 110
- ENDERR
 - TKlog.h, 122
- ENDL
 - TKlog.h, 122
- ERRORCOLOR
 - TKlog.h, 122
- earth_ssb
 - observation, 41
- earthMoonBary_earth
 - observation, 41
- earthMoonBary_ssb
 - observation, 41
- eccRes
 - GWsim.h, 77
- eccResWithEnergy
 - GWsim.h, 77
- eclCoord
 - pulsar, 55
- efac
 - observation, 41
- einsteinRate
 - observation, 41
- elem_size
 - DynamicArray, 32
- emrat
 - jpl_eph_data, 37
- enum_str.h, 76

- constraint_str, 76
 - label_str, 76
- eopc04_file
 - pulsar, 56
- ephem_end
 - jpl_eph_data, 37
- ephem_start
 - jpl_eph_data, 37
- ephem_step
 - jpl_eph_data, 37
- ephemeris
 - pulsar, 56
- ephemeris_version
 - jpl_eph_data, 38
- equ2ecl
 - tempo2.h, 110
- equad
 - observation, 41
- err
 - parameter, 47
- F77_FUNC
 - config.h, 73
- F77_FUNC_
 - config.h, 73
- FB90_TIMEEPH
 - tempo2.h, 98
- FITfuncs
 - tempo2.h, 110
- FMT_LD
 - TKlongdouble.float128.h, 124
- fabsl
 - TKlongdouble.float128.h, 124
- Fe
 - GWsim.h, 77
- fileName
 - TabulatedFunction, 69
- filterStr
 - pulsar, 56
- Findphi
 - GWsim.h, 77
- fit4
 - TKspectrum.h, 129
- fitChisq
 - pulsar, 56
- fitFlag
 - parameter, 47
- fitFunc
 - pulsar, 56
- FitInfo, 32
 - constraintCounters, 32
 - constraintDerivs, 32
 - constraintIndex, 32
 - nConstraints, 32
 - nParams, 32
 - paramCounters, 32
 - paramDerivs, 32
 - paramIndex, 33
 - tempo2.h, 104
 - updateFunctions, 33
- fitJump
 - pulsar, 56
- fitMode
 - pulsar, 56
- fitNfree
 - pulsar, 56
- fitParamGlobalI
 - pulsar, 56
- fitParamGlobalK
 - pulsar, 56
- fitParamI
 - pulsar, 56
- fitParamK
 - pulsar, 56
- fitinfo
 - pulsar, 56
- fixedFormat
 - pulsar, 56
- fjumpID
 - pulsar, 56
- flagID
 - observation, 41
- flagVal
 - observation, 41
- floorl
 - TKlongdouble.float128.h, 124
- fname
 - observation, 41
- forceGlobalFit
 - tempo2.h, 114
- formBats
 - tempo2.h, 110
- formBatsAll
 - tempo2.h, 110
- formResiduals
 - tempo2.h, 110
- fortran_mod
 - tempo2.h, 110
- fortran_nint
 - tempo2.h, 110
- fortran_nlong
 - tempo2.h, 110
- free_2df
 - TKmatrix.h, 127
- free_blas
 - TKmatrix.h, 127
- free_uinv
 - TKmatrix.h, 127
- freq
 - observation, 41
- freq_end
 - ChebyModel, 30
- freq_start
 - ChebyModel, 30
- freqSSB
 - observation, 42
- frequency_cheby

- ChebyModel, [30](#)
- frequency_obs
 - T1Polyco, [68](#)
- frequency_psr_0
 - T1Polyco, [68](#)
- GM
 - tempo2.h, [98](#)
- GM_C3
 - tempo2.h, [99](#)
- GMJ_C3
 - tempo2.h, [99](#)
- GMN_C3
 - tempo2.h, [99](#)
- GMS_C3
 - tempo2.h, [99](#)
- GMU_C3
 - tempo2.h, [99](#)
- GMV_C3
 - tempo2.h, [99](#)
- GWanisotropicbackground
 - GWsim.h, [78](#)
- GWbackground
 - GWsim.h, [78](#)
- GWbackground_read
 - GWsim.h, [78](#)
- GWbackground_write
 - GWsim.h, [78](#)
- GWdipolebackground
 - GWsim.h, [78](#)
- GWgeneralanisotropicbackground
 - GWsim.h, [78](#)
- GWgeneralbackground
 - GWsim.h, [78](#)
- GWgeneralbackground_read
 - GWsim.h, [78](#)
- GWgeneralbackground_write
 - GWsim.h, [78](#)
- GWsim.h, [76](#)
 - calculateResidualGW, [77](#)
 - calculateResidualgeneralGW, [77](#)
 - dadt, [77](#)
 - dedt, [77](#)
 - dotProduct, [77](#)
 - dtdt, [77](#)
 - eccRes, [77](#)
 - eccResWithEnergy, [77](#)
 - Fe, [77](#)
 - Findphi, [77](#)
 - GWanisotropicbackground, [78](#)
 - GWbackground, [78](#)
 - GWbackground_read, [78](#)
 - GWbackground_write, [78](#)
 - GWdipolebackground, [78](#)
 - GWgeneralanisotropicbackground, [78](#)
 - GWgeneralbackground, [78](#)
 - GWgeneralbackground_read, [78](#)
 - GWgeneralbackground_write, [78](#)
 - gwSrc, [77](#)
 - gwgenSpec, [77](#)
 - gwgeneralSrc, [77](#)
 - matrixMult, [78](#)
 - psrangle, [78](#)
 - Rs, [78](#)
 - setupGW, [78](#)
 - setupPulsar_GWsim, [78](#)
 - setupgeneralGW, [78](#)
 - sphharm, [78](#)
- genrand_int32
 - T2toolkit.h, [90](#)
- genrand_real1
 - T2toolkit.h, [90](#)
- get_EOP
 - tempo2.h, [110](#)
- get_OneobsCoord
 - tempo2.h, [111](#)
- get_blas_cols
 - TKmatrix.h, [127](#)
- get_blas_rows
 - TKmatrix.h, [127](#)
- get_constraint_name
 - constraints.h, [75](#)
- get_obsCoord
 - tempo2.h, [111](#)
- get_obsCoord_IAU2000B
 - tempo2.h, [111](#)
- getCholeskyMatrix
 - tempo2.h, [111](#)
- getClockCorrections
 - tempo2.h, [111](#)
- getCorrection
 - tempo2.h, [111](#)
- getCorrectionTT
 - tempo2.h, [111](#)
- getInputs
 - tempo2.h, [111](#)
- getObservatory
 - tempo2.h, [111](#)
- getParamDeriv
 - tempo2.h, [111](#)
- getParameterValue
 - tempo2.h, [111](#)
- getprtj
 - TKspectrum.h, [129](#)
- getweights
 - TKspectrum.h, [129](#)
- globalNfit
 - pulsar, [56](#)
- globalNoConstrain
 - pulsar, [56](#)
- gwSrc, [35](#)
 - across_g, [36](#)
 - across_im_g, [36](#)
 - aplus_g, [36](#)
 - aplus_im_g, [36](#)
 - dist_bin, [36](#)
 - GWsim.h, [77](#)

- 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](#)
- gwb_decj
 - pulsar, [57](#)
- gwb_epoch
 - pulsar, [57](#)
- gwb_geom_c
 - pulsar, [57](#)
- gwb_geom_p
 - pulsar, [57](#)
- gwb_raj
 - pulsar, [57](#)
- gwb_width
 - pulsar, [57](#)
- gwecc_dec
 - pulsar, [57](#)
- gwecc_distance
 - pulsar, [57](#)
- gwecc_e
 - pulsar, [57](#)
- gwecc_epoch
 - pulsar, [57](#)
- gwecc_inc
 - pulsar, [57](#)
- gwecc_m1
 - pulsar, [57](#)
- gwecc_m2
 - pulsar, [57](#)
- gwecc_nodes_orientation
 - pulsar, [57](#)
- gwecc_orbital_period
 - pulsar, [57](#)
- gwecc_psrdist
 - pulsar, [57](#)
- gwecc_pulsarTermOn
 - pulsar, [57](#)
- gwecc_ra
 - pulsar, [57](#)
- gwecc_redshift
 - pulsar, [57](#)
- gwecc_theta_0
 - pulsar, [57](#)
- gwecc_theta_nodes
 - pulsar, [57](#)
- gwgenSpec, [34](#)
 - GWsim.h, [77](#)
 - 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, [33](#)
 - across_im_g, [33](#)
 - aplus_g, [33](#)
 - aplus_im_g, [33](#)
 - asl_g, [33](#)
 - asl_im_g, [33](#)
 - 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, [77](#)
 - 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, [57](#)
- gwm_dphase
 - pulsar, [57](#)
- gwm_epoch
 - pulsar, [57](#)
- gwm_phi
 - pulsar, [57](#)
- gwm_raj
 - pulsar, [57](#)
- gwsrc_across_i
 - pulsar, [57](#)
- gwsrc_across_i_e
 - pulsar, [58](#)
- gwsrc_across_r
 - pulsar, [58](#)
- gwsrc_across_r_e
 - pulsar, [58](#)
- gwsrc_aplus_i
 - pulsar, [58](#)
- gwsrc_aplus_i_e
 - pulsar, [58](#)
- gwsrc_aplus_r
 - pulsar, [58](#)
- gwsrc_aplus_r_e
 - pulsar, [58](#)
- gwsrc_dec

- pulsar, [58](#)
- gwsrc_epoch
 - pulsar, [58](#)
- gwsrc_psrdist
 - pulsar, [58](#)
- gwsrc_ra
 - pulsar, [58](#)
- h
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- h_im
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- HAVE_BLAS
 - config.h, [73](#)
- HAVE_CFITSIO
 - config.h, [73](#)
- HAVE_DLERROR
 - config.h, [73](#)
- HAVE_DLFCN_H
 - config.h, [73](#)
- HAVE_FFTW3
 - config.h, [73](#)
- HAVE_GWSIM_H
 - tempo2.h, [99](#)
- HAVE_INTTYPES_H
 - config.h, [73](#)
- HAVE_LAPACK
 - config.h, [73](#)
- HAVE_LIBDL
 - config.h, [73](#)
- HAVE_LIBDLLOADER
 - config.h, [73](#)
- HAVE_LIBM
 - config.h, [73](#)
- HAVE_MEMORY_H
 - config.h, [73](#)
- HAVE_PGPLOT
 - config.h, [73](#)
- HAVE_PTHREAD
 - config.h, [73](#)
- HAVE_STDINT_H
 - config.h, [73](#)
- HAVE_STDLIB_H
 - config.h, [73](#)
- HAVE_STRING_H
 - config.h, [73](#)
- HAVE_STRINGS_H
 - config.h, [73](#)
- HAVE_SYS_STAT_H
 - config.h, [73](#)
- HAVE_SYS_TYPES_H
 - config.h, [73](#)
- HAVE_UNISTD_H
 - config.h, [73](#)
- header_line
 - TabulatedFunction, [69](#)
- height_grs80
 - observatory, [46](#)
- hms_turn
 - tempo2.h, [111](#)
 - tempo2Util.h, [119](#)
- IF99_TIMEEPH
 - tempo2.h, [99](#)
- IFTE_DeltaT
 - ifteph.h, [79](#)
- IFTE_DeltaTDot
 - ifteph.h, [79](#)
- IFTE_JD0
 - ifteph.h, [79](#)
- IFTE_K
 - ifteph.h, [79](#)
- IFTE_KM1
 - ifteph.h, [79](#)
- IFTE_LC
 - ifteph.h, [79](#)
- IFTE_MJD0
 - ifteph.h, [79](#)
- IFTE_TEPH0
 - ifteph.h, [79](#)
- IFTE_close_file
 - ifteph.h, [79](#)
- IFTE_get_DeltaT_DeltaTDot
 - ifteph.h, [79](#)
- IFTE_get_vE
 - ifteph.h, [79](#)
- IFTE_get_vE_vEDot
 - ifteph.h, [79](#)
- IFTE_get_vEDot
 - ifteph.h, [79](#)
- IFTE_init
 - ifteph.h, [79](#)
- IFTEPH_FILE
 - tempo2.h, [99](#)
- id_residual
 - tempo2.h, [111](#)
- ifile
 - jpl_eph_data, [38](#)
- ifteph.h, [78](#)
 - IFTE_DeltaT, [79](#)
 - IFTE_DeltaTDot, [79](#)
 - IFTE_JD0, [79](#)
 - IFTE_K, [79](#)
 - IFTE_KM1, [79](#)
 - IFTE_LC, [79](#)
 - IFTE_MJD0, [79](#)
 - IFTE_TEPH0, [79](#)
 - IFTE_close_file, [79](#)
 - IFTE_get_DeltaT_DeltaTDot, [79](#)
 - IFTE_get_vE, [79](#)
 - IFTE_get_vE_vEDot, [79](#)
 - IFTE_get_vEDot, [79](#)
 - IFTE_init, [79](#)
- ifunc
 - t2fit_ifunc.h, [87](#)
- ifunc_weights

