

Tempo2

Generated by Doxygen 1.8.14



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
<b>2</b>	<b>User Guide</b>	<b>3</b>
2.1	Tempo2 User Manual . . . . .	3
2.1.1	About tempo2 . . . . .	3
2.1.2	Terminology and basic usage . . . . .	3
<b>3</b>	<b>Core Developers</b>	<b>5</b>
<b>4</b>	<b>Developer Guide</b>	<b>7</b>
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 . . . . .	8
4.1.4	Coding style . . . . .	9
<b>5</b>	<b>Directory structure</b>	<b>11</b>
<b>6</b>	<b>Plugin Documentation</b>	<b>13</b>
6.1	Tempo2 Plugins . . . . .	13
<b>7</b>	<b>Git INSTALLATION README</b>	<b>15</b>
<b>8</b>	<b>Todo List</b>	<b>19</b>
<b>9</b>	<b>Module Index</b>	<b>21</b>
9.1	Modules . . . . .	21

<b>10 Class Index</b>	<b>23</b>
10.1 Class List . . . . .	23
<b>11 File Index</b>	<b>25</b>
11.1 File List . . . . .	25
<b>12 Module Documentation</b>	<b>27</b>
12.1 libt2toolkit API . . . . .	27
12.1.1 Detailed Description . . . . .	27
12.2 libtempo2 External API . . . . .	28
12.2.1 Detailed Description . . . . .	28
<b>13 Class Documentation</b>	<b>29</b>
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 . . . . .	30
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 . . . . .	31
13.2.1.6 mjd_end . . . . .	31
13.2.1.7 mjd_start . . . . .	31
13.2.1.8 psrname . . . . .	31
13.2.1.9 sitename . . . . .	31
13.3 ChebyModelSet Struct Reference . . . . .	31
13.3.1 Member Data Documentation . . . . .	32
13.3.1.1 nsegments . . . . .	32

13.3.1.2	segments	32
13.4	clock_correction Struct Reference	32
13.4.1	Detailed Description	32
13.4.2	Member Data Documentation	32
13.4.2.1	correction	32
13.4.2.2	corrects_to	33
13.5	complexVal Struct Reference	33
13.5.1	Member Data Documentation	33
13.5.1.1	imag	33
13.5.1.2	real	33
13.6	constraint_param_info Struct Reference	33
13.6.1	Member Data Documentation	34
13.6.1.1	err	34
13.6.1.2	param	34
13.6.1.3	param_k	34
13.6.1.4	val	34
13.7	DynamicArray Struct Reference	34
13.7.1	Member Data Documentation	34
13.7.1.1	data	35
13.7.1.2	elem_size	35
13.7.1.3	nallocated	35
13.7.1.4	nelem	35
13.8	FitInfo Struct Reference	35
13.8.1	Detailed Description	36
13.8.2	Member Data Documentation	36
13.8.2.1	constraintCounters	36
13.8.2.2	constraintDerivs	36
13.8.2.3	constraintIndex	36
13.8.2.4	constraintSpecial	36
13.8.2.5	constraintValue	36

13.8.2.6	nConstraints	36
13.8.2.7	nParams	37
13.8.2.8	output	37
13.8.2.9	paramCounters	37
13.8.2.10	paramDerivs	37
13.8.2.11	paramIndex	37
13.8.2.12	updateFunctions	37
13.9	FitOutput Struct Reference	37
13.9.1	Member Data Documentation	38
13.9.1.1	errorEstimates	38
13.9.1.2	indexCounter	38
13.9.1.3	indexParam	38
13.9.1.4	indexPsr	38
13.9.1.5	parameterEstimates	38
13.9.1.6	totalNfit	39
13.10	gwgeneralSrc Struct Reference	39
13.10.1	Member Data Documentation	39
13.10.1.1	across_g	39
13.10.1.2	across_im_g	40
13.10.1.3	aplus_g	40
13.10.1.4	aplus_im_g	40
13.10.1.5	asl_g	40
13.10.1.6	asl_im_g	40
13.10.1.7	ast_g	40
13.10.1.8	ast_im_g	40
13.10.1.9	avx_g	40
13.10.1.10	avx_im_g	41
13.10.1.11	avy_g	41
13.10.1.12	avy_im_g	41
13.10.1.13	dist_bin	41

13.10.1.14h	41
13.10.1.15h_im	41
13.10.1.16nc_bin	41
13.10.1.17kg	41
13.10.1.18omega_g	42
13.10.1.19phase_g	42
13.10.1.20phi_bin	42
13.10.1.21phi_g	42
13.10.1.22phi_polar_g	42
13.10.1.23theta_bin	42
13.10.1.24theta_g	42
13.11gwgenSpec Struct Reference	43
13.11.1 Member Data Documentation	43
13.11.1.1 sl_alpha	43
13.11.1.2 sl_amp	43
13.11.1.3 st_alpha	43
13.11.1.4 st_amp	43
13.11.1.5 tensor_alpha	44
13.11.1.6 tensor_amp	44
13.11.1.7 vl_alpha	44
13.11.1.8 vl_amp	44
13.12gwSrc Struct Reference	44
13.12.1 Member Data Documentation	45
13.12.1.1 across_g	45
13.12.1.2 across_im_g	45
13.12.1.3 aplus_g	45
13.12.1.4 aplus_im_g	45
13.12.1.5 dist_bin	45
13.12.1.6 h	45
13.12.1.7 h_im	45

13.12.1.8 inc_bin . . . . .	46
13.12.1.9 kg . . . . .	46
13.12.1.10 omega_g . . . . .	46
13.12.1.11 phase_g . . . . .	46
13.12.1.12 phi_bin . . . . .	46
13.12.1.13 phi_g . . . . .	46
13.12.1.14 phi_polar_g . . . . .	46
13.12.1.15 theta_bin . . . . .	46
13.12.1.16 theta_g . . . . .	47
13.13 interpolation_info Struct Reference . . . . .	47
13.13.1 Member Data Documentation . . . . .	47
13.13.1.1 n_posn_avail . . . . .	47
13.13.1.2 n_vel_avail . . . . .	47
13.13.1.3 posn_coeff . . . . .	47
13.13.1.4 twot . . . . .	48
13.13.1.5 vel_coeff . . . . .	48
13.14 jpl_eph_data Struct Reference . . . . .	48
13.14.1 Member Data Documentation . . . . .	48
13.14.1.1 au . . . . .	49
13.14.1.2 cache . . . . .	49
13.14.1.3 curr_cache_loc . . . . .	49
13.14.1.4 emrat . . . . .	49
13.14.1.5 ephem_end . . . . .	49
13.14.1.6 ephem_start . . . . .	49
13.14.1.7 ephem_step . . . . .	49
13.14.1.8 ephemeris_version . . . . .	49
13.14.1.9 ifile . . . . .	50
13.14.1.10 info . . . . .	50
13.14.1.11 ipt . . . . .	50
13.14.1.12 kernel_size . . . . .	50