- pulsar, 58
- ifuncE
 - pulsar, 58
- ifuncN
 - pulsar, 58
- ifuncT
 - pulsar, 58
- ifuncV
 - pulsar, 58
- iinfo
 - jpl_eph_data, 38
- imag
 - complexVal, 31
- inc_bin
 - gwSrc, 36
 - gwgeneralSrc, 34
- indexx8
 - TKspectrum.h, 129
- init_genrand
 - T2toolkit.h, 90
- initialise
 - tempo2.h, 111
- initialiseOne
 - tempo2.h, 111
- interpolation_info, 36
 - n_posn_avail, 36
 - n_vel_avail, 36
 - posn_coeff, 37
 - twot, 37
 - vel_coeff, 37
- ipm
 - pulsar, 58
- ipt
 - jpl_eph_data, 38
- JPL_EPH_FSEEK_ERROR
 - jpleph.h, 81
- JPL_EPH_INVALID_INDEX
 - jpleph.h, 81
- JPL_EPH_OUTSIDE_RANGE
 - jpleph.h, 81
- JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS
 - jpleph.h, 81
- JPL_EPH_READ_ERROR
 - jpleph.h, 81
- JPL_EPHEM_AU_IN_KM
 - jpleph.h, 81
- JPL_EPHEM_EARTH_MOON_RATIO
 - jpleph.h, 81
- JPL_EPHEM_END_JD
 - jpleph.h, 81
- JPL_EPHEM_EPHEMERIS_VERSION
 - jpleph.h, 81
- JPL_EPHEM_IPT_ARRAY
 - jpleph.h, 81
- JPL_EPHEM_KERNEL_NCOEFF
 - jpleph.h, 81
- JPL_EPHEM_KERNEL_RECORD_SIZE
 - jpleph.h, 81
- JPL_EPHEM_KERNEL_SIZE
 - jpleph.h, 81
- JPL_EPHEM_KERNEL_SWAP_BYTES
 - jpleph.h, 81
- JPL_EPHEM_N_CONSTANTS
 - jpleph.h, 81
- JPL_EPHEM_START_JD
 - jpleph.h, 81
- JPL_EPHEM_STEP
 - jpleph.h, 81
- JPL_EPHEMERIS
 - pulsar, 58
- JPL_HEADER_SIZE
 - jpl_int.h, 80
- JPL_INIT_FILE_CORRUPT
 - jpleph.h, 81
- JPL_INIT_FILE_NOT_FOUND
 - jpleph.h, 81
- JPL_INIT_FREAD2_FAILED
 - jpleph.h, 81
- JPL_INIT_FREAD3_FAILED
 - jpleph.h, 81
- JPL_INIT_FREAD4_FAILED
 - jpleph.h, 81
- JPL_INIT_FREAD5_FAILED
 - jpleph.h, 81
- JPL_INIT_FREAD_FAILED
 - jpleph.h, 81
- JPL_INIT_FSEEK_FAILED
 - jpleph.h, 81
- JPL_INIT_MEMORY_FAILURE
 - jpleph.h, 81
- JPL_INIT_NO_ERROR
 - jpleph.h, 82
- JPL_INIT_NOT_CALLED
 - jpleph.h, 82
- JVmodel
 - tempo2.h, 111
- jboFormat
 - pulsar, 58
- jpl_close_ephemeris
 - jpleph.h, 82
- jpl_eph_data, 37
 - au, 37
 - cache, 37
 - curr_cache_loc, 37
 - emrat, 37
 - ephem_end, 37
 - ephem_start, 37
 - ephem_step, 37
 - ephemeris_version, 38
 - ifile, 38
 - iinfo, 38
 - ipt, 38
 - kernel_size, 38
 - ncoeff, 38
 - ncon, 38
 - pvsun, 38

- pvsun_t, 38
 - recsize, 38
 - swap_bytes, 38
- jpl_get_constant
 - jpleph.h, 82
- jpl_get_double
 - jpleph.h, 82
- jpl_get_long
 - jpleph.h, 82
- jpl_get_pvsun
 - jpleph.h, 81
- jpl_init_ephemeris
 - jpleph.h, 82
- jpl_init_error_code
 - jpleph.h, 82
- jpl_int.h, 79
 - JPL_HEADER_SIZE, 80
 - MAX_CHEBY, 80
- jpl_pleph
 - jpleph.h, 82
- jpl_state
 - jpleph.h, 82
- jpleph.h, 80
 - DLL_FUNC, 81
 - JPL_EPH_FSEEK_ERROR, 81
 - JPL_EPH_INVALID_INDEX, 81
 - JPL_EPH_OUTSIDE_RANGE, 81
 - JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS, 81
 - JPL_EPH_READ_ERROR, 81
 - JPL_EPHEM_AU_IN_KM, 81
 - JPL_EPHEM_EARTH_MOON_RATIO, 81
 - JPL_EPHEM_END_JD, 81
 - JPL_EPHEM_EPHEMERIS_VERSION, 81
 - JPL_EPHEM_IPT_ARRAY, 81
 - JPL_EPHEM_KERNEL_NCOEFF, 81
 - JPL_EPHEM_KERNEL_RECORD_SIZE, 81
 - JPL_EPHEM_KERNEL_SIZE, 81
 - JPL_EPHEM_KERNEL_SWAP_BYTES, 81
 - JPL_EPHEM_N_CONSTANTS, 81
 - JPL_EPHEM_START_JD, 81
 - JPL_EPHEM_STEP, 81
 - JPL_INIT_FILE_CORRUPT, 81
 - JPL_INIT_FILE_NOT_FOUND, 81
 - JPL_INIT_FREAD2_FAILED, 81
 - JPL_INIT_FREAD3_FAILED, 81
 - JPL_INIT_FREAD4_FAILED, 81
 - JPL_INIT_FREAD5_FAILED, 81
 - JPL_INIT_FREAD_FAILED, 81
 - JPL_INIT_FSEEK_FAILED, 81
 - JPL_INIT_MEMORY_FAILURE, 81
 - JPL_INIT_NO_ERROR, 82
 - JPL_INIT_NOT_CALLED, 82
 - jpl_close_ephemeris, 82
 - jpl_get_constant, 82
 - jpl_get_double, 82
 - jpl_get_long, 82
 - jpl_get_pvsun, 81
 - jpl_init_ephemeris, 82
 - jpl_init_error_code, 82
 - jpl_pleph, 82
 - jpl_state, 82
 - make_sub_ephem, 82
- jump
 - observation, 42
- jumpStr
 - pulsar, 58
- jumpVal
 - pulsar, 58
- jumpValErr
 - pulsar, 58
- jupiter_earth
 - observation, 42
- kernel_size
 - jpl_eph_data, 38
- kg
 - gwSrc, 36
 - gwgeneralSrc, 34
- kind
 - T2Predictor, 69
- LD_PI
 - TKlongdouble.float128.h, 124
 - TKlongdouble.h, 125
 - TKlongdouble.ld.h, 126
- LEAPSECOND_FILE
 - tempo2.h, 99
- LOG_OUTFILE
 - TKlog.h, 122
- LONGDOUBLE_IS_FLOAT128
 - TKlongdouble.float128.h, 124
- LONGDOUBLE_IS_IEEE754
 - TKlongdouble.h, 125
 - TKlongdouble.ld.h, 126
- LONGDOUBLE_ONE
 - TKlongdouble.float128.h, 124
 - TKlongdouble.h, 125
 - TKlongdouble.ld.h, 126
- LT_OBJDIR
 - config.h, 73
- label
 - parameter, 47
 - tempo2.h, 106
- label_str
 - enum_str.h, 76
- latitude_grs80
 - observatory, 46
- ld_fprintf
 - TKlongdouble.float128.h, 125
 - TKlongdouble.h, 125
 - TKlongdouble.ld.h, 126
- ld_printf
 - TKlongdouble.float128.h, 125
 - TKlongdouble.h, 125
 - TKlongdouble.ld.h, 126
- ld_sprintf
 - TKlongdouble.float128.h, 125

- TKlongdouble.h, [125](#)
- TKlongdouble.ld.h, [126](#)
- libt2toolkit API, [27](#)
- libtempo2 External API, [28](#)
- linkFrom
 - parameter, [47](#)
- linkTo
 - parameter, [47](#)
- log10rms
 - T1Polyco, [68](#)
- logdbg
 - TKlog.h, [122](#)
- logerr
 - TKlog.h, [122](#)
- logerr_check
 - TKlog.h, [123](#)
- logicFlag
 - tempo2.h, [111](#)
- logmsg
 - TKlog.h, [122](#)
- logtchk
 - TKlog.h, [122](#)
- logwarn
 - TKlog.h, [122](#)
- longdouble
 - TKlongdouble.float128.h, [124](#)
 - TKlongdouble.h, [125](#), [126](#)
 - TKlongdouble.ld.h, [126](#), [127](#)
- longitude_grs80
 - observatory, [46](#)
- lookup_observatory_alias
 - tempo2.h, [111](#)
- MASYR2RADS
 - tempo2.h, [99](#)
- MAX
 - TKspectrum.h, [129](#)
- MAX_BPJ_JUMPS
 - tempo2.h, [99](#)
- MAX_CHEBY
 - jpl_int.h, [80](#)
- MAX_CLK_CORR
 - tempo2.h, [99](#)
- MAX_CLKCORR
 - tempo2.h, [100](#)
- MAX_COEFF
 - tempo2.h, [100](#)
- MAX_COMPANIONS
 - tempo2.h, [100](#)
- MAX_DM_DERIVATIVES
 - tempo2.h, [100](#)
- MAX_DMx
 - tempo2.h, [100](#)
- MAX_FILELEN
 - tempo2.h, [100](#)
- MAX_FIT
 - tempo2.h, [100](#)
- MAX_FLAG_LEN
 - tempo2.h, [100](#)
- MAX_FLAGS
 - tempo2.h, [100](#)
- MAX_FREQ_DERIVATIVES
 - tempo2.h, [100](#)
- MAX_IFUNC
 - tempo2.h, [100](#)
- MAX_JUMPS
 - tempo2.h, [100](#)
- MAX_LEAPSEC
 - tempo2.h, [101](#)
- MAX_MSG
 - tempo2.h, [101](#)
- MAX_OBSN
 - tempo2.h, [114](#)
- MAX_OBSN_VAL
 - tempo2.h, [101](#)
- MAX_PARAMS
 - tempo2.h, [101](#)
- MAX_PSR
 - tempo2.h, [114](#)
- MAX_PSR_VAL
 - tempo2.h, [101](#)
- MAX_QUAD
 - tempo2.h, [101](#)
- MAX_SITE
 - tempo2.h, [101](#)
- MAX_STOREPRECISION
 - tempo2.h, [101](#)
- MAX_STRLEN
 - tempo2.h, [101](#)
- MAX_T2EFAC
 - tempo2.h, [101](#)
- MAX_T2EQUAD
 - tempo2.h, [101](#)
- MAX_TEL_CLK_OFFS
 - tempo2.h, [101](#)
- MAX_TEL_DX
 - tempo2.h, [102](#)
- MAX_TEL_DY
 - tempo2.h, [102](#)
- MAX_TEL_DZ
 - tempo2.h, [102](#)
- MAX_TNBN
 - tempo2.h, [102](#)
- MAX_TNDMEv
 - tempo2.h, [102](#)
- MAX_TNECORR
 - tempo2.h, [102](#)
- MAX_TNEF
 - tempo2.h, [102](#)
- MAX_TNEQ
 - tempo2.h, [102](#)
- MAX_TNGN
 - tempo2.h, [102](#)
- MAX_TNSQ
 - tempo2.h, [102](#)
- MAX_TOFFSET
 - tempo2.h, [102](#)