13.14.1.13	ncoeff	50
13.14.1.14	ncon	50
13.14.1.15	pvsun	50
13.14.1.16	pvsun_t	50
13.14.1.17	recsize	51
13.14.1.18	swap_bytes	51
13.15	observation Struct Reference	51
13.15.1	Detailed Description	53
13.15.2	Member Data Documentation	53
13.15.2.1	addedNoise	53
13.15.2.2	averagebat	53
13.15.2.3	averagedmbat	53
13.15.2.4	averagedmerr	53
13.15.2.5	averagedmres	53
13.15.2.6	averageerr	54
13.15.2.7	averageres	54
13.15.2.8	bat	54
13.15.2.9	batCorr	54
13.15.2.10	bbat	54
13.15.2.11	bline	54
13.15.2.12	chisq	54
13.15.2.13	clockCorr	54
13.15.2.14	correctionsTT	55
13.15.2.15	correctionTT_calcEph	55
13.15.2.16	correctionTT_TB	55
13.15.2.17	correctionTT_Teph	55
13.15.2.18	correctionUT1	55
13.15.2.19	delayCorr	55
13.15.2.20	deleted	55
13.15.2.21	earth_ssb	55

13.15.2.22earthMoonBary_earth	56
13.15.2.23earthMoonBary_ssb	56
13.15.2.24efac	56
13.15.2.25einsteinRate	56
13.15.2.26equad	56
13.15.2.27lagID	56
13.15.2.28lagVal	56
13.15.2.29name	56
13.15.2.30freq	57
13.15.2.31freqSSB	57
13.15.2.32jump	57
13.15.2.33jupiter_earth	57
13.15.2.34nclk_correction	57
13.15.2.35neptune_earth	57
13.15.2.36nFlags	57
13.15.2.37nphase	57
13.15.2.38nutations	58
13.15.2.39observatory_earth	58
13.15.2.40obsNjump	58
13.15.2.41origErr	58
13.15.2.42brigsat	58
13.15.2.43pet	58
13.15.2.44phase	58
13.15.2.45phaseOffset	58
13.15.2.46planet_ssb	59
13.15.2.47planet_ssb_derv	59
13.15.2.48planet_ssb_tmr	59
13.15.2.49pnoise	59
13.15.2.50prefitResidual	59
13.15.2.51psrPos	59

13.15.2.52	pulseN	59
13.15.2.53	residual	59
13.15.2.54	residualtn	60
13.15.2.55	roemer	60
13.15.2.56	sat	60
13.15.2.57	sat_day	60
13.15.2.58	sat_sec	60
13.15.2.59	saturn_earth	60
13.15.2.60	shapiroDelayJupiter	60
13.15.2.61	shapiroDelayNeptune	60
13.15.2.62	shapiroDelaySaturn	61
13.15.2.63	shapiroDelaySun	61
13.15.2.64	shapiroDelayUranus	61
13.15.2.65	shapiroDelayVenus	61
13.15.2.66	shklovskii	61
13.15.2.67	siteVel	61
13.15.2.68	snr	61
13.15.2.69	sun_earth	61
13.15.2.70	sun_ssb	62
13.15.2.71	tdis1	62
13.15.2.72	tdis2	62
13.15.2.73	telID	62
13.15.2.74	TNDMErr	62
13.15.2.75	TNDMSignal	62
13.15.2.76	TNGroupErr	62
13.15.2.77	TNGroupSignal	62
13.15.2.78	TNRedErr	63
13.15.2.79	TNRedSignal	63
13.15.2.80	toaDMErr	63
13.15.2.81	toaErr	63

13.15.2.82obs	63
13.15.2.83orb	63
13.15.2.84troposphericDelay	63
13.15.2.85uranus_earth	63
13.15.2.86venus_earth	64
13.15.2.87zenith	64
13.16observatory Struct Reference	64
13.16.1 Member Data Documentation	64
13.16.1.1 clock_name	64
13.16.1.2 code	64
13.16.1.3 height_grs80	65
13.16.1.4 latitude_grs80	65
13.16.1.5 longitude_grs80	65
13.16.1.6 name	65
13.16.1.7 x	65
13.16.1.8 y	65
13.16.1.9 z	65
13.17parameter Struct Reference	66
13.17.1 Detailed Description	66
13.17.2 Member Data Documentation	66
13.17.2.1 aSize	66
13.17.2.2 err	66
13.17.2.3 fitFlag	67
13.17.2.4 label	67
13.17.2.5 linkFrom	67
13.17.2.6 linkTo	67
13.17.2.7 nLinkFrom	67
13.17.2.8 nLinkTo	67
13.17.2.9 paramSet	67
13.17.2.10prefit	67

13.17.2.11	prefitErr	68
13.17.2.12	shortlabel	68
13.17.2.13	val	68
13.18	pulsar Struct Reference	68
13.18.1	Detailed Description	74
13.18.2	Member Data Documentation	74
13.18.2.1	addTNGlobalEQ	74
13.18.2.2	auto_constraints	74
13.18.2.3	AverageDMResiduals	74
13.18.2.4	AverageEpochWidth	74
13.18.2.5	AverageFlag	74
13.18.2.6	AverageResiduals	74
13.18.2.7	binaryModel	75
13.18.2.8	bootStrap	75
13.18.2.9	brace	75
13.18.2.10	calcShapiro	75
13.18.2.11	cgw_angpol	75
13.18.2.12	cgw_cosinc	75
13.18.2.13	cgw_h0	75
13.18.2.14	cgw_mc	76
13.18.2.15	clk_offsE	76
13.18.2.16	clk_offsT	76
13.18.2.17	clk_offsV	76
13.18.2.18	clkOffsN	76
13.18.2.19	clock	76
13.18.2.20	clockFromOverride	76
13.18.2.21	constraint_efactor	77
13.18.2.22	constraint_special	77
13.18.2.23	constraints	77
13.18.2.24	correctTroposphere	77

13.18.2.25covar . . . . .	77
13.18.2.26decjStrPost . . . . .	77
13.18.2.27decjStrPre . . . . .	77
13.18.2.28decsim . . . . .	77
13.18.2.29deleteFileName . . . . .	78
13.18.2.30detUinv . . . . .	78
13.18.2.31dilateFreq . . . . .	78
13.18.2.32dmoffsCM . . . . .	78
13.18.2.33dmoffsCM_error . . . . .	78
13.18.2.34dmoffsCM_mjd . . . . .	78
13.18.2.35dmoffsCM_weight . . . . .	78
13.18.2.36dmoffsCMnum . . . . .	78
13.18.2.37dmoffsDM . . . . .	79
13.18.2.38dmoffsDM_error . . . . .	79
13.18.2.39dmoffsDM_mjd . . . . .	79
13.18.2.40dmoffsDM_weight . . . . .	79
13.18.2.41dmoffsDMnum . . . . .	79
13.18.2.42dmOffset . . . . .	79
13.18.2.43eclCoord . . . . .	79
13.18.2.44eopc04_file . . . . .	79
13.18.2.45ephemeris . . . . .	80
13.18.2.46filterStr . . . . .	80
13.18.2.47fitChisq . . . . .	80
13.18.2.48fitFunc . . . . .	80
13.18.2.49fitinfo . . . . .	80
13.18.2.50fitJump . . . . .	80
13.18.2.51fitMode . . . . .	80
13.18.2.52fitNfree . . . . .	80
13.18.2.53fixedFormat . . . . .	81
13.18.2.54jumpID . . . . .	81