MAX_WHITE
 tempo2.h, 102
 MIN
 TKspectrum.h, 129
 MSSmodel
 tempo2.h, 111
 make_sub_ephem
 jpleph.h, 82
 malloc_2df
 TKmatrix.h, 127
 malloc_blas
 TKmatrix.h, 127
 malloc_uinv
 TKmatrix.h, 127
 mat20
 TKspectrum.h, 129
 matrixMult
 GWsim.h, 78
 minPrec
 storePrecision, 67
 mjd_end
 ChebyModel, 30
 mjd_mid
 T1Polyco, 68
 mjd_start
 ChebyModel, 30
 modelset
 T2Predictor, 69

 n_posn_avail
 interpolation_info, 36
 n_vel_avail
 interpolation_info, 36
 nCompanion
 pulsar, 59
 nConstraints
 FitInfo, 32
 nDMEvents
 pulsar, 59
 NE_SW_DEFAULT
 tempo2.h, 102
 NEWFIT
 tempo2.h, 114
 nFit
 pulsar, 59
 nFlags
 observation, 42
 nGlobal
 pulsar, 59
 nJumps
 pulsar, 59
 nLinkFrom
 parameter, 47
 nLinkTo
 parameter, 47
 nParam
 pulsar, 59
 nParams
 FitInfo, 32

 nPhaseJump
 pulsar, 60
 nQuad
 pulsar, 60
 nStorePrecision
 pulsar, 60
 nT2efac
 pulsar, 60
 nT2equad
 pulsar, 60
 nTNBandNoise
 pulsar, 60
 nTNECORR
 pulsar, 60
 nTNEF
 pulsar, 60
 nTNEQ
 pulsar, 60
 nTNGroupNoise
 pulsar, 60
 nTNSQ
 pulsar, 60
 nTNShapeletEvents
 pulsar, 60
 nTelDX
 pulsar, 60
 nTelDY
 pulsar, 60
 nTelDZ
 pulsar, 60
 nToffset
 pulsar, 60
 nWhite
 pulsar, 60
 nWhite_dm
 pulsar, 60
 nallocated
 DynamicArray, 32
 name
 observatory, 46
 pulsar, 59
 nclock_correction
 observation, 42
 ncoeff
 jpl_eph_data, 38
 T1Polyco, 68
 ncon
 jpl_eph_data, 38
 nconstraints
 pulsar, 59
 ndmx
 pulsar, 59
 ne_sw
 pulsar, 59
 nelem
 DynamicArray, 32
 neptune_earth
 observation, 42

- nits
 - pulsar, [59](#)
- noWarnings
 - pulsar, [59](#)
- nobs
 - pulsar, [59](#)
- NonePredType
 - tempo2pred.h, [116](#)
- nphase
 - observation, [42](#)
- nsegments
 - ChebyModelSet, [30](#)
 - T1PolycoSet, [68](#)
- nutations
 - observation, [42](#)
- nx
 - Cheby2D, [29](#)
- ny
 - Cheby2D, [29](#)
- OBLQ
 - tempo2.h, [103](#)
- OBSSYS_FILE
 - tempo2.h, [103](#)
- obsNjump
 - observation, [42](#)
- observation, [38](#)
 - addedNoise, [40](#)
 - averagebat, [40](#)
 - averagedmbat, [40](#)
 - averagedmerr, [40](#)
 - averagedmres, [40](#)
 - averageerr, [40](#)
 - averageres, [40](#)
 - bat, [40](#)
 - batCorr, [40](#)
 - bbat, [40](#)
 - clockCorr, [40](#)
 - correctionTT_TB, [40](#)
 - correctionTT_Teph, [40](#)
 - correctionTT_calcEph, [40](#)
 - correctionUT1, [41](#)
 - correctionsTT, [40](#)
 - delayCorr, [41](#)
 - deleted, [41](#)
 - earth_ssb, [41](#)
 - earthMoonBary_earth, [41](#)
 - earthMoonBary_ssb, [41](#)
 - efac, [41](#)
 - einsteinRate, [41](#)
 - equad, [41](#)
 - flagID, [41](#)
 - flagVal, [41](#)
 - fname, [41](#)
 - freq, [41](#)
 - freqSSB, [42](#)
 - jump, [42](#)
 - jupiter_earth, [42](#)
 - nFlags, [42](#)
 - nclock_correction, [42](#)
 - neptune_earth, [42](#)
 - nphase, [42](#)
 - nutations, [42](#)
 - obsNjump, [42](#)
 - observatory_earth, [42](#)
 - origErr, [42](#)
 - origsat, [42](#)
 - pet, [42](#)
 - phase, [42](#)
 - phaseOffset, [43](#)
 - planet_ssb, [43](#)
 - planet_ssb_derv, [43](#)
 - planet_ssb_tmr, [43](#)
 - prefitResidual, [43](#)
 - psrPos, [43](#)
 - pulseN, [43](#)
 - residual, [43](#)
 - roemer, [43](#)
 - sat, [43](#)
 - sat_day, [43](#)
 - sat_sec, [43](#)
 - saturn_earth, [43](#)
 - shapiroDelayJupiter, [43](#)
 - shapiroDelayNeptune, [43](#)
 - shapiroDelaySaturn, [44](#)
 - shapiroDelaySun, [44](#)
 - shapiroDelayUranus, [44](#)
 - shapiroDelayVenus, [44](#)
 - shklovskii, [44](#)
 - siteVel, [44](#)
 - sun_earth, [44](#)
 - sun_ssb, [44](#)
 - TNDMErr, [44](#)
 - TNDMSignal, [45](#)
 - TNGroupErr, [45](#)
 - TNGroupSignal, [45](#)
 - TNRedErr, [45](#)
 - TNRedSignal, [45](#)
 - tdis1, [44](#)
 - tdis2, [44](#)
 - tellID, [44](#)
 - tempo2.h, [104](#)
 - toaDMErr, [45](#)
 - toaErr, [45](#)
 - torb, [45](#)
 - troposphericDelay, [45](#)
 - uranus_earth, [45](#)
 - venus_earth, [45](#)
 - zenith, [45](#)
- observatory, [46](#)
 - clock_name, [46](#)
 - code, [46](#)
 - height_grs80, [46](#)
 - latitude_grs80, [46](#)
 - longitude_grs80, [46](#)
 - name, [46](#)
 - x, [46](#)

- y, [46](#)
 - z, [46](#)
- observatory_earth
 - observation, [42](#)
- obsn
 - pulsar, [60](#)
- offset
 - pulsar, [60](#)
- offset_e
 - pulsar, [60](#)
- omega_g
 - gwSrc, [36](#)
 - gwgeneralSrc, [34](#)
- open_file
 - read_fortran.h, [83](#)
- open_file2
 - read_fortran2.h, [83](#)
- origErr
 - observation, [42](#)
- origsat
 - observation, [42](#)
- outputTMatrix
 - pulsar, [60](#)
- PACKAGE
 - config.h, [73](#)
- PACKAGE_BUGREPORT
 - config.h, [73](#)
- PACKAGE_NAME
 - config.h, [73](#)
- PACKAGE_STRING
 - config.h, [73](#)
- PACKAGE_TARNAME
 - config.h, [74](#)
- PACKAGE_URL
 - config.h, [74](#)
- PACKAGE_VERSION
 - config.h, [74](#)
- PCM
 - tempo2.h, [103](#)
- param
 - pulsar, [61](#)
- param_JUMP
 - tempo2.h, [109](#)
- param_LAST
 - tempo2.h, [109](#)
- param_ZERO
 - tempo2.h, [109](#)
- param_a0
 - tempo2.h, [108](#)
- param_a1
 - tempo2.h, [107](#)
- param_a1dot
 - tempo2.h, [107](#)
- param_a2dot
 - tempo2.h, [107](#)
- param_afac
 - tempo2.h, [108](#)
- param_b0
 - tempo2.h, [108](#)
- param_bp
 - tempo2.h, [107](#)
- param_bpja1
 - tempo2.h, [108](#)
- param_bpjec
 - tempo2.h, [108](#)
- param_bpjep
 - tempo2.h, [108](#)
- param_bpjom
 - tempo2.h, [108](#)
- param_bpjpb
 - tempo2.h, [108](#)
- param_bpjph
 - tempo2.h, [108](#)
- param_bpp
 - tempo2.h, [107](#)
- param_brake
 - tempo2.h, [109](#)
- param_cgw
 - tempo2.h, [108](#)
- param_clk_offs
 - tempo2.h, [108](#)
- param_daop
 - tempo2.h, [108](#)
- param_decj
 - tempo2.h, [106](#)
- param_df1
 - tempo2.h, [109](#)
- param_dm
 - tempo2.h, [107](#)
- param_dm_cos1yr
 - tempo2.h, [109](#)
- param_dm_sin1yr
 - tempo2.h, [109](#)
- param_dmassplanet
 - tempo2.h, [108](#)
- param_dmepoch
 - tempo2.h, [106](#)
- param_dmmodel
 - tempo2.h, [108](#)
- param_dmx
 - tempo2.h, [108](#)
- param_dmxr1
 - tempo2.h, [108](#)
- param_dmxr2
 - tempo2.h, [108](#)
- param_dphaseplanet
 - tempo2.h, [108](#)
- param_dr
 - tempo2.h, [107](#)
- param_dshk
 - tempo2.h, [108](#)
- param_dth
 - tempo2.h, [108](#)
- param_dtheta
 - tempo2.h, [108](#)
- param_e2dot
 - tempo2.h, [108](#)

tempo2.h, [107](#)
param_ecc
tempo2.h, [107](#)
param_edot
tempo2.h, [107](#)
param_ephver
tempo2.h, [108](#)
param_eps1
tempo2.h, [107](#)
param_eps1dot
tempo2.h, [108](#)
param_eps2
tempo2.h, [107](#)
param_eps2dot
tempo2.h, [108](#)
param_f
tempo2.h, [106](#)
param_fb
tempo2.h, [107](#)
param_fd
tempo2.h, [107](#)
param_fddc
tempo2.h, [107](#)
param_fddi
tempo2.h, [107](#)
param_finish
tempo2.h, [107](#)
param_gamma
tempo2.h, [107](#)
param_glep
tempo2.h, [107](#)
param_glf0
tempo2.h, [107](#)
param_glf0d
tempo2.h, [107](#)
param_glf1
tempo2.h, [107](#)
param_glf2
tempo2.h, [107](#)
param_glph
tempo2.h, [107](#)
param_gltd
tempo2.h, [107](#)
param_gwb_amp
tempo2.h, [109](#)
param_gwecc
tempo2.h, [109](#)
param_gwm_amp
tempo2.h, [109](#)
param_gwsingle
tempo2.h, [108](#)
param_h3
tempo2.h, [108](#)
param_h4
tempo2.h, [108](#)
param_ifunc
tempo2.h, [108](#)
param_iperharm
tempo2.h, [108](#)
param_kin
tempo2.h, [108](#)
param_kom
tempo2.h, [108](#)
param_label
tempo2.h, [104](#)
param_m2
tempo2.h, [107](#)
param_mtot
tempo2.h, [107](#)
param_nharm
tempo2.h, [108](#)
param_om
tempo2.h, [107](#)
param_om2dot
tempo2.h, [107](#)
param_omdot
tempo2.h, [107](#)
param_orbpx
tempo2.h, [107](#)
param_pb
tempo2.h, [107](#)
param_pbdot
tempo2.h, [107](#)
param_pepoch
tempo2.h, [106](#)
param_pmdec
tempo2.h, [107](#)
param_pmra
tempo2.h, [107](#)
param_pmrsv
tempo2.h, [107](#)
param_posepoch
tempo2.h, [106](#)
param_px
tempo2.h, [107](#)
param_quad_ifunc_c
tempo2.h, [109](#)
param_quad_ifunc_p
tempo2.h, [109](#)
param_quad_om
tempo2.h, [108](#)
param_raj
tempo2.h, [106](#)
param_shapmax
tempo2.h, [108](#)
param_sini
tempo2.h, [107](#)
param_start
tempo2.h, [107](#)
param_stateSwitchT
tempo2.h, [109](#)
param_stig
tempo2.h, [108](#)
param_t0
tempo2.h, [107](#)
param_tasc

- tempo2.h, 107
- param_tel_dx
 - tempo2.h, 109
- param_tel_dy
 - tempo2.h, 109
- param_tel_dz
 - tempo2.h, 109
- param_tel_vx
 - tempo2.h, 109
- param_tel_vy
 - tempo2.h, 109
- param_tel_vz
 - tempo2.h, 109
- param_tel_x0
 - tempo2.h, 109
- param_tel_y0
 - tempo2.h, 109
- param_tel_z0
 - tempo2.h, 109
- param_telEpoch
 - tempo2.h, 109
- param_telx
 - tempo2.h, 108
- param_tely
 - tempo2.h, 108
- param_telz
 - tempo2.h, 108
- param_track
 - tempo2.h, 107
- param_tres
 - tempo2.h, 108
- param_tspan
 - tempo2.h, 108
- param_tzrfreq
 - tempo2.h, 107
- param_tzrmjd
 - tempo2.h, 107
- param_wave_dm
 - tempo2.h, 108
- param_wave_om
 - tempo2.h, 108
- param_waveepoch
 - tempo2.h, 108
- param_waveepoch_dm
 - tempo2.h, 108
- param_xomdot
 - tempo2.h, 108
- param_xpbdot
 - tempo2.h, 107
- paramCounters
 - FitInfo, 32
- paramDerivFunc
 - tempo2.h, 104
- paramDerivs
 - FitInfo, 32
- paramIndex
 - FitInfo, 33
- paramSet
 - parameter, 47
 - paramUpdateFunc
 - tempo2.h, 105
 - parameter, 46
 - aSize, 47
 - err, 47
 - fitFlag, 47
 - label, 47
 - linkFrom, 47
 - linkTo, 47
 - nLinkFrom, 47
 - nLinkTo, 47
 - paramSet, 47
 - prefit, 47
 - prefitErr, 48
 - shortlabel, 48
 - tempo2.h, 104
 - val, 48
 - parse_longdouble
 - TKlongdouble.float128.h, 125
 - TKlongdouble.h, 126
 - TKlongdouble.ld.h, 127
 - passStr
 - pulsar, 61
 - pet
 - observation, 42
 - phase
 - observation, 42
 - phase_g
 - gwSrc, 36
 - gwgeneralSrc, 34
 - phaseJump
 - pulsar, 61
 - phaseJumpDir
 - pulsar, 61
 - phaseJumpID
 - pulsar, 61
 - phaseOffset
 - observation, 43
 - phi_bin
 - gwSrc, 36
 - gwgeneralSrc, 34
 - phi_g
 - gwSrc, 36
 - gwgeneralSrc, 34
 - phi_polar_g
 - gwSrc, 36
 - gwgeneralSrc, 34
 - planet_ssb
 - observation, 43
 - planet_ssb_derv
 - observation, 43
 - planet_ssb_tmr
 - observation, 43
 - planetShapiro
 - pulsar, 61
 - polycos
 - tempo2.h, 111