13.18.2.55globalNfit . . . . .	81
13.18.2.56globalNoConstrain . . . . .	81
13.18.2.57gwb_decj . . . . .	81
13.18.2.58gwb_epoch . . . . .	81
13.18.2.59gwb_geom_c . . . . .	81
13.18.2.60gwb_geom_p . . . . .	81
13.18.2.61gwb_raj . . . . .	82
13.18.2.62gwb_width . . . . .	82
13.18.2.63gwcs_decj . . . . .	82
13.18.2.64gwcs_epoch . . . . .	82
13.18.2.65gwcs_geom_c . . . . .	82
13.18.2.66gwcs_geom_p . . . . .	82
13.18.2.67gwcs_raj . . . . .	82
13.18.2.68gwcs_width . . . . .	82
13.18.2.69gwecc_dec . . . . .	83
13.18.2.70gwecc_distance . . . . .	83
13.18.2.71gwecc_e . . . . .	83
13.18.2.72gwecc_epoch . . . . .	83
13.18.2.73gwecc_inc . . . . .	83
13.18.2.74gwecc_m1 . . . . .	83
13.18.2.75gwecc_m2 . . . . .	83
13.18.2.76gwecc_nodes_orientation . . . . .	83
13.18.2.77gwecc_orbital_period . . . . .	84
13.18.2.78gwecc_psrdist . . . . .	84
13.18.2.79gwecc_pulsarTermOn . . . . .	84
13.18.2.80gwecc_ra . . . . .	84
13.18.2.81gwecc_redshift . . . . .	84
13.18.2.82gwecc_theta_0 . . . . .	84
13.18.2.83gwecc_theta_nodes . . . . .	84
13.18.2.84gwm_decj . . . . .	84

13.18.2.85	<code>gwm_dphase</code>	85
13.18.2.86	<code>gwm_epoch</code>	85
13.18.2.87	<code>gwm_phi</code>	85
13.18.2.88	<code>gwm_raj</code>	85
13.18.2.89	<code>gwsrsrc_across_i</code>	85
13.18.2.90	<code>gwsrsrc_across_i_e</code>	85
13.18.2.91	<code>gwsrsrc_across_r</code>	85
13.18.2.92	<code>gwsrsrc_across_r_e</code>	85
13.18.2.93	<code>gwsrsrc_aplus_i</code>	86
13.18.2.94	<code>gwsrsrc_aplus_i_e</code>	86
13.18.2.95	<code>gwsrsrc_aplus_r</code>	86
13.18.2.96	<code>gwsrsrc_aplus_r_e</code>	86
13.18.2.97	<code>gwsrsrc_dec</code>	86
13.18.2.98	<code>gwsrsrc_epoch</code>	86
13.18.2.99	<code>gwsrsrc_psrdist</code>	86
13.18.2.100	<code>gwsrsrc_ra</code>	86
13.18.2.101	<code>unc_weights</code>	87
13.18.2.102	<code>uncE</code>	87
13.18.2.103	<code>uncN</code>	87
13.18.2.104	<code>uncT</code>	87
13.18.2.105	<code>uncV</code>	87
13.18.2.106	<code>pm</code>	87
13.18.2.107	<code>toFormat</code>	87
13.18.2.108	<code>PL_EPHEMERIS</code>	87
13.18.2.109	<code>mpSAT</code>	88
13.18.2.110	<code>mpStr</code>	88
13.18.2.111	<code>mpVal</code>	88
13.18.2.112	<code>mpValErr</code>	88
13.18.2.113	<code>name</code>	88
13.18.2.114	<code>Companion</code>	88



13.18.2.115	constraints	88
13.18.2.116	DMEvents	88
13.18.2.117	dmx	89
13.18.2.118	do_sw	89
13.18.2.119	Fit	89
13.18.2.120	Global	89
13.18.2.121	Hits	89
13.18.2.122	Jumps	89
13.18.2.123	Obs	89
13.18.2.124	Warnings	89
13.18.2.125	Param	90
13.18.2.126	PhaseJump	90
13.18.2.127	Quad	90
13.18.2.128	StorePrecision	90
13.18.2.129	Sx	90
13.18.2.130	T2efac	90
13.18.2.131	T2equad	90
13.18.2.132	TelDX	90
13.18.2.133	TelDY	91
13.18.2.134	TelDZ	91
13.18.2.135	TNBandNoise	91
13.18.2.136	TNECORR	91
13.18.2.137	TNEF	91
13.18.2.138	TNEQ	91
13.18.2.139	TNGroupNoise	91
13.18.2.140	TNShapeletEvents	91
13.18.2.141	TNSQ	92
13.18.2.142	TOffset	92
13.18.2.143	White	92
13.18.2.144	White_dm	92

13.18.2.145	bsn	92
13.18.2.146	offset	92
13.18.2.147	offset_e	92
13.18.2.148	outputTMatrix	92
13.18.2.149	param	93
13.18.2.150	passStr	93
13.18.2.151	phaseJump	93
13.18.2.152	phaseJumpDir	93
13.18.2.153	phaseJumpID	93
13.18.2.154	planetShapiro	93
13.18.2.155	posPulsar	93
13.18.2.156	quad_across_i	93
13.18.2.157	quad_across_i_e	94
13.18.2.158	quad_across_r	94
13.18.2.159	quad_across_r_e	94
13.18.2.160	quad_aplus_i	94
13.18.2.161	quad_aplus_i_e	94
13.18.2.162	quad_aplus_r	94
13.18.2.163	quad_aplus_r_e	94
13.18.2.164	quad_ifunc_c_DEC	94
13.18.2.165	quad_ifunc_c_RA	95
13.18.2.166	quad_ifunc_geom_c	95
13.18.2.167	quad_ifunc_geom_p	95
13.18.2.168	quad_ifunc_p_DEC	95
13.18.2.169	quad_ifunc_p_RA	95
13.18.2.170	quad_ifuncE_c	95
13.18.2.171	quad_ifuncE_p	95
13.18.2.172	quad_ifuncN_c	95
13.18.2.173	quad_ifuncN_p	96
13.18.2.174	quad_ifuncT_c	96

13.18.2.175	quad_ifuncT_p	96
13.18.2.176	quad_ifuncV_c	96
13.18.2.177	quad_ifuncV_p	96
13.18.2.178	quadDEC	96
13.18.2.179	quadEpoch	96
13.18.2.180	quadRA	96
13.18.2.181	rajStrPost	97
13.18.2.182	rajStrPre	97
13.18.2.183	rsim	97
13.18.2.184	rscaleErrChisq	97
13.18.2.185	rsnsPost	97
13.18.2.186	rsnsPre	97
13.18.2.187	rstn	97
13.18.2.188	rust	97
13.18.2.189	setTelVelX	98
13.18.2.190	setTelVelY	98
13.18.2.191	setTelVelZ	98
13.18.2.192	setUnits	98
13.18.2.193	simflag	98
13.18.2.194	sorted	98
13.18.2.195	storePrec	98
13.18.2.196	svm	99
13.18.2.197	tcMethod	99
13.18.2.198	tefacFlagID	99
13.18.2.199	tefacFlagVal	99
13.18.2.200	tefacVal	99
13.18.2.201	tequadFlagID	99
13.18.2.202	tequadFlagVal	99
13.18.2.203	tequadVal	99
13.18.2.204	teglobalEfac	100