- posPulsar
 - pulsar, [61](#)
- posn_coeff
 - interpolation_info, [37](#)
- powl
 - TKlongdouble.float128.h, [124](#)
- preProcess
 - tempo2.h, [111](#)
- preProcessSimple
 - tempo2.h, [111](#)
- preProcessSimple1
 - tempo2.h, [111](#)
- preProcessSimple2
 - tempo2.h, [111](#)
- preProcessSimple3
 - tempo2.h, [112](#)
- prefit
 - parameter, [47](#)
- prefitErr
 - parameter, [48](#)
- prefitResidual
 - observation, [43](#)
- processFlag
 - tempo2.h, [112](#)
- processSimultaneous
 - tempo2.h, [112](#)
- psrPos
 - observation, [43](#)
- psrangle
 - GWsim.h, [78](#)
- psrname
 - ChebyModel, [30](#)
 - T1Polyco, [68](#)
- pulsar, [48](#)
 - addTNGlobalEQ, [54](#)
 - auto_constraints, [54](#)
 - AverageDMResiduals, [54](#)
 - AverageEpochWidth, [54](#)
 - AverageFlag, [54](#)
 - AverageResiduals, [54](#)
 - binaryModel, [54](#)
 - bootStrap, [54](#)
 - calcShapiro, [54](#)
 - cgw_angpol, [54](#)
 - cgw_cosinc, [54](#)
 - cgw_h0, [54](#)
 - cgw_mc, [54](#)
 - clk_offsE, [54](#)
 - clk_offsT, [54](#)
 - clk_offsV, [54](#)
 - clkOffsN, [54](#)
 - clock, [54](#)
 - clockFromOverride, [54](#)
 - constraints, [54](#)
 - correctTroposphere, [55](#)
 - covar, [55](#)
 - decjStrPost, [55](#)
 - decjStrPre, [55](#)
 - decsim, [55](#)
 - deleteFileName, [55](#)
 - dilateFreq, [55](#)
 - dmOffset, [55](#)
 - dmoftsCM, [55](#)
 - dmoftsCM_error, [55](#)
 - dmoftsCM_mjd, [55](#)
 - dmoftsCM_weight, [55](#)
 - dmoftsCMnum, [55](#)
 - dmoftsDM, [55](#)
 - dmoftsDM_error, [55](#)
 - dmoftsDM_mjd, [55](#)
 - dmoftsDM_weight, [55](#)
 - dmoftsDMnum, [55](#)
 - eclCoord, [55](#)
 - eopc04_file, [56](#)
 - ephemeris, [56](#)
 - filterStr, [56](#)
 - fitChisq, [56](#)
 - fitFunc, [56](#)
 - fitJump, [56](#)
 - fitMode, [56](#)
 - fitNfree, [56](#)
 - fitParamGlobal, [56](#)
 - fitParamGlobalK, [56](#)
 - fitParamI, [56](#)
 - fitParamK, [56](#)
 - fitinfo, [56](#)
 - fixedFormat, [56](#)
 - fjumpID, [56](#)
 - globalNfit, [56](#)
 - globalNoConstrain, [56](#)
 - gwb_decj, [57](#)
 - gwb_epoch, [57](#)
 - gwb_geom_c, [57](#)
 - gwb_geom_p, [57](#)
 - gwb_raj, [57](#)
 - gwb_width, [57](#)
 - gwecc_dec, [57](#)
 - gwecc_distance, [57](#)
 - gwecc_e, [57](#)
 - gwecc_epoch, [57](#)
 - gwecc_inc, [57](#)
 - gwecc_m1, [57](#)
 - gwecc_m2, [57](#)
 - gwecc_nodes_orientation, [57](#)
 - gwecc_orbital_period, [57](#)
 - gwecc_psrdist, [57](#)
 - gwecc_pulsarTermOn, [57](#)
 - gwecc_ra, [57](#)
 - gwecc_redshift, [57](#)
 - gwecc_theta_0, [57](#)
 - gwecc_theta_nodes, [57](#)
 - gwm_decj, [57](#)
 - gwm_dphase, [57](#)
 - gwm_epoch, [57](#)
 - gwm_phi, [57](#)
 - gwm_raj, [57](#)

gwsrc_across_i, 57
 gwsrc_across_i_e, 58
 gwsrc_across_r, 58
 gwsrc_across_r_e, 58
 gwsrc_aplus_i, 58
 gwsrc_aplus_i_e, 58
 gwsrc_aplus_r, 58
 gwsrc_aplus_r_e, 58
 gwsrc_dec, 58
 gwsrc_epoch, 58
 gwsrc_psrdis, 58
 gwsrc_ra, 58
 ifunc_weights, 58
 ifuncE, 58
 ifuncN, 58
 ifuncT, 58
 ifuncV, 58
 ipm, 58
 JPL_EPHEMERIS, 58
 jboFormat, 58
 jumpStr, 58
 jumpVal, 58
 jumpValErr, 58
 nCompanion, 59
 nDMEvents, 59
 nFit, 59
 nGlobal, 59
 nJumps, 59
 nParam, 59
 nPhaseJump, 60
 nQuad, 60
 nStorePrecision, 60
 nT2efac, 60
 nT2equad, 60
 nTNBandNoise, 60
 nTNECORR, 60
 nTNEF, 60
 nTNEQ, 60
 nTNGroupNoise, 60
 nTNSQ, 60
 nTNShapeletEvents, 60
 nTelDX, 60
 nTelDY, 60
 nTelDZ, 60
 nToffset, 60
 nWhite, 60
 nWhite_dm, 60
 name, 59
 nconstraints, 59
 ndmx, 59
 ne_sw, 59
 nits, 59
 noWarnings, 59
 nob, 59
 obsn, 60
 offset, 60
 offset_e, 60
 outputTMatrix, 60
 param, 61
 passStr, 61
 phaseJump, 61
 phaseJumpDir, 61
 phaseJumpID, 61
 planetShapiro, 61
 posPulsar, 61
 quad_across_i, 61
 quad_across_i_e, 61
 quad_across_r, 61
 quad_across_r_e, 61
 quad_aplus_i, 61
 quad_aplus_i_e, 61
 quad_aplus_r, 61
 quad_aplus_r_e, 61
 quad_ifunc_c_DEC, 61
 quad_ifunc_c_RA, 61
 quad_ifunc_geom_c, 61
 quad_ifunc_geom_p, 61
 quad_ifunc_p_DEC, 61
 quad_ifunc_p_RA, 62
 quad_ifuncE_c, 62
 quad_ifuncE_p, 62
 quad_ifuncN_c, 62
 quad_ifuncN_p, 62
 quad_ifuncT_c, 62
 quad_ifuncT_p, 62
 quad_ifuncV_c, 62
 quad_ifuncV_p, 62
 quadDEC, 62
 quadEpoch, 62
 quadRA, 62
 rajStrPost, 62
 rajStrPre, 62
 rasim, 62
 rescaleErrChisq, 62
 rmsPost, 62
 rmsPre, 62
 robust, 62
 setTelVelX, 62
 setTelVelY, 62
 setTelVelZ, 62
 setUnits, 62
 simflag, 62
 sorted, 62
 storePrec, 63
 swm, 63
 t2cMethod, 63
 T2efacFlagID, 63
 T2efacFlagVal, 63
 T2efacVal, 63
 T2equadFlagID, 63
 T2equadFlagVal, 63
 T2equadVal, 63
 T2globalEfac, 63
 TNBandDMAmp, 64
 TNBandDMC, 64
 TNBandDMGam, 64

TNBandNoiseAmp, 64
 TNBandNoiseC, 64
 TNBandNoiseGam, 64
 TNBandNoiseHF, 64
 TNBandNoiseLF, 64
 TNDMAmp, 64
 TNDMC, 64
 TNDMCoeffs, 64
 TNDMEvAmp, 64
 TNDMEvGam, 64
 TNDMEvLength, 64
 TNDMEvLin, 64
 TNDMEvOff, 64
 TNDMEvQuad, 64
 TNDMEvStart, 64
 TNDMGam, 64
 TNECORRFlagID, 64
 TNECORRFlagVal, 64
 TNECORRVal, 64
 TNEFFlagID, 64
 TNEFFlagVal, 65
 TNEFVal, 65
 TNEQFlagID, 65
 TNEQFlagVal, 65
 TNEQVal, 65
 TNGlobalEF, 65
 TNGlobalEQ, 65
 TNGroupNoiseAmp, 65
 TNGroupNoiseC, 65
 TNGroupNoiseFlagID, 65
 TNGroupNoiseFlagVal, 65
 TNGroupNoiseGam, 65
 TNRedAmp, 65
 TNRedC, 65
 TNRedCoeffs, 65
 TNRedCorner, 65
 TNRedFLow, 65
 TNRedGam, 65
 TNSQFlagID, 65
 TNSQFlagVal, 65
 TNSQVal, 65
 TNShapeletEvFScale, 65
 TNShapeletEvN, 65
 TNShapeletEvPos, 65
 TNShapeletEvWidth, 65
 TNsubtractDM, 65
 TNsubtractRed, 65
 tOffset, 66
 tOffset_f1, 66
 tOffset_f2, 66
 tOffset_t1, 66
 tOffset_t2, 66
 tOffsetFlags, 66
 tOffsetSite, 66
 telDX_e, 63
 telDX_t, 63
 telDX_v, 63
 telDX_vel, 63
 telDX_vel_e, 63
 telDY_e, 63
 telDY_t, 63
 telDY_v, 63
 telDY_vel, 63
 telDY_vel_e, 63
 telDZ_e, 63
 telDZ_t, 63
 telDZ_v, 63
 telDZ_vel, 63
 telDZ_vel_e, 64
 tempo1, 64
 tempo2.h, 105
 timeEphemeris, 64
 ToAextraCovar, 65
 tzrsite, 66
 units, 66
 useCalceph, 66
 useTNorth, 66
 velPulsar, 66
 wave_cos, 66
 wave_cos_dm, 66
 wave_cos_dm_err, 66
 wave_cos_err, 66
 wave_sine, 66
 wave_sine_dm, 66
 wave_sine_dm_err, 66
 wave_sine_err, 66
 waveScale, 66
 whiteNoiseModelFile, 66
 pulseN
 observation, 43
 pvsun
 jpl_eph_data, 38
 pvsun_t
 jpl_eph_data, 38
 quad_across_i
 pulsar, 61
 quad_across_i_e
 pulsar, 61
 quad_across_r
 pulsar, 61
 quad_across_r_e
 pulsar, 61
 quad_aplus_i
 pulsar, 61
 quad_aplus_i_e
 pulsar, 61
 quad_aplus_r
 pulsar, 61
 quad_aplus_r_e
 pulsar, 61
 quad_ifunc_c_DEC
 pulsar, 61
 quad_ifunc_c_RA
 pulsar, 61
 quad_ifunc_geom_c
 pulsar, 61

- quad_ifunc_geom_p
 - pulsar, 61
- quad_ifunc_p_DEC
 - pulsar, 61
- quad_ifunc_p_RA
 - pulsar, 62
- quad_ifuncE_c
 - pulsar, 62
- quad_ifuncE_p
 - pulsar, 62
- quad_ifuncN_c
 - pulsar, 62
- quad_ifuncN_p
 - pulsar, 62
- quad_ifuncT_c
 - pulsar, 62
- quad_ifuncT_p
 - pulsar, 62
- quad_ifuncV_c
 - pulsar, 62
- quad_ifuncV_p
 - pulsar, 62
- quadDEC
 - pulsar, 62
- quadEpoch
 - pulsar, 62
- quadRA
 - pulsar, 62
- README.md, 84
- RESETCOLOR
 - TKlog.h, 123
- rajStrPost
 - pulsar, 62
- rajStrPre
 - pulsar, 62
- rasim
 - pulsar, 62
- read_char
 - read_fortran.h, 83
- read_character
 - read_fortran.h, 83
- read_character2
 - read_fortran2.h, 83
- read_double
 - read_fortran.h, 83
- read_double2
 - read_fortran2.h, 83
- read_float
 - read_fortran.h, 83
- read_float2
 - read_fortran2.h, 84
- read_fortran.h, 82
 - c_fileptr, 83
 - close_file, 83
 - open_file, 83
 - read_char, 83
 - read_character, 83
 - read_double, 83
 - read_float, 83
 - read_int, 83
 - read_record_int, 83
 - swapByte, 83
- read_fortran2.h, 83
 - c_fileptr2, 84
 - close_file2, 83
 - open_file2, 83
 - read_character2, 83
 - read_double2, 83
 - read_float2, 84
 - read_int2, 84
 - read_record_int2, 84
 - swapByte2, 84
- read_int
 - read_fortran.h, 83
- read_int2
 - read_fortran2.h, 84
- read_record_int
 - read_fortran.h, 83
- read_record_int2
 - read_fortran2.h, 84
- readEphemeris
 - tempo2.h, 112
- readEphemeris_calceph
 - tempo2.h, 112
- readJBO_bat
 - tempo2.h, 112
- readObsFile
 - tempo2.h, 112
- readOneEphemeris
 - tempo2.h, 112
- readParfile
 - tempo2.h, 112
- readParfileGlobal
 - tempo2.h, 112
- readSimpleParfile
 - tempo2.h, 112
- readTimfile
 - tempo2.h, 112
- real
 - complexVal, 31
- recordPrecision
 - tempo2.h, 112
- resize
 - jpl_eph_data, 38
- reference_phase
 - T1Polyco, 68
- rescaleErrChisq
 - pulsar, 62
- residual
 - observation, 43
- rmsPost
 - pulsar, 62
- rmsPre
 - pulsar, 62
- robust
 - pulsar, 62

- roemer
 - observation, [43](#)
- routine
 - storePrecision, [67](#)
- Rs
 - GWsim.h, [78](#)
- SECDAY
 - tempo2.h, [103](#)
- SECDAYI
 - tempo2.h, [103](#)
- SI_UNITS
 - tempo2.h, [103](#)
- SOLAR_MASS
 - tempo2.h, [103](#)
- SOLAR_RADIUS
 - tempo2.h, [103](#)
- SPEED_LIGHT
 - tempo2.h, [103](#)
- STDC_HEADERS
 - config.h, [74](#)
- samples
 - TabulatedFunction, [70](#)
- sat
 - observation, [43](#)
- sat_day
 - observation, [43](#)
- sat_sec
 - observation, [43](#)
- saturn_earth
 - observation, [43](#)
- secularMotion
 - tempo2.h, [112](#)
- segments
 - ChebyModelSet, [30](#)
 - T1PolycoSet, [68](#)
- setPlugPath
 - tempo2.h, [112](#)
- setStart
 - tempo2.h, [112](#)
- setTelVelX
 - pulsar, [62](#)
- setTelVelY
 - pulsar, [62](#)
- setTelVelZ
 - pulsar, [62](#)
- setUnits
 - pulsar, [62](#)
- setupGW
 - GWsim.h, [78](#)
- setupParameterFileDefaults
 - tempo2.h, [112](#)
- setupPulsar_GWsim
 - GWsim.h, [78](#)
- setupgeneralGW
 - GWsim.h, [78](#)
- shapiro_delay
 - tempo2.h, [112](#)
- shapiroDelayJupiter
 - observation, [43](#)
- shapiroDelayNeptune
 - observation, [43](#)
- shapiroDelaySaturn
 - observation, [44](#)
- shapiroDelaySun
 - observation, [44](#)
- shapiroDelayUranus
 - observation, [44](#)
- shapiroDelayVenus
 - observation, [44](#)
- shklovskii
 - observation, [44](#)
- shortlabel
 - parameter, [48](#)
- simflag
 - pulsar, [62](#)
- simplePlot
 - tempo2.h, [112](#)
- sineFunc
 - TKspectrum.h, [129](#)
- sinefunc
 - t2fit_ifunc.h, [87](#)
- sinl
 - TKlongdouble.float128.h, [124](#)
- siteVel
 - observation, [44](#)
- sitename
 - ChebyModel, [30](#)
 - T1Polyco, [68](#)
- sl_alpha
 - gwgenSpec, [35](#)
- sl_amp
 - gwgenSpec, [35](#)
- solarWindModel
 - tempo2.h, [112](#)
- sortToAs
 - tempo2.h, [112](#)
- sorted
 - pulsar, [62](#)
- span
 - T1Polyco, [68](#)
- sphharm
 - GWsim.h, [78](#)
- st_alpha
 - gwgenSpec, [35](#)
- st_amp
 - gwgenSpec, [35](#)
- standardConstraintFunctions
 - constraints.h, [75](#)
- storePrec
 - pulsar, [63](#)
- storePrecision, [67](#)
 - comment, [67](#)
 - minPrec, [67](#)
 - routine, [67](#)
 - tempo2.h, [105](#)
- sun_earth

- observation, 44
- sun_ssb
 - observation, 44
- swap_bytes
 - jpl_eph_data, 38
- swapByte
 - read_fortran.h, 83
- swapByte2
 - read_fortran2.h, 84
- swm
 - pulsar, 63
- T1
 - tempo2pred.h, 116
- t1
 - T2Predictor, 69
- T1Polyco, 67
 - binary_frequency, 68
 - binary_phase, 68
 - coeff, 68
 - date_string, 68
 - dm, 68
 - doppler, 68
 - frequency_obs, 68
 - frequency_psr_0, 68
 - log10rms, 68
 - mjd_mid, 68
 - ncoeff, 68
 - psrname, 68
 - reference_phase, 68
 - sitename, 68
 - span, 68
 - utc_string, 68
- T1Polyco_GetFrequency
 - tempo2pred_int.h, 118
- T1Polyco_GetPhase
 - tempo2pred_int.h, 118
- T1Polyco_Read
 - tempo2pred_int.h, 119
- T1Polyco_Write
 - tempo2pred_int.h, 119
- T1PolycoSet, 68
 - nsegments, 68
 - segments, 68
- T1PolycoSet_Destroy
 - tempo2pred_int.h, 119
- T1PolycoSet_GetFrequency
 - tempo2pred_int.h, 119
- T1PolycoSet_GetNearest
 - tempo2pred_int.h, 119
- T1PolycoSet_GetPhase
 - tempo2pred_int.h, 119
- T1PolycoSet_Read
 - tempo2pred_int.h, 119
- T1PolycoSet_Write
 - tempo2pred_int.h, 119
- T2_PTAmodel
 - tempo2.h, 112
- T2C_IAU2000B
 - tempo2.h, 103
- T2C_TEMPO
 - tempo2.h, 103
- t2Fit
 - t2fit.h, 85
- t2Fit_buildConstraintsMatrix
 - t2fit.h, 85
- t2Fit_buildDesignMatrix
 - t2fit.h, 85
- t2Fit_fillFitInfo
 - t2fit.h, 85
- t2Fit_fillGlobalFitInfo
 - t2fit.h, 85
- t2Fit_getFitData
 - t2fit.h, 85
- t2Fit_updateParameters
 - t2fit.h, 85
- t2FitFunc_binaryModels
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_dmmodelCM
 - t2fit_dmmodel.h, 86
- t2FitFunc_dmmodelDM
 - t2fit_dmmodel.h, 86
- t2FitFunc_dmsinusoids
 - t2fit_dmother.h, 86
- t2FitFunc_dmx
 - t2fit_dmother.h, 86
- t2FitFunc_fd
 - t2fit_dmother.h, 86
- t2FitFunc_fddc
 - t2fit_dmother.h, 86
- t2FitFunc_fitwaves
 - t2fit_fitwaves.h, 86
- t2FitFunc_ifunc
 - t2fit_ifunc.h, 87
- t2FitFunc_stdFitFuncs.h, 88
- t2FitFunc_jump
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_notImplemented
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_planet
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_sifunc
 - t2fit_ifunc.h, 87
- t2FitFunc_stdDm
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_stdFreq
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_stdGlitch
 - t2fit_glitch.h, 87
- t2FitFunc_stdGravWav
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_stdPosition
 - t2fit_position.h, 87
- t2FitFunc_telPos
 - t2fit_stdFitFuncs.h, 88
- t2FitFunc_zero
 - t2fit_stdFitFuncs.h, 88

- T2Predictor, [69](#)
 - cheby, [69](#)
 - kind, [69](#)
 - modelset, [69](#)
 - t1, [69](#)
- T2Predictor_Copy
 - tempo2pred.h, [116](#)
- T2Predictor_Destroy
 - tempo2pred.h, [116](#)
- T2Predictor_FRead
 - tempo2pred.h, [116](#)
- T2Predictor_FWrite
 - tempo2pred.h, [116](#)
- T2Predictor_GetEndFreq
 - tempo2pred.h, [116](#)
- T2Predictor_GetEndMJD
 - tempo2pred.h, [116](#)
- T2Predictor_GetFrequency
 - tempo2pred.h, [116](#)
- T2Predictor_GetPSRName
 - tempo2pred.h, [116](#)
- T2Predictor_GetPhase
 - tempo2pred.h, [116](#)
- T2Predictor_GetPlan
 - tempo2pred.h, [116](#)
- T2Predictor_GetPlan_Ext
 - tempo2pred.h, [116](#)
- T2Predictor_GetSiteName
 - tempo2pred.h, [116](#)
- T2Predictor_GetStartFreq
 - tempo2pred.h, [116](#)
- T2Predictor_GetStartMJD
 - tempo2pred.h, [116](#)
- T2Predictor_Init
 - tempo2pred.h, [116](#)
- T2Predictor_Insert
 - tempo2pred.h, [116](#)
- T2Predictor_Keep
 - tempo2pred.h, [116](#)
- T2Predictor_Kind
 - tempo2pred.h, [116](#)
- T2Predictor_Read
 - tempo2pred.h, [116](#)
- T2Predictor_Write
 - tempo2pred.h, [117](#)
- T2PredictorKind
 - tempo2pred.h, [116](#)
- t2UpdateFunc_binaryModels
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_dmmodelCM
 - t2fit_dmmodel.h, [86](#)
- t2UpdateFunc_dmmodelDM
 - t2fit_dmmodel.h, [86](#)
- t2UpdateFunc_fitwaves
 - t2fit_fitwaves.h, [86](#)
- t2UpdateFunc_ifunc
 - t2fit_ifunc.h, [87](#)
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_jump
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_notImplemented
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_planet
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_simpleAdd
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_simpleMinus
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_stdFreq
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_stdGlitch
 - t2fit_glitch.h, [87](#)
- t2UpdateFunc_stdGravWav
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_stdPosition
 - t2fit_position.h, [87](#)
- t2UpdateFunc_telPos
 - t2fit_stdFitFuncs.h, [89](#)
- t2UpdateFunc_zero
 - t2fit_stdFitFuncs.h, [89](#)
- T2accel.h, [84](#)
 - ACCEL_LSQ, [84](#)
 - ACCEL_MULTMATRIX, [84](#)
 - ACCEL_UINV, [84](#)
 - accel_lsq_qr, [84](#)
 - accel_multMatrix, [84](#)
 - accel_multMatrixVec, [84](#)
 - accel_uinv, [85](#)
 - useT2accel, [85](#)
- t2cMethod
 - pulsar, [63](#)
- T2efacFlagID
 - pulsar, [63](#)
- T2efacFlagVal
 - pulsar, [63](#)
- T2efacVal
 - pulsar, [63](#)
- T2equadFlagID
 - pulsar, [63](#)
- T2equadFlagVal
 - pulsar, [63](#)
- T2equadVal
 - pulsar, [63](#)
- t2fit.h, [85](#)
 - t2Fit, [85](#)
 - t2Fit_buildConstraintsMatrix, [85](#)
 - t2Fit_buildDesignMatrix, [85](#)
 - t2Fit_fillFitInfo, [85](#)
 - t2Fit_fillGlobalFitInfo, [85](#)
 - t2Fit_getFitData, [85](#)
 - t2Fit_updateParameters, [85](#)
- t2fit_dmmodel.h, [85](#)
 - t2FitFunc_dmmodelCM, [86](#)
 - t2FitFunc_dmmodelDM, [86](#)
 - t2UpdateFunc_dmmodelCM, [86](#)
 - t2UpdateFunc_dmmodelDM, [86](#)