13.18.2.205	IDX_e . . . . .	100
13.18.2.206	IDX_t . . . . .	100
13.18.2.207	IDX_v . . . . .	100
13.18.2.208	IDX_vel . . . . .	100
13.18.2.209	IDX_vel_e . . . . .	100
13.18.2.210	IDY_e . . . . .	100
13.18.2.211	IDY_t . . . . .	100
13.18.2.212	IDY_v . . . . .	101
13.18.2.213	IDY_vel . . . . .	101
13.18.2.214	IDY_vel_e . . . . .	101
13.18.2.215	IDZ_e . . . . .	101
13.18.2.216	IDZ_t . . . . .	101
13.18.2.217	IDZ_v . . . . .	101
13.18.2.218	IDZ_vel . . . . .	101
13.18.2.219	IDZ_vel_e . . . . .	101
13.18.2.220	Tempo1 . . . . .	102
13.18.2.221	TimeEphemeris . . . . .	102
13.18.2.222	TNBandDMamp . . . . .	102
13.18.2.223	TNBandDMC . . . . .	102
13.18.2.224	TNBandDMGam . . . . .	102
13.18.2.225	TNBandNoiseAmp . . . . .	102
13.18.2.226	TNBandNoiseC . . . . .	102
13.18.2.227	TNBandNoiseGam . . . . .	102
13.18.2.228	TNBandNoiseHF . . . . .	103
13.18.2.229	TNBandNoiseLF . . . . .	103
13.18.2.230	TNDMAmp . . . . .	103
13.18.2.231	TNDMC . . . . .	103
13.18.2.232	TNDMCoeffs . . . . .	103
13.18.2.233	TNDMEvAmp . . . . .	103
13.18.2.234	TNDMEvGam . . . . .	103

13.18.2.23	ENDMEvLength	103
13.18.2.23	ENDMEvLin	104
13.18.2.23	ENDMEvOff	104
13.18.2.23	ENDMEvQuad	104
13.18.2.23	ENDMEvStart	104
13.18.2.24	ENDMGam	104
13.18.2.24	NECORRFlagID	104
13.18.2.24	NECORRFlagVal	104
13.18.2.24	NECORRVal	104
13.18.2.24	NEFFFlagID	105
13.18.2.24	NEFFFlagVal	105
13.18.2.24	NEFVal	105
13.18.2.24	NEQFlagID	105
13.18.2.24	NEQFlagVal	105
13.18.2.24	NEQVal	105
13.18.2.25	NGlobalEF	105
13.18.2.25	NGlobalEQ	105
13.18.2.25	NGroupNoiseAmp	106
13.18.2.25	NGroupNoiseC	106
13.18.2.25	NGroupNoiseFlagID	106
13.18.2.25	NGroupNoiseFlagVal	106
13.18.2.25	NGroupNoiseGam	106
13.18.2.25	NRedAmp	106
13.18.2.25	NRedC	106
13.18.2.25	NRedCoeffs	106
13.18.2.26	NRedCorner	107
13.18.2.26	NRedFlow	107
13.18.2.26	NRedGam	107
13.18.2.26	NShapeletEvFScale	107
13.18.2.26	NShapeletEvN	107

13.18.2.265	NShapeletEvPos	107
13.18.2.266	NShapeletEvWidth	107
13.18.2.267	NSQFlagID	107
13.18.2.268	NSQFlagVal	108
13.18.2.269	NSQVal	108
13.18.2.270	NsubtractDM	108
13.18.2.271	NsubtractRed	108
13.18.2.272	BaextraCovar	108
13.18.2.273	Offset	108
13.18.2.274	Offset_f1	108
13.18.2.275	Offset_f2	108
13.18.2.276	Offset_t1	109
13.18.2.277	Offset_t2	109
13.18.2.278	OffsetFlags	109
13.18.2.279	OffsetSite	109
13.18.2.280	Orsite	109
13.18.2.281	Units	109
13.18.2.282	UseCalceph	109
13.18.2.283	UseTNorth	109
13.18.2.284	UseIPulsar	110
13.18.2.285	Wave_cos	110
13.18.2.286	Wave_cos_dm	110
13.18.2.287	Wave_cos_dm_err	110
13.18.2.288	Wave_cos_err	110
13.18.2.289	Wave_sine	110
13.18.2.290	Wave_sine_dm	110
13.18.2.291	Wave_sine_dm_err	110
13.18.2.292	Wave_sine_err	111
13.18.2.293	WaveScale	111
13.18.2.294	WhiteNoiseModelFile	111

13.19storePrecision Struct Reference . . . . .	111
13.19.1 Member Data Documentation . . . . .	111
13.19.1.1 comment . . . . .	111
13.19.1.2 minPrec . . . . .	111
13.19.1.3 routine . . . . .	112
13.20T1Polyco Struct Reference . . . . .	112
13.20.1 Member Data Documentation . . . . .	112
13.20.1.1 binary_frequency . . . . .	112
13.20.1.2 binary_phase . . . . .	112
13.20.1.3 coeff . . . . .	113
13.20.1.4 date_string . . . . .	113
13.20.1.5 dm . . . . .	113
13.20.1.6 doppler . . . . .	113
13.20.1.7 frequency_obs . . . . .	113
13.20.1.8 frequency_psr_0 . . . . .	113
13.20.1.9 log10rms . . . . .	113
13.20.1.10mjd_mid . . . . .	113
13.20.1.11ncoeff . . . . .	114
13.20.1.12psrname . . . . .	114
13.20.1.13reference_phase . . . . .	114
13.20.1.14sitename . . . . .	114
13.20.1.15span . . . . .	114
13.20.1.16utc_string . . . . .	114
13.21T1PolycoSet Struct Reference . . . . .	114
13.21.1 Member Data Documentation . . . . .	115
13.21.1.1 nsegments . . . . .	115
13.21.1.2 segments . . . . .	115
13.22T2Predictor Struct Reference . . . . .	115
13.22.1 Member Data Documentation . . . . .	115
13.22.1.1 cheby . . . . .	116
13.22.1.2 kind . . . . .	116
13.22.1.3 modelset . . . . .	116
13.22.1.4 t1 . . . . .	116
13.23TabulatedFunction Struct Reference . . . . .	116
13.23.1 Member Data Documentation . . . . .	116
13.23.1.1 fileName . . . . .	117
13.23.1.2 header_line . . . . .	117
13.23.1.3 samples . . . . .	117
13.24TabulatedFunctionSample Struct Reference . . . . .	117
13.24.1 Member Data Documentation . . . . .	117
13.24.1.1 x . . . . .	117
13.24.1.2 y . . . . .	117

<b>14 File Documentation</b>	<b>119</b>
14.1 cholesky.h File Reference	119
14.1.1 Function Documentation	119
14.1.1.1 cholesky_covarFunc2matrix()	119
14.1.1.2 cholesky_dmModel()	120
14.1.1.3 cholesky_dmModelCovarParam()	120
14.1.1.4 cholesky_ecm()	120
14.1.1.5 cholesky_formUinv()	120
14.1.1.6 cholesky_powerlawModel()	121
14.1.1.7 cholesky_powerlawModel_withBeta()	121
14.1.1.8 cholesky_readFromCovarianceFunction()	121
14.2 choleskyRoutines.h File Reference	121
14.3 config.h File Reference	121
14.3.1 Macro Definition Documentation	122
14.3.1.1 _DARWIN_USE_64_BIT_INODE	122
14.3.1.2 F77_FUNC	123
14.3.1.3 F77_FUNC_	123
14.3.1.4 HAVE_BLAS	123
14.3.1.5 HAVE_CFITSIO	123
14.3.1.6 HAVE_DLERROR	123
14.3.1.7 HAVE_DLFCN_H	123
14.3.1.8 HAVE_FFTW3	123
14.3.1.9 HAVE_INTTYPES_H	124
14.3.1.10 HAVE_LAPACK	124
14.3.1.11 HAVE_LIBDL	124
14.3.1.12 HAVE_LIBDLLOADER	124
14.3.1.13 HAVE_LIBM	124
14.3.1.14 HAVE_MEMORY_H	124
14.3.1.15 HAVE_PGPLOT	124
14.3.1.16 HAVE_PTHREAD	124