- t2fit_dmother.h, 86
 - t2FitFunc_dmsinusoids, 86
 - t2FitFunc_dmx, 86
 - t2FitFunc_fd, 86
 - t2FitFunc_fddc, 86
- t2fit_fitwaves.h, 86
 - t2FitFunc_fitwaves, 86
 - t2UpdateFunc_fitwaves, 86
- t2fit_glitch.h, 86
 - t2FitFunc_stdGlitch, 87
 - t2UpdateFunc_stdGlitch, 87
- t2fit_ifunc.h, 87
 - ifunc, 87
 - sifunc, 87
 - t2FitFunc_ifunc, 87
 - t2FitFunc_sifunc, 87
 - t2UpdateFunc_ifunc, 87
- t2fit_position.h, 87
 - t2FitFunc_stdPosition, 87
 - t2UpdateFunc_stdPosition, 87
- t2fit_stdFitFuncs.h, 88
 - t2FitFunc_binaryModels, 88
 - t2FitFunc_ifunc, 88
 - t2FitFunc_jump, 88
 - t2FitFunc_notImplemented, 88
 - t2FitFunc_planet, 88
 - t2FitFunc_stdDm, 88
 - t2FitFunc_stdFreq, 88
 - t2FitFunc_stdGravWav, 88
 - t2FitFunc_telPos, 88
 - t2FitFunc_zero, 88
 - t2UpdateFunc_binaryModels, 89
 - t2UpdateFunc_ifunc, 89
 - t2UpdateFunc_jump, 89
 - t2UpdateFunc_notImplemented, 89
 - t2UpdateFunc_planet, 89
 - t2UpdateFunc_simpleAdd, 89
 - t2UpdateFunc_simpleMinus, 89
 - t2UpdateFunc_stdFreq, 89
 - t2UpdateFunc_stdGravWav, 89
 - t2UpdateFunc_telPos, 89
 - t2UpdateFunc_zero, 89
- T2globalEfac
 - pulsar, 63
- T2model
 - tempo2.h, 112
- T2toolkit.h, 89
 - genrand_int32, 90
 - genrand_real1, 90
 - init_genrand, 90
 - TKconvertFloat1, 90
 - TKconvertFloat2, 90
 - TKfindMax_d, 90
 - TKfindMax_f, 90
 - TKfindMedian_d, 90
 - TKfindMedian_f, 90
 - TKfindMin_d, 90
 - TKfindMin_f, 90
 - TKfindRMS_d, 90
 - TKfindRMS_f, 90
 - TKfindRMSweight_d, 90
 - TKgaussDev, 90
 - TKmean_d, 90
 - TKmean_f, 90
 - TKranDev, 90
 - TKrange_d, 91
 - TKrange_f, 91
 - TKretMax_d, 91
 - TKretMax_f, 91
 - TKretMin_d, 91
 - TKretMin_f, 91
 - TKretMin_i, 91
 - TKsetSeed, 91
 - TKsign_d, 91
 - TKsort_2f, 91
 - TKsort_3d, 91
 - TKsort_d, 91
 - TKsort_f, 91
 - TKvariance_d, 91
 - TKzeromean_d, 91
- TDB_UNITS
 - tempo2.h, 103
- TDBTDT_FILE
 - tempo2.h, 103
- TEMPO2_ARCH
 - config.h, 74
- TEMPO2_ENVIRON
 - tempo2.h, 114
- TEMPO2_ERROR
 - tempo2.h, 114
- TEMPO2_h_HASH
 - tempo2.h, 104
- TEMPO2_h_MAJOR_VER
 - tempo2.h, 104
- TEMPO2_h_MINOR_VER
 - tempo2.h, 104
- TEMPO2_h_VER
 - tempo2.h, 104
- TK_MAX_ERROR_LEN
 - TKlog.h, 123
- TK_MAX_ERRORS
 - TKlog.h, 123
- TK_STORE_ERROR
 - TKlog.h, 123
- TK_STORE_WARNING
 - TKlog.h, 123
- TK_dft
 - TKspectrum.h, 129
- TK_errorCount
 - TKlog.h, 123
- TK_errorlog
 - TKlog.h, 123
- TK_fft
 - TKspectrum.h, 129
- TK_fitSine
 - TKspectrum.h, 129

- TK_fitSinusoids
 - TKspectrum.h, [129](#)
- TK_warnCount
 - TKlog.h, [123](#)
- TK_warnlog
 - TKlog.h, [123](#)
- TK_weightLS
 - TKspectrum.h, [129](#)
- TKaveragePts
 - TKspectrum.h, [129](#)
- TKbacksubstitution_svd
 - TKsvd.h, [130](#)
- TKbidiagonal
 - TKsvd.h, [130](#)
- TKboxcar
 - TKspectrum.h, [129](#)
- TKcholesky.h, [119](#)
 - cholesky_covarFunc2matrix, [120](#)
 - cholesky_dmModel, [120](#)
 - cholesky_dmModelCovarParam, [120](#)
 - cholesky_ecm, [120](#)
 - cholesky_formUinv, [120](#)
 - cholesky_powerlawModel, [120](#)
 - cholesky_powerlawModel_withBeta, [120](#)
 - cholesky_readFromCovarianceFunction, [120](#)
- TKcmonot
 - TKspectrum.h, [129](#)
- TKconstrainedLeastSquares
 - TKfit.h, [120](#)
- TKconvertFloat1
 - T2toolkit.h, [90](#)
- TKconvertFloat2
 - T2toolkit.h, [90](#)
- TKfindMax_d
 - T2toolkit.h, [90](#)
- TKfindMax_f
 - T2toolkit.h, [90](#)
- TKfindMedian_d
 - T2toolkit.h, [90](#)
- TKfindMedian_f
 - T2toolkit.h, [90](#)
- TKfindMin_d
 - T2toolkit.h, [90](#)
- TKfindMin_f
 - T2toolkit.h, [90](#)
- TKfindPoly_d
 - TKfit.h, [121](#)
- TKfindRMS_d
 - T2toolkit.h, [90](#)
- TKfindRMS_f
 - T2toolkit.h, [90](#)
- TKfindRMSweight_d
 - T2toolkit.h, [90](#)
- TKfirstDifference
 - TKspectrum.h, [129](#)
- TKfit.h, [120](#)
 - TKconstrainedLeastSquares, [120](#)
 - TKfindPoly_d, [121](#)
 - TKfitPoly, [121](#)
 - TKleastSquares, [121](#)
 - TKleastSquares_svd, [121](#)
 - TKleastSquares_svd_noErr, [121](#)
 - TKremovePoly_d, [121](#)
 - TKremovePoly_f, [121](#)
 - TKrobustConstrainedLeastSquares, [121](#)
 - TKrobustLeastSquares, [121](#)
- TKfitPoly
 - TKfit.h, [121](#)
- TKgaussDev
 - T2toolkit.h, [90](#)
- TKhann
 - TKspectrum.h, [129](#)
- TKinterpolateSplineSmoothFixedXPts
 - TKspectrum.h, [129](#)
- TKleastSquares
 - TKfit.h, [121](#)
- TKleastSquares_svd
 - TKfit.h, [121](#)
- TKleastSquares_svd_noErr
 - TKfit.h, [121](#)
- TKlog.h, [121](#)
 - _LOG, [122](#)
 - _TKchklog, [123](#)
 - BOLDCOLOR, [122](#)
 - DEPRECATED, [122](#)
 - debugFlag, [123](#)
 - ENDERR, [122](#)
 - ENDL, [122](#)
 - ERRORCOLOR, [122](#)
 - LOG_OUTFILE, [122](#)
 - logdbg, [122](#)
 - logerr, [122](#)
 - logerr_check, [123](#)
 - logmsg, [122](#)
 - logtchk, [122](#)
 - logwarn, [122](#)
 - RESETCOLOR, [123](#)
 - TK_MAX_ERROR_LEN, [123](#)
 - TK_MAX_ERRORS, [123](#)
 - TK_STORE_ERROR, [123](#)
 - TK_STORE_WARNING, [123](#)
 - TK_errorCount, [123](#)
 - TK_errorlog, [123](#)
 - TK_warnCount, [123](#)
 - TK_warnlog, [123](#)
 - tcheck, [123](#)
 - timer_clk, [123](#)
 - WARNCOLOR, [123](#)
 - WHEREARG, [123](#)
 - WHEREERR, [123](#)
 - WHERESTR, [123](#)
 - WHERECHK, [123](#)
 - WHEREWARN, [123](#)
 - writeResiduals, [123](#)
- TKlomb_d
 - TKspectrum.h, [129](#)

- TKlongdouble.float128.h, [123](#)
 - cosl, [124](#)
 - FMT_LD, [124](#)
 - fabsl, [124](#)
 - floorl, [124](#)
 - LD_PI, [124](#)
 - LONGDOUBLE_IS_FLOAT128, [124](#)
 - LONGDOUBLE_ONE, [124](#)
 - ld_fprintf, [125](#)
 - ld_printf, [125](#)
 - ld_sprintf, [125](#)
 - longdouble, [124](#)
 - parse_longdouble, [125](#)
 - powl, [124](#)
 - sinl, [124](#)
 - USE_BUILTIN_LONGDOUBLE, [124](#)
- TKlongdouble.h, [125](#)
 - LD_PI, [125](#)
 - LONGDOUBLE_IS_IEEE754, [125](#)
 - LONGDOUBLE_ONE, [125](#)
 - ld_fprintf, [125](#)
 - ld_printf, [125](#)
 - ld_sprintf, [125](#)
 - longdouble, [125](#), [126](#)
 - parse_longdouble, [126](#)
 - USE_BUILTIN_LONGDOUBLE, [125](#)
- TKlongdouble.ld.h, [126](#)
 - LD_PI, [126](#)
 - LONGDOUBLE_IS_IEEE754, [126](#)
 - LONGDOUBLE_ONE, [126](#)
 - ld_fprintf, [126](#)
 - ld_printf, [126](#)
 - ld_sprintf, [126](#)
 - longdouble, [126](#), [127](#)
 - parse_longdouble, [127](#)
 - USE_BUILTIN_LONGDOUBLE, [126](#)
- TKmatrix.h, [127](#)
 - free_2df, [127](#)
 - free_blas, [127](#)
 - free_uinv, [127](#)
 - get_blas_cols, [127](#)
 - get_blas_rows, [127](#)
 - malloc_2df, [127](#)
 - malloc_blas, [127](#)
 - malloc_uinv, [127](#)
 - TKmultMatrix, [127](#)
 - TKmultMatrix_sq, [127](#)
 - TKmultMatrixVec, [127](#)
 - TKmultMatrixVec_sq, [127](#)
- TKmean_d
 - T2toolkit.h, [90](#)
- TKmean_f
 - T2toolkit.h, [90](#)
- TKmultMatrix
 - TKmatrix.h, [127](#)
- TKmultMatrix_sq
 - TKmatrix.h, [127](#)
- TKmultMatrixVec
 - TKmatrix.h, [127](#)
- TKmultMatrixVec_sq
 - TKmatrix.h, [127](#)
- TKpythag
 - TKsvd.h, [130](#)
- TKranDev
 - T2toolkit.h, [90](#)
- TKrange_d
 - T2toolkit.h, [91](#)
- TKrange_f
 - T2toolkit.h, [91](#)
- TKremovePoly_d
 - TKfit.h, [121](#)
- TKremovePoly_f
 - TKfit.h, [121](#)
- TKretMax_d
 - T2toolkit.h, [91](#)
- TKretMax_f
 - T2toolkit.h, [91](#)
- TKretMin_d
 - T2toolkit.h, [91](#)
- TKretMin_f
 - T2toolkit.h, [91](#)
- TKretMin_i
 - T2toolkit.h, [91](#)
- TKrobustConstrainedLeastSquares
 - TKfit.h, [121](#)
- TKrobustLeastSquares
 - TKfit.h, [121](#)
- TKsetSeed
 - T2toolkit.h, [91](#)
- TKsign_d
 - T2toolkit.h, [91](#)
- TKsingularValueDecomposition_lsq
 - TKsvd.h, [130](#)
- TKsort_2f
 - T2toolkit.h, [91](#)
- TKsort_3d
 - T2toolkit.h, [91](#)
- TKsort_d
 - T2toolkit.h, [91](#)
- TKsort_f
 - T2toolkit.h, [91](#)
- TKsortit
 - TKspectrum.h, [130](#)
- TKspectrum
 - TKspectrum.h, [130](#)
- TKspectrum.h, [128](#)
 - ABS, [129](#)
 - calcSpectra, [129](#)
 - calcSpectraErr, [129](#)
 - complexVal, [129](#)
 - fit4, [129](#)
 - getprtj, [129](#)
 - getweights, [129](#)
 - indexx8, [129](#)
 - MAX, [129](#)
 - MIN, [129](#)

- mat20, [129](#)
- sineFunc, [129](#)
- TK_dft, [129](#)
- TK_fft, [129](#)
- TK_fitSine, [129](#)
- TK_fitSinusoids, [129](#)
- TK_weightLS, [129](#)
- TKaveragePts, [129](#)
- TKboxcar, [129](#)
- TKcmonot, [129](#)
- TKfirstDifference, [129](#)
- TKhann, [129](#)
- TKinterpolateSplineSmoothFixedXPts, [129](#)
- TKlomb_d, [129](#)
- TKsortit, [130](#)
- TKspectrum, [130](#)
- TKspline_interpolate, [130](#)
- verbose_calc_spectra, [130](#)
- TKspline_interpolate
 - TKspectrum.h, [130](#)
- TKsvd.h, [130](#)
 - TKbacksubstitution_svd, [130](#)
 - TKbidiagonal, [130](#)
 - TKpythag, [130](#)
 - TKsingularValueDecomposition_Isq, [130](#)
- TKvariance_d
 - T2toolkit.h, [91](#)
- TKzeromean_d
 - T2toolkit.h, [91](#)
- TNBandDMAmp
 - pulsar, [64](#)
- TNBandDMC
 - pulsar, [64](#)
- TNBandDMGam
 - pulsar, [64](#)
- TNBandNoiseAmp
 - pulsar, [64](#)
- TNBandNoiseC
 - pulsar, [64](#)
- TNBandNoiseGam
 - pulsar, [64](#)
- TNBandNoiseHF
 - pulsar, [64](#)
- TNBandNoiseLF
 - pulsar, [64](#)
- TNDMAmp
 - pulsar, [64](#)
- TNDMC
 - pulsar, [64](#)
- TNDMCoeffs
 - pulsar, [64](#)
- TNDMErr
 - observation, [44](#)
- TNDMEvAmp
 - pulsar, [64](#)
- TNDMEvGam
 - pulsar, [64](#)
- TNDMEvLength
 - pulsar, [64](#)
- TNDMEvLin
 - pulsar, [64](#)
- TNDMEvOff
 - pulsar, [64](#)
- TNDMEvQuad
 - pulsar, [64](#)
- TNDMEvStart
 - pulsar, [64](#)
- TNDMGam
 - pulsar, [64](#)
- TNDMSignal
 - observation, [45](#)
- TNECORRFlagID
 - pulsar, [64](#)
- TNECORRFlagVal
 - pulsar, [64](#)
- TNECORRVal
 - pulsar, [64](#)
- TNEFFlagID
 - pulsar, [64](#)
- TNEFFlagVal
 - pulsar, [65](#)
- TNEFVal
 - pulsar, [65](#)
- TNEQFlagID
 - pulsar, [65](#)
- TNEQFlagVal
 - pulsar, [65](#)
- TNEQVal
 - pulsar, [65](#)
- TNGlobalIEF
 - pulsar, [65](#)
- TNGlobalIEQ
 - pulsar, [65](#)
- TNGroupErr
 - observation, [45](#)
- TNGroupNoiseAmp
 - pulsar, [65](#)
- TNGroupNoiseC
 - pulsar, [65](#)
- TNGroupNoiseFlagID
 - pulsar, [65](#)
- TNGroupNoiseFlagVal
 - pulsar, [65](#)
- TNGroupNoiseGam
 - pulsar, [65](#)
- TNGroupSignal
 - observation, [45](#)
- TNRedAmp
 - pulsar, [65](#)
- TNRedC
 - pulsar, [65](#)
- TNRedCoeffs
 - pulsar, [65](#)
- TNRedCorner
 - pulsar, [65](#)
- TNRedErr

- observation, [45](#)
- TNRedFlow
 - pulsar, [65](#)
- TNRedGam
 - pulsar, [65](#)
- TNRedSignal
 - observation, [45](#)
- TNSQFlagID
 - pulsar, [65](#)
- TNSQFlagVal
 - pulsar, [65](#)
- TNSQVal
 - pulsar, [65](#)
- TNShapeletEvFScale
 - pulsar, [65](#)
- TNShapeletEvN
 - pulsar, [65](#)
- TNShapeletEvPos
 - pulsar, [65](#)
- TNShapeletEvWidth
 - pulsar, [65](#)
- TNsubtractDM
 - pulsar, [65](#)
- TNsubtractRed
 - pulsar, [65](#)
- tOffset
 - pulsar, [66](#)
- tOffset_f1
 - pulsar, [66](#)
- tOffset_f2
 - pulsar, [66](#)
- tOffset_t1
 - pulsar, [66](#)
- tOffset_t2
 - pulsar, [66](#)
- tOffsetFlags
 - pulsar, [66](#)
- tOffsetSite
 - pulsar, [66](#)
- TSUN
 - tempo2.h, [104](#)
- TabulatedFunction, [69](#)
 - fileName, [69](#)
 - header_line, [69](#)
 - samples, [70](#)
- TabulatedFunction_getEndX
 - tabulatedfunction.h, [91](#)
- TabulatedFunction_getStartX
 - tabulatedfunction.h, [91](#)
- TabulatedFunction_getValue
 - tabulatedfunction.h, [92](#)
- TabulatedFunction_load
 - tabulatedfunction.h, [92](#)
- TabulatedFunctionSample, [70](#)
 - x, [70](#)
 - y, [70](#)
- tabulatedfunction.h, [91](#)
 - TabulatedFunction_getEndX, [91](#)
 - TabulatedFunction_getStartX, [91](#)
 - TabulatedFunction_getValue, [92](#)
 - TabulatedFunction_load, [92](#)
- tai2tt
 - tempo2.h, [112](#)
- tai2ut1
 - tempo2.h, [112](#)
- tcheck
 - TKlog.h, [123](#)
- tdis1
 - observation, [44](#)
- tdis2
 - observation, [44](#)
- telDX_e
 - pulsar, [63](#)
- telDX_t
 - pulsar, [63](#)
- telDX_v
 - pulsar, [63](#)
- telDX_vel
 - pulsar, [63](#)
- telDX_vel_e
 - pulsar, [63](#)
- telDY_e
 - pulsar, [63](#)
- telDY_t
 - pulsar, [63](#)
- telDY_v
 - pulsar, [63](#)
- telDY_vel
 - pulsar, [63](#)
- telDY_vel_e
 - pulsar, [63](#)
- telDZ_e
 - pulsar, [63](#)
- telDZ_t
 - pulsar, [63](#)
- telDZ_v
 - pulsar, [63](#)
- telDZ_vel
 - pulsar, [63](#)
- telDZ_vel_e
 - pulsar, [64](#)
- telID
 - observation, [44](#)
- tempo1
 - pulsar, [64](#)
- tempo2.h, [92](#)
 - AU_DIST, [98](#)
 - AULTSC, [98](#)
 - allocateMemory, [109](#)
 - autoConstraints, [109](#)
 - BIG_G, [98](#)
 - BTJmodel, [109](#)
 - BTXmodel, [109](#)
 - BTmodel, [109](#)
 - bootstrap, [109](#)
 - CVSdisplayVersion, [109](#)

calcRMS, 109
calculate_bclt, 109
compute_tropospheric_delays, 109
constraint, 105
constraint_LAST, 106
constraint_dmmodel_cw_0, 105
constraint_dmmodel_cw_1, 105
constraint_dmmodel_cw_2, 105
constraint_dmmodel_cw_3, 105
constraint_dmmodel_cw_px, 106
constraint_dmmodel_cw_year_cos, 106
constraint_dmmodel_cw_year_cos2, 106
constraint_dmmodel_cw_year_sin, 106
constraint_dmmodel_cw_year_sin2, 106
constraint_dmmodel_cw_year_xcos, 106
constraint_dmmodel_cw_year_xsin, 106
constraint_dmmodel_dm1, 105
constraint_dmmodel_mean, 105
constraint_ifunc_0, 105
constraint_ifunc_1, 105
constraint_ifunc_2, 105
constraint_ifunc_year_cos, 106
constraint_ifunc_year_cos2, 106
constraint_ifunc_year_sin, 106
constraint_ifunc_year_sin2, 106
constraint_ifunc_year_xcos, 106
constraint_ifunc_year_xsin, 106
constraint_label, 104
constraint_qifunc_c_year_cos, 106
constraint_qifunc_c_year_cos2, 106
constraint_qifunc_c_year_sin, 106
constraint_qifunc_c_year_sin2, 106
constraint_qifunc_c_year_xcos, 106
constraint_qifunc_c_year_xsin, 106
constraint_qifunc_p_year_cos, 106
constraint_qifunc_p_year_cos2, 106
constraint_qifunc_p_year_sin, 106
constraint_qifunc_p_year_sin2, 106
constraint_qifunc_p_year_xcos, 106
constraint_qifunc_p_year_xsin, 106
constraint_quad_ifunc_c_0, 105
constraint_quad_ifunc_c_1, 106
constraint_quad_ifunc_c_2, 106
constraint_quad_ifunc_p_0, 105
constraint_quad_ifunc_p_1, 105
constraint_quad_ifunc_p_2, 105
constraint_tel_dx_0, 105
constraint_tel_dx_1, 105
constraint_tel_dx_2, 105
constraint_tel_dy_0, 105
constraint_tel_dy_1, 105
constraint_tel_dy_2, 105
constraint_tel_dz_0, 105
constraint_tel_dz_1, 105
constraint_tel_dz_2, 105
constraintDerivFunc, 104
copyPSR, 109
copyParam, 109
covarFuncFile, 114
DDGRmodel, 110
DDHmodel, 110
DDKmodel, 110
DDSmodel, 110
DDmodel, 110
DM_CONST, 98
DM_CONST_SI, 98
dcmFile, 114
defineClockCorrectionSequence, 110
destroyMemory, 110
destroyOne, 110
displayCVSversion, 114
displayMsg, 110
displayParameters, 110
dm_delays, 110
dms_turn, 110
doFit, 110
doFitAll, 110
doFitDCM, 110
doFitGlobal, 110
dotproduct, 110
ECLIPTIC_OBLIQUITY, 114
ECLIPTIC_OBLIQUITY_VAL, 98
ELL1Hmodel, 110
ELL1model, 110
equ2ecl, 110
FB90_TIMEEPH, 98
FITfuncs, 110
FitInfo, 104
forceGlobalFit, 114
formBats, 110
formBatsAll, 110
formResiduals, 110
fortran_mod, 110
fortran_nint, 110
fortran_nlong, 110
GM, 98
GM_C3, 99
GMJ_C3, 99
GMN_C3, 99
GMS_C3, 99
GMU_C3, 99
GMV_C3, 99
get_EOP, 110
get_OneobsCoord, 111
get_obsCoord, 111
get_obsCoord_IAU2000B, 111
getCholeskyMatrix, 111
getClockCorrections, 111
getCorrection, 111
getCorrectionTT, 111
getInputs, 111
getObservatory, 111
getParamDeriv, 111
getParameterValue, 111
HAVE_GWSIM_H, 99
hms_turn, 111