14.3.1.17 HAVE_STDINT_H . . . . .	125
14.3.1.18 HAVE_STDLIB_H . . . . .	125
14.3.1.19 HAVE_STRING_H . . . . .	125
14.3.1.20 HAVE_STRINGS_H . . . . .	125
14.3.1.21 HAVE_SYS_STAT_H . . . . .	125
14.3.1.22 HAVE_SYS_TYPES_H . . . . .	125
14.3.1.23 HAVE_UNISTD_H . . . . .	125
14.3.1.24 LT_OBJDIR . . . . .	125
14.3.1.25 PACKAGE . . . . .	126
14.3.1.26 PACKAGE_BUGREPORT . . . . .	126
14.3.1.27 PACKAGE_NAME . . . . .	126
14.3.1.28 PACKAGE_STRING . . . . .	126
14.3.1.29 PACKAGE_TARNAME . . . . .	126
14.3.1.30 PACKAGE_URL . . . . .	126
14.3.1.31 PACKAGE_VERSION . . . . .	126
14.3.1.32 QR_DEFAULT . . . . .	126
14.3.1.33 STDC_HEADERS . . . . .	127
14.3.1.34 TEMPO2_ARCH . . . . .	127
14.3.1.35 VERSION . . . . .	127
14.4 constraints.h File Reference . . . . .	127
14.4.1 Function Documentation . . . . .	127
14.4.1.1 autosetDMCM() . . . . .	128
14.4.1.2 computeConstraintWeights() . . . . .	128
14.4.1.3 consFunc_dmmodel_cw() . . . . .	128
14.4.1.4 consFunc_dmmodel_cw_year() . . . . .	128
14.4.1.5 consFunc_dmmodel_dm1() . . . . .	128
14.4.1.6 consFunc_dmmodel_mean() . . . . .	129
14.4.1.7 consFunc_ifunc() . . . . .	129
14.4.1.8 consFunc_ifunc_year() . . . . .	129
14.4.1.9 consFunc_qifunc_c_year() . . . . .	129

14.4.1.10 consFunc_qifunc_p_year()	129
14.4.1.11 consFunc_quad_ifunc_c()	130
14.4.1.12 consFunc_quad_ifunc_p()	130
14.4.1.13 consFunc_tel_dx()	130
14.4.1.14 consFunc_tel_dy()	130
14.4.1.15 consFunc_tel_dz()	130
14.4.1.16 CONSTRAINTfuncs()	131
14.4.1.17 get_constraint_name()	131
14.4.1.18 standardConstraintFunctions()	131
14.5 constraints_covar.h File Reference	131
14.5.1 Function Documentation	131
14.5.1.1 constraints_covar_ifunc()	132
14.6 constraints_nestlike.h File Reference	132
14.6.1 Function Documentation	132
14.6.1.1 constraints_nestlike_band()	132
14.6.1.2 constraints_nestlike_group()	133
14.6.1.3 constraints_nestlike_jitter()	133
14.6.1.4 constraints_nestlike_red()	133
14.6.1.5 constraints_nestlike_red_dm()	133
14.7 constraints_param.h File Reference	134
14.7.1 Function Documentation	134
14.7.1.1 constraint_param_function()	134
14.8 documentation/1_USER_GUIDE.md File Reference	134
14.9 documentation/2_developers.md File Reference	134
14.10documentation/3_DEVELOPER_GUIDE.md File Reference	134
14.11documentation/4_directories.md File Reference	134
14.12documentation/5_plugins.md File Reference	134
14.13dynarr.h File Reference	134
14.13.1 Function Documentation	135
14.13.1.1 DynamicArray_free()	135

14.13.1.2 DynamicArray_init()	135
14.13.1.3 DynamicArray_push_back()	135
14.13.1.4 DynamicArray_resize()	135
14.14enum_str.h File Reference	136
14.14.1 Variable Documentation	136
14.14.1.1 constraint_str	136
14.14.1.2 label_str	136
14.15GWsim.h File Reference	136
14.15.1 Typedef Documentation	137
14.15.1.1 gwgeneralSrc	137
14.15.1.2 gwgenSpec	137
14.15.1.3 gwSrc	138
14.15.2 Function Documentation	138
14.15.2.1 calculateResidualgeneralGW()	138
14.15.2.2 calculateResidualGW()	138
14.15.2.3 dadt()	138
14.15.2.4 dedt()	138
14.15.2.5 dotProduct()	139
14.15.2.6 dtdt()	139
14.15.2.7 eccRes()	139
14.15.2.8 eccResWithEnergy()	139
14.15.2.9 Fe()	139
14.15.2.10Findphi()	140
14.15.2.11GWanisotropicbackground()	140
14.15.2.12GWbackground()	140
14.15.2.13GWbackground_read()	140
14.15.2.14GWbackground_write()	141
14.15.2.15GWdipolebackground()	141
14.15.2.16GWgeneralanisotropicbackground()	141
14.15.2.17GWgeneralbackground()	141

14.15.2.18	GWgeneralbackground_read()	142
14.15.2.19	GWgeneralbackground_write()	142
14.15.2.20	matrixMult()	142
14.15.2.21	psrange()	142
14.15.2.22	Rs()	142
14.15.2.23	setupgeneralGW()	142
14.15.2.24	setupGW()	143
14.15.2.25	setupPulsar_GWsim()	143
14.15.2.26	spharm()	143
14.16	ifteph.h File Reference	143
14.16.1	Macro Definition Documentation	144
14.16.1.1	IFTE_JD0	144
14.16.1.2	IFTE_K	144
14.16.1.3	IFTE_KM1	144
14.16.1.4	IFTE_LC	144
14.16.1.5	IFTE_MJD0	144
14.16.1.6	IFTE_TEPH0	144
14.16.2	Function Documentation	144
14.16.2.1	IFTE_close_file()	145
14.16.2.2	IFTE_DeltaT()	145
14.16.2.3	IFTE_DeltaTDot()	145
14.16.2.4	IFTE_get_DeltaT_DeltaTDot()	145
14.16.2.5	IFTE_get_vE()	145
14.16.2.6	IFTE_get_vE_vEDot()	145
14.16.2.7	IFTE_get_vEDot()	146
14.16.2.8	IFTE_init()	146
14.17	ifunc.h File Reference	146
14.17.1	Function Documentation	146
14.17.1.1	ifunc() [1/2]	146
14.17.1.2	ifunc() [2/2]	146

14.17.1.3 <code>sfunc()</code> . . . . .	147
14.18 <code>jpl_int.h</code> File Reference . . . . .	147
14.18.1 Macro Definition Documentation . . . . .	147
14.18.1.1 <code>JPL_HEADER_SIZE</code> . . . . .	147
14.18.1.2 <code>MAX_CHEBY</code> . . . . .	147
14.19 <code>jpleph.h</code> File Reference . . . . .	148
14.19.1 Macro Definition Documentation . . . . .	148
14.19.1.1 <code>DLL_FUNC</code> . . . . .	149
14.19.1.2 <code>JPL_EPH_FSEEK_ERROR</code> . . . . .	149
14.19.1.3 <code>JPL_EPH_INVALID_INDEX</code> . . . . .	149
14.19.1.4 <code>JPL_EPH_OUTSIDE_RANGE</code> . . . . .	149
14.19.1.5 <code>JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS</code> . . . . .	149
14.19.1.6 <code>JPL_EPH_READ_ERROR</code> . . . . .	149
14.19.1.7 <code>JPL_EPHEM_AU_IN_KM</code> . . . . .	149
14.19.1.8 <code>JPL_EPHEM_EARTH_MOON_RATIO</code> . . . . .	149
14.19.1.9 <code>JPL_EPHEM_END_JD</code> . . . . .	150
14.19.1.10 <code>JPL_EPHEM_EPHEMERIS_VERSION</code> . . . . .	150
14.19.1.11 <code>JPL_EPHEM_IPT_ARRAY</code> . . . . .	150
14.19.1.12 <code>JPL_EPHEM_KERNEL_NCOEFF</code> . . . . .	150
14.19.1.13 <code>JPL_EPHEM_KERNEL_RECORD_SIZE</code> . . . . .	150
14.19.1.14 <code>JPL_EPHEM_KERNEL_SIZE</code> . . . . .	150
14.19.1.15 <code>JPL_EPHEM_KERNEL_SWAP_BYTES</code> . . . . .	150
14.19.1.16 <code>JPL_EPHEM_N_CONSTANTS</code> . . . . .	150
14.19.1.17 <code>JPL_EPHEM_START_JD</code> . . . . .	151
14.19.1.18 <code>JPL_EPHEM_STEP</code> . . . . .	151
14.19.1.19 <code>jpl_get_pvsun</code> . . . . .	151
14.19.1.20 <code>JPL_INIT_FILE_CORRUPT</code> . . . . .	151
14.19.1.21 <code>JPL_INIT_FILE_NOT_FOUND</code> . . . . .	151
14.19.1.22 <code>JPL_INIT_FREAD2_FAILED</code> . . . . .	151
14.19.1.23 <code>JPL_INIT_FREAD3_FAILED</code> . . . . .	151

14.19.1.24JPL_INIT_FREAD4_FAILED . . . . .	152
14.19.1.25JPL_INIT_FREAD5_FAILED . . . . .	152
14.19.1.26JPL_INIT_FREAD_FAILED . . . . .	152
14.19.1.27JPL_INIT_FSEEK_FAILED . . . . .	152
14.19.1.28JPL_INIT_MEMORY_FAILURE . . . . .	152
14.19.1.29JPL_INIT_NO_ERROR . . . . .	152
14.19.1.30JPL_INIT_NOT_CALLED . . . . .	152
14.19.2 Function Documentation . . . . .	152
14.19.2.1 jpl_close_ephemeris() . . . . .	153
14.19.2.2 jpl_get_constant() . . . . .	153
14.19.2.3 jpl_get_double() . . . . .	153
14.19.2.4 jpl_get_long() . . . . .	153
14.19.2.5 jpl_init_ephemeris() . . . . .	153
14.19.2.6 jpl_init_error_code() . . . . .	153
14.19.2.7 jpl_pleph() . . . . .	154
14.19.2.8 jpl_state() . . . . .	154
14.19.2.9 make_sub_ephem() . . . . .	154
14.20read_fortran.h File Reference . . . . .	154
14.20.1 Function Documentation . . . . .	155
14.20.1.1 close_file() . . . . .	155
14.20.1.2 open_file() . . . . .	155
14.20.1.3 read_char() . . . . .	155
14.20.1.4 read_character() . . . . .	155
14.20.1.5 read_double() . . . . .	155
14.20.1.6 read_float() . . . . .	155
14.20.1.7 read_int() . . . . .	156
14.20.1.8 read_record_int() . . . . .	156
14.20.2 Variable Documentation . . . . .	156
14.20.2.1 c_fileptr . . . . .	156
14.20.2.2 swapByte . . . . .	156

14.21 read_fortran2.h File Reference	156
14.21.1 Function Documentation	157
14.21.1.1 close_file2()	157
14.21.1.2 open_file2()	157
14.21.1.3 read_character2()	157
14.21.1.4 read_double2()	157
14.21.1.5 read_float2()	157
14.21.1.6 read_int2()	157
14.21.1.7 read_record_int2()	158
14.21.2 Variable Documentation	158
14.21.2.1 c_fileptr2	158
14.21.2.2 swapByte2	158
14.22 README.md File Reference	158
14.23 T2accel.h File Reference	158
14.23.1 Macro Definition Documentation	159
14.23.1.1 ACCEL_LSQ	159
14.23.1.2 ACCEL_MULTMATRIX	159
14.23.1.3 ACCEL_UINV	159
14.23.2 Function Documentation	159
14.23.2.1 accel_lsqr()	159
14.23.2.2 accel_multMatrix()	159
14.23.2.3 accel_multMatrixVec()	160
14.23.2.4 accel_uinv()	160
14.23.3 Variable Documentation	160
14.23.3.1 useT2accel	160
14.24 t2fit.h File Reference	160
14.24.1 Function Documentation	160
14.24.1.1 t2Fit()	161
14.24.1.2 t2Fit_buildConstraintsMatrix()	161
14.24.1.3 t2Fit_buildDesignMatrix()	161

14.24.1.4 t2Fit_fillFitInfo()	161
14.24.1.5 t2Fit_fillGlobalFitInfo()	161
14.24.1.6 t2Fit_getFitData()	162
14.24.1.7 t2Fit_getParamDeriv()	162
14.24.1.8 t2Fit_getParamMatrixRow()	162
14.24.1.9 t2Fit_updateParameters()	162
14.25t2fit_dmmodel.h File Reference	162
14.25.1 Function Documentation	163
14.25.1.1 t2FitFunc_dmmodelCM()	163
14.25.1.2 t2FitFunc_dmmodelDM()	163
14.25.1.3 t2UpdateFunc_dmmodelCM()	163
14.25.1.4 t2UpdateFunc_dmmodelDM()	163
14.26t2fit_dmother.h File Reference	164
14.26.1 Function Documentation	164
14.26.1.1 t2FitFunc_dmsinusoids()	164
14.26.1.2 t2FitFunc_dmx()	164
14.26.1.3 t2FitFunc_fd()	164
14.26.1.4 t2FitFunc_fddc()	165
14.26.1.5 t2FitFunc_ne_sw()	165
14.26.1.6 t2UpdateFunc_ne_sw()	165
14.27t2fit_fitwaves.h File Reference	165
14.27.1 Function Documentation	165
14.27.1.1 t2FitFunc_fitwaves()	166
14.27.1.2 t2UpdateFunc_fitwaves()	166
14.28t2fit_glitch.h File Reference	166
14.28.1 Function Documentation	166
14.28.1.1 t2FitFunc_stdGlitch()	166
14.28.1.2 t2UpdateFunc_stdGlitch()	167
14.29t2fit_gw.h File Reference	167
14.29.1 Function Documentation	167