IF99_TIMEEPH, 99
 IFTEPH_FILE, 99
 id_residual, 111
 initialise, 111
 initialiseOne, 111
 JVmodel, 111
 LEAPSECOND_FILE, 99
 label, 106
 logicFlag, 111
 lookup_observatory_alias, 111
 MASYR2RADS, 99
 MAX_BPJ_JUMPS, 99
 MAX_CLK_CORR, 99
 MAX_CLKCORR, 100
 MAX_COEFF, 100
 MAX_COMPANIONS, 100
 MAX_DM_DERIVATIVES, 100
 MAX_DMX, 100
 MAX_FILELEN, 100
 MAX_FIT, 100
 MAX_FLAG_LEN, 100
 MAX_FLAGS, 100
 MAX_FREQ_DERIVATIVES, 100
 MAX_IFUNC, 100
 MAX_JUMPS, 100
 MAX_LEAPSEC, 101
 MAX_MSG, 101
 MAX_OBSN, 114
 MAX_OBSN_VAL, 101
 MAX_PARAMS, 101
 MAX_PSR, 114
 MAX_PSR_VAL, 101
 MAX_QUAD, 101
 MAX_SITE, 101
 MAX_STOREPRECISION, 101
 MAX_STRLEN, 101
 MAX_T2EFAC, 101
 MAX_T2EQUAD, 101
 MAX_TEL_CLK_OFFS, 101
 MAX_TEL_DX, 102
 MAX_TEL_DY, 102
 MAX_TEL_DZ, 102
 MAX_TNBN, 102
 MAX_TNDMEv, 102
 MAX_TNECORR, 102
 MAX_TNEF, 102
 MAX_TNEQ, 102
 MAX_TNGN, 102
 MAX_TNSQ, 102
 MAX_TOFFSET, 102
 MAX_WHITE, 102
 MSSmodel, 111
 NE_SW_DEFAULT, 102
 NEWFIT, 114
 OBLQ, 103
 OBSSYS_FILE, 103
 observation, 104
 PCM, 103
 param_JUMP, 109
 param_LAST, 109
 param_ZERO, 109
 param_a0, 108
 param_a1, 107
 param_a1dot, 107
 param_a2dot, 107
 param_afac, 108
 param_b0, 108
 param_bp, 107
 param_bpja1, 108
 param_bpjec, 108
 param_bpjep, 108
 param_bpjom, 108
 param_bpjpb, 108
 param_bpjph, 108
 param_bpp, 107
 param_brake, 109
 param_cgw, 108
 param_clk_offs, 108
 param_daop, 108
 param_decj, 106
 param_df1, 109
 param_dm, 107
 param_dm_cos1yr, 109
 param_dm_sin1yr, 109
 param_dmassplanet, 108
 param_dmepoch, 106
 param_dmmodel, 108
 param_dmx, 108
 param_dmrx1, 108
 param_dmrx2, 108
 param_dphaseplanet, 108
 param_dr, 107
 param_dshk, 108
 param_dth, 108
 param_dtheta, 108
 param_e2dot, 107
 param_ecc, 107
 param_edot, 107
 param_ephver, 108
 param_eps1, 107
 param_eps1dot, 108
 param_eps2, 107
 param_eps2dot, 108
 param_f, 106
 param_fb, 107
 param_fd, 107
 param_fddc, 107
 param_fddi, 107
 param_finish, 107
 param_gamma, 107
 param_glep, 107
 param_glf0, 107
 param_glf0d, 107
 param_glf1, 107
 param_glf2, 107
 param_glph, 107

param_gltid, 107
 param_gwb_amp, 109
 param_gwecc, 109
 param_gwm_amp, 109
 param_gwsingle, 108
 param_h3, 108
 param_h4, 108
 param_ifunc, 108
 param_iperharm, 108
 param_kin, 108
 param_kom, 108
 param_label, 104
 param_m2, 107
 param_mtot, 107
 param_nharm, 108
 param_om, 107
 param_om2dot, 107
 param_omdot, 107
 param_orbpx, 107
 param_pb, 107
 param_pbdot, 107
 param_pepoch, 106
 param_pmdec, 107
 param_pmra, 107
 param_pmr, 107
 param_posepoch, 106
 param_px, 107
 param_quad_ifunc_c, 109
 param_quad_ifunc_p, 109
 param_quad_om, 108
 param_raj, 106
 param_shapmax, 108
 param_sini, 107
 param_start, 107
 param_stateSwitchT, 109
 param_stig, 108
 param_t0, 107
 param_tasc, 107
 param_tel_dx, 109
 param_tel_dy, 109
 param_tel_dz, 109
 param_tel_vx, 109
 param_tel_vy, 109
 param_tel_vz, 109
 param_tel_x0, 109
 param_tel_y0, 109
 param_tel_z0, 109
 param_telEpoch, 109
 param_telx, 108
 param_tely, 108
 param_telz, 108
 param_track, 107
 param_tres, 108
 param_tspan, 108
 param_tzrfreq, 107
 param_tzrmjd, 107
 param_wave_dm, 108
 param_wave_om, 108
 param_waveepoch, 108
 param_waveepoch_dm, 108
 param_xomdot, 108
 param_xpbdot, 107
 paramDerivFunc, 104
 paramUpdateFunc, 105
 parameter, 104
 polyco, 111
 preProcess, 111
 preProcessSimple, 111
 preProcessSimple1, 111
 preProcessSimple2, 111
 preProcessSimple3, 112
 processFlag, 112
 processSimultaneous, 112
 pulsar, 105
 readEphemeris, 112
 readEphemeris_calceph, 112
 readJBO_bat, 112
 readObsFile, 112
 readOneEphemeris, 112
 readParfile, 112
 readParfileGlobal, 112
 readSimpleParfile, 112
 readTimfile, 112
 recordPrecision, 112
 SECDAY, 103
 SECDAYI, 103
 SI_UNITS, 103
 SOLAR_MASS, 103
 SOLAR_RADIUS, 103
 SPEED_LIGHT, 103
 secularMotion, 112
 setPlugPath, 112
 setStart, 112
 setupParameterFileDefaults, 112
 shapiro_delay, 112
 simplePlot, 112
 solarWindModel, 112
 sortToAs, 112
 storePrecision, 105
 T2_PTAmode, 112
 T2C_IAU2000B, 103
 T2C_TEMPO, 103
 T2model, 112
 TDB_UNITS, 103
 TDBTDT_FILE, 103
 TEMPO2_ENVIRON, 114
 TEMPO2_ERROR, 114
 TEMPO2_h_HASH, 104
 TEMPO2_h_MAJOR_VER, 104
 TEMPO2_h_MINOR_VER, 104
 TEMPO2_h_VER, 104
 TSUN, 104
 tai2tt, 112
 tai2ut1, 112
 tempo2_plug_path, 114
 tempo2_plug_path_len, 114

- tempo2MachineType, 115
- textOutput, 112
- toa2utc, 113
- transform_units, 113
- tt2tb, 113
- tt2tb_calceph, 113
- turn_deg, 113
- turn_dms, 113
- turn_hms, 113
- UT1_FILE, 104
- updateBT, 113
- updateBTJ, 113
- updateBTX, 113
- updateBatsAll, 113
- updateDD, 113
- updateDDGR, 113
- updateDDH, 113
- updateDDK, 113
- updateDDS, 113
- updateELL1, 113
- updateELL1H, 113
- updateJV, 113
- updateMSS, 113
- updateParameters, 113
- updateT2, 113
- updateT2_PTA, 113
- useSelectFile, 113
- utc2tai, 113
- vectorPulsar, 113
- vectorscale, 113
- vectorsum, 113
- veryFast, 115
- writeTim, 114
- zoom_graphics, 114
- tempo2_plug_path
 - tempo2.h, 114
- tempo2_plug_path_len
 - tempo2.h, 114
- tempo2MachineType
 - tempo2.h, 115
- tempo2Util.h, 119
 - dms_turn, 119
 - hms_turn, 119
 - turn_deg, 119
- tempo2pred.h, 115
 - Cheby, 116
 - ChebyModelSet_OutOfRange, 117
 - NonePredType, 116
 - T1, 116
 - T2Predictor_Copy, 116
 - T2Predictor_Destroy, 116
 - T2Predictor_FRead, 116
 - T2Predictor_FWrite, 116
 - T2Predictor_GetEndFreq, 116
 - T2Predictor_GetEndMJD, 116
 - T2Predictor_GetFrequency, 116
 - T2Predictor_GetPSRName, 116
 - T2Predictor_GetPhase, 116
 - T2Predictor_GetPlan, 116
 - T2Predictor_GetPlan_Ext, 116
 - T2Predictor_GetSiteName, 116
 - T2Predictor_GetStartFreq, 116
 - T2Predictor_GetStartMJD, 116
 - T2Predictor_Init, 116
 - T2Predictor_Insert, 116
 - T2Predictor_Keep, 116
 - T2Predictor_Kind, 116
 - T2Predictor_Read, 116
 - T2Predictor_Write, 117
 - T2PredictorKind, 116
- tempo2pred_int.h, 117
 - Cheby2D_Construct, 118
 - Cheby2D_Construct_x_Derivative, 118
 - Cheby2D_Test, 118
 - ChebyModel_Construct, 118
 - ChebyModel_Copy, 118
 - ChebyModel_Destroy, 118
 - ChebyModel_GetFrequency, 118
 - ChebyModel_GetPhase, 118
 - ChebyModel_Init, 118
 - ChebyModel_Read, 118
 - ChebyModel_Test, 118
 - ChebyModel_Write, 118
 - ChebyModelSet_Construct, 118
 - ChebyModelSet_Destroy, 118
 - ChebyModelSet_GetFrequency, 118
 - ChebyModelSet_GetNearest, 118
 - ChebyModelSet_GetPhase, 118
 - ChebyModelSet_Init, 118
 - ChebyModelSet_Insert, 118
 - ChebyModelSet_Keep, 118
 - ChebyModelSet_Read, 118
 - ChebyModelSet_Test, 118
 - ChebyModelSet_Write, 118
 - T1Polyco_GetFrequency, 118
 - T1Polyco_GetPhase, 118
 - T1Polyco_Read, 119
 - T1Polyco_Write, 119
 - T1PolycoSet_Destroy, 119
 - T1PolycoSet_GetFrequency, 119
 - T1PolycoSet_GetNearest, 119
 - T1PolycoSet_GetPhase, 119
 - T1PolycoSet_Read, 119
 - T1PolycoSet_Write, 119
- tensor_alpha
 - gwwgenSpec, 35
- tensor_amp
 - gwwgenSpec, 35
- textOutput
 - tempo2.h, 112
- theta_bin
 - gwSrc, 36
 - gwgeneralSrc, 34
- theta_g
 - gwSrc, 36
 - gwgeneralSrc, 34

- timeEphemeris
 - pulsar, 64
- timer_clk
 - TKlog.h, 123
- ToAextraCovar
 - pulsar, 65
- toa2utc
 - tempo2.h, 113
- toaDMErr
 - observation, 45
- toaErr
 - observation, 45
- torb
 - observation, 45
- transform_units
 - tempo2.h, 113
- troposphericDelay
 - observation, 45
- tt2tb
 - tempo2.h, 113
- tt2tb_calceph
 - tempo2.h, 113
- turn_deg
 - tempo2.h, 113
 - tempo2Util.h, 119
- turn_dms
 - tempo2.h, 113
- turn_hms
 - tempo2.h, 113
- twot
 - interpolation_info, 37
- tzrsite
 - pulsar, 66
- USE_BUILTIN_LONGDOUBLE
 - TKlongdouble.float128.h, 124
 - TKlongdouble.h, 125
 - TKlongdouble.ld.h, 126
- UT1_FILE
 - tempo2.h, 104
- units
 - pulsar, 66
- updateBT
 - tempo2.h, 113
- updateBTJ
 - tempo2.h, 113
- updateBTX
 - tempo2.h, 113
- updateBatsAll
 - tempo2.h, 113
- updateDD
 - tempo2.h, 113
- updateDDGR
 - tempo2.h, 113
- updateDDH
 - tempo2.h, 113
- updateDDK
 - tempo2.h, 113
- updateDDS
 - tempo2.h, 113
- updateELL1
 - tempo2.h, 113
- updateELL1H
 - tempo2.h, 113
- updateFunctions
 - FitInfo, 33
- updateJV
 - tempo2.h, 113
- updateMSS
 - tempo2.h, 113
- updateParameters
 - tempo2.h, 113
- updateT2
 - tempo2.h, 113
- updateT2_PTA
 - tempo2.h, 113
- uranus_earth
 - observation, 45
- useCalceph
 - pulsar, 66
- useSelectFile
 - tempo2.h, 113
- useT2accel
 - T2accel.h, 85
- useTNOrth
 - pulsar, 66
- utc2tai
 - tempo2.h, 113
- utc_string
 - T1Polyco, 68
- VERSION
 - config.h, 74
- val
 - parameter, 48
- vectorPulsar
 - tempo2.h, 113
- vectorscale
 - tempo2.h, 113
- vectorsum
 - tempo2.h, 113
- vel_coeff
 - interpolation_info, 37
- velPulsar
 - pulsar, 66
- venus_earth
 - observation, 45
- verbose_calc_spectra
 - TKspectrum.h, 130
- veryFast
 - tempo2.h, 115
- vl_alpha
 - gwgenSpec, 35
- vl_amp
 - gwgenSpec, 35
- WARNCOLOR
 - TKlog.h, 123

WHEREARG
 TKlog.h, [123](#)
WHEREERR
 TKlog.h, [123](#)
WHERESTR
 TKlog.h, [123](#)
WHERECHK
 TKlog.h, [123](#)
WHEREWARN
 TKlog.h, [123](#)
wave_cos
 pulsar, [66](#)
wave_cos_dm
 pulsar, [66](#)
wave_cos_dm_err
 pulsar, [66](#)
wave_cos_err
 pulsar, [66](#)
wave_sine
 pulsar, [66](#)
wave_sine_dm
 pulsar, [66](#)
wave_sine_dm_err
 pulsar, [66](#)
wave_sine_err
 pulsar, [66](#)
waveScale
 pulsar, [66](#)
whiteNoiseModelFile
 pulsar, [66](#)
writeResiduals
 TKlog.h, [123](#)
writeTim
 tempo2.h, [114](#)

x
 observatory, [46](#)
 TabulatedFunctionSample, [70](#)

y
 observatory, [46](#)
 TabulatedFunctionSample, [70](#)

z
 observatory, [46](#)
zenith
 observation, [45](#)
zoom_graphics
 tempo2.h, [114](#)