14.29.1.1 t2FitFunc_gwb_amp()	167
14.29.1.2 t2FitFunc_gwcs_amp()	167
14.29.1.3 t2FitFunc_gwm_amp()	168
14.29.1.4 t2FitFunc_gwsingle()	168
14.29.1.5 t2FitFunc_quad_om()	168
14.29.1.6 t2UpdateFunc_gwsingle()	168
14.29.1.7 t2UpdateFunc_quad_om()	169
14.30 t2fit_ifunc.h File Reference	169
14.30.1 Function Documentation	169
14.30.1.1 t2FitFunc_ifunc()	169
14.30.1.2 t2FitFunc_sifunc()	169
14.30.1.3 t2UpdateFunc_ifunc()	170
14.31 t2fit_nestlike.h File Reference	170
14.31.1 Function Documentation	170
14.31.1.1 t2FitFunc_nestlike_band()	170
14.31.1.2 t2FitFunc_nestlike_group()	171
14.31.1.3 t2FitFunc_nestlike_jitter()	171
14.31.1.4 t2FitFunc_nestlike_red()	171
14.31.1.5 t2FitFunc_nestlike_red_dm()	171
14.31.1.6 t2UpdateFunc_nestlike_band()	172
14.31.1.7 t2UpdateFunc_nestlike_group()	172
14.31.1.8 t2UpdateFunc_nestlike_jitter()	172
14.31.1.9 t2UpdateFunc_nestlike_red()	172
14.31.1.10 t2UpdateFunc_nestlike_red_dm()	173
14.32 t2fit_position.h File Reference	173
14.32.1 Function Documentation	173
14.32.1.1 t2FitFunc_stdPosition()	173
14.32.1.2 t2UpdateFunc_stdPosition()	173
14.33 t2fit_stdFitFuncs.h File Reference	174
14.33.1 Function Documentation	174

14.33.1.1 t2FitFunc_binaryModels()	174
14.33.1.2 t2FitFunc_ifunc()	175
14.33.1.3 t2FitFunc_jump()	175
14.33.1.4 t2FitFunc_notImplemented()	175
14.33.1.5 t2FitFunc_planet()	175
14.33.1.6 t2FitFunc_stdDm()	176
14.33.1.7 t2FitFunc_stdFreq()	176
14.33.1.8 t2FitFunc_telPos()	176
14.33.1.9 t2FitFunc_telPos_delta()	176
14.33.1.10 t2FitFunc_zero()	177
14.33.1.11 t2UpdateFunc_binaryModels()	177
14.33.1.12 t2UpdateFunc_ifunc()	177
14.33.1.13 t2UpdateFunc_jump()	177
14.33.1.14 t2UpdateFunc_notImplemented()	178
14.33.1.15 t2UpdateFunc_simpleAdd()	178
14.33.1.16 t2UpdateFunc_simpleMinus()	178
14.33.1.17 t2UpdateFunc_stdFreq()	178
14.33.1.18 t2UpdateFunc_telPos_delta()	179
14.33.1.19 t2UpdateFunc_zero()	179
14.34 T2toolkit.h File Reference	179
14.34.1 Detailed Description	180
14.34.2 Function Documentation	180
14.34.2.1 genrand_int32()	180
14.34.2.2 genrand_real1()	180
14.34.2.3 init_genrand()	180
14.34.2.4 TKconvertFloat1()	181
14.34.2.5 TKconvertFloat2()	181
14.34.2.6 TKfindMax_d()	181
14.34.2.7 TKfindMax_f()	181
14.34.2.8 TKfindMedian_d()	181

14.34.2.9 TKfindMedian_f()	181
14.34.2.10TKfindMin_d()	182
14.34.2.11TKfindMin_f()	182
14.34.2.12TKfindRMS_d()	182
14.34.2.13TKfindRMS_f()	182
14.34.2.14TKfindRMSweight_d()	182
14.34.2.15TKgaussDev()	182
14.34.2.16TKmean_d()	183
14.34.2.17TKmean_f()	183
14.34.2.18TKranDev()	183
14.34.2.19TKrange_d()	183
14.34.2.20TKrange_f()	183
14.34.2.21TKretMax_d()	183
14.34.2.22TKretMax_f()	184
14.34.2.23TKretMin_d()	184
14.34.2.24TKretMin_f()	184
14.34.2.25TKretMin_i()	184
14.34.2.26TKsetSeed()	184
14.34.2.27TKsign_d()	184
14.34.2.28TKsort_2f()	185
14.34.2.29TKsort_3d()	185
14.34.2.30TKsort_d()	185
14.34.2.31TKsort_f()	185
14.34.2.32TKvariance_d()	185
14.34.2.33TKzeromean_d()	185
14.35tabulatedfunction.h File Reference	186
14.35.1 Function Documentation	186
14.35.1.1 TabulatedFunction_getEndX()	186
14.35.1.2 TabulatedFunction_getStartX()	186
14.35.1.3 TabulatedFunction_getValue()	186

14.35.1.4 TabulatedFunction_load()	187
14.36tempo2.h File Reference	187
14.36.1 Detailed Description	193
14.36.2 Macro Definition Documentation	193
14.36.2.1 AU_DIST	193
14.36.2.2 AULTSC	193
14.36.2.3 BIG_G	193
14.36.2.4 DM_CONST	194
14.36.2.5 DM_CONST_SI	194
14.36.2.6 ECLIPTIC_OBLIQUITY_VAL	194
14.36.2.7 FB90_TIMEEPH	194
14.36.2.8 GM	194
14.36.2.9 GM_C3	194
14.36.2.10GMJ_C3	194
14.36.2.11GMN_C3	194
14.36.2.12GMS_C3	195
14.36.2.13GMU_C3	195
14.36.2.14GMV_C3	195
14.36.2.15HAVE_GWSIM_H	195
14.36.2.16F99_TIMEEPH	195
14.36.2.17FTEPH_FILE	195
14.36.2.18LEAPSECOND_FILE	195
14.36.2.19MASYR2RADS	195
14.36.2.20MAX_BPJ_JUMPS	196
14.36.2.21MAX_CLK_CORR	196
14.36.2.22MAX_CLKCORR	196
14.36.2.23MAX_COEFF	196
14.36.2.24MAX_COMPANIONS	196
14.36.2.25MAX_DM_DERIVATIVES	196
14.36.2.26MAX_DMX	196

14.36.2.27	MAX_FILELEN . . . . .	196
14.36.2.28	MAX_FIT . . . . .	197
14.36.2.29	MAX_FLAG_LEN . . . . .	197
14.36.2.30	MAX_FLAGS . . . . .	197
14.36.2.31	MAX_FREQ_DERIVATIVES . . . . .	197
14.36.2.32	MAX_IFUNC . . . . .	197
14.36.2.33	MAX_JUMPS . . . . .	197
14.36.2.34	MAX_LEAPSEC . . . . .	197
14.36.2.35	MAX_MSG . . . . .	197
14.36.2.36	MAX_OBSN_VAL . . . . .	198
14.36.2.37	MAX_PARAMS . . . . .	198
14.36.2.38	MAX_PSR_VAL . . . . .	198
14.36.2.39	MAX_QUAD . . . . .	198
14.36.2.40	MAX_SITE . . . . .	198
14.36.2.41	MAX_STOREPRECISION . . . . .	198
14.36.2.42	MAX_STRLEN . . . . .	198
14.36.2.43	MAX_SX . . . . .	198
14.36.2.44	MAX_T2EFAC . . . . .	199
14.36.2.45	MAX_T2EQUAD . . . . .	199
14.36.2.46	MAX_TEL_CLK_OFFS . . . . .	199
14.36.2.47	MAX_TEL_DX . . . . .	199
14.36.2.48	MAX_TEL_DY . . . . .	199
14.36.2.49	MAX_TEL_DZ . . . . .	199
14.36.2.50	MAX_TNBN . . . . .	199
14.36.2.51	MAX_TNDMEv . . . . .	199
14.36.2.52	MAX_TNECORR . . . . .	200
14.36.2.53	MAX_TNEF . . . . .	200
14.36.2.54	MAX_TNEQ . . . . .	200
14.36.2.55	MAX_TNGN . . . . .	200
14.36.2.56	MAX_TNSQ . . . . .	200

14.36.2.57	MAX_TOFFSET	200
14.36.2.58	MAX_WHITE	200
14.36.2.59	NE_SW_DEFAULT	200
14.36.2.60	OBLQ	201
14.36.2.61	OBSSYS_FILE	201
14.36.2.62	PCM	201
14.36.2.63	SECDAY	201
14.36.2.64	SECDAYI	201
14.36.2.65	SI_UNITS	201
14.36.2.66	SOLAR_MASS	201
14.36.2.67	SOLAR_RADIUS	201
14.36.2.68	SPEED_LIGHT	202
14.36.2.69	T2C_IAU2000B	202
14.36.2.70	T2C_TEMPO	202
14.36.2.71	TDB_UNITS	202
14.36.2.72	TDBTDT_FILE	202
14.36.2.73	TEMPO2_h_HASH	202
14.36.2.74	TEMPO2_h_MAJOR_VER	202
14.36.2.75	TEMPO2_h_MINOR_VER	202
14.36.2.76	TEMPO2_h_VER	203
14.36.2.77	TUN	203
14.36.2.78	UT1_FILE	203
14.36.3	Typedef Documentation	203
14.36.3.1	constraint_label	203
14.36.3.2	constraintDerivFunc	203
14.36.3.3	FitInfo	203
14.36.3.4	FitOutput	204
14.36.3.5	observation	204
14.36.3.6	param_label	204
14.36.3.7	paramDerivFunc	204

14.36.3.8 parameter . . . . .	204
14.36.3.9 paramUpdateFunc . . . . .	204
14.36.3.10 pulsar . . . . .	205
14.36.3.11 storePrecision . . . . .	205
14.36.4 Enumeration Type Documentation . . . . .	205
14.36.4.1 constraint . . . . .	205
14.36.4.2 label . . . . .	206
14.36.5 Function Documentation . . . . .	210
14.36.5.1 allocateMemory() . . . . .	210
14.36.5.2 autoConstraints() . . . . .	210
14.36.5.3 bootstrap() . . . . .	210
14.36.5.4 BTJmodel() . . . . .	210
14.36.5.5 BTmodel() . . . . .	210
14.36.5.6 BTXmodel() . . . . .	211
14.36.5.7 calcRMS() . . . . .	211
14.36.5.8 calculate_bclt() . . . . .	211
14.36.5.9 compute_tropospheric_delays() . . . . .	211
14.36.5.10 copyParam() . . . . .	211
14.36.5.11 copyPSR() . . . . .	211
14.36.5.12 CVSdisplayVersion() . . . . .	212
14.36.5.13 DDGRmodel() . . . . .	212
14.36.5.14 DDHmodel() . . . . .	212
14.36.5.15 DDKmodel() . . . . .	212
14.36.5.16 DDmodel() . . . . .	212
14.36.5.17 DDSmodel() . . . . .	213
14.36.5.18 defineClockCorrectionSequence() . . . . .	213
14.36.5.19 destroyMemory() . . . . .	213
14.36.5.20 destroyOne() . . . . .	213
14.36.5.21 displayMsg() . . . . .	213
14.36.5.22 displayParameters() . . . . .	213

14.36.5.23dm_delays()	214
14.36.5.24dms_turn()	214
14.36.5.25doFitAll()	214
14.36.5.26dotproduct()	214
14.36.5.27ELL1Hmodel()	214
14.36.5.28ELL1kmodel()	215
14.36.5.29ELL1model()	215
14.36.5.30equ2ecl()	215
14.36.5.31formBats()	215
14.36.5.32formBatsAll()	215
14.36.5.33formResiduals()	215
14.36.5.34fortran_mod()	216
14.36.5.35fortran_nint()	216
14.36.5.36fortran_nlong()	216
14.36.5.37get_EOP()	216
14.36.5.38get_obsCoord()	216
14.36.5.39get_obsCoord_IAU2000B()	216
14.36.5.40get_OneobsCoord()	217
14.36.5.41getCholeskyMatrix()	217
14.36.5.42getClockCorrections()	217
14.36.5.43getCorrection()	217
14.36.5.44getCorrectionTT()	217
14.36.5.45getInputs()	218
14.36.5.46getObservatory()	218
14.36.5.47getParamDeriv()	218
14.36.5.48getParameterValue()	218
14.36.5.49hms_turn()	219
14.36.5.50d_residual()	219
14.36.5.51initialise()	219
14.36.5.52initialiseOne()	219



14.36.5.53	<code>Vmodel()</code>	219
14.36.5.54	<code>logicFlag()</code>	219
14.36.5.55	<code>lookup_observatory_alias()</code>	220
14.36.5.56	<code>MSSmodel()</code>	220
14.36.5.57	<code>polyco()</code>	220
14.36.5.58	<code>preProcess()</code>	220
14.36.5.59	<code>preProcessSimple()</code>	220
14.36.5.60	<code>preProcessSimple1()</code>	221
14.36.5.61	<code>preProcessSimple2()</code>	221
14.36.5.62	<code>preProcessSimple3()</code>	221
14.36.5.63	<code>processFlag()</code>	221
14.36.5.64	<code>processSimultaneous()</code>	221
14.36.5.65	<code>readEphemeris()</code>	221
14.36.5.66	<code>readEphemeris_calceph()</code>	222
14.36.5.67	<code>readJBO_bat()</code>	222
14.36.5.68	<code>readObsFile()</code>	222
14.36.5.69	<code>readOneEphemeris()</code>	222
14.36.5.70	<code>readParfile()</code>	222
14.36.5.71	<code>readParfileGlobal()</code>	223
14.36.5.72	<code>readSimpleParfile()</code>	223
14.36.5.73	<code>readTimfile()</code>	223
14.36.5.74	<code>recordPrecision()</code>	223
14.36.5.75	<code>secularMotion()</code>	223
14.36.5.76	<code>setPlugPath()</code>	223
14.36.5.77	<code>setStart()</code>	224
14.36.5.78	<code>setupParameterFileDefaults()</code>	224
14.36.5.79	<code>shapiro_delay()</code>	224
14.36.5.80	<code>simplePlot()</code>	224
14.36.5.81	<code>solarWindModel()</code>	224
14.36.5.82	<code>sortToAs()</code>	224

14.36.5.83T2_PTAmodel()	225
14.36.5.84T2model()	225
14.36.5.85tai2tt()	225
14.36.5.86tai2ut1()	225
14.36.5.87textOutput()	225
14.36.5.88toa2utc()	226
14.36.5.89transform_units()	226
14.36.5.90tt2tb()	226
14.36.5.91tt2tb_calceph()	226
14.36.5.92turn_deg()	226
14.36.5.93turn_dms()	226
14.36.5.94turn_hms()	227
14.36.5.95updateBatsAll()	227
14.36.5.96updateBT()	227
14.36.5.97updateBTJ()	227
14.36.5.98updateBTX()	227
14.36.5.99updateDD()	228
14.36.5.100updateDDGR()	228
14.36.5.101updateDDH()	228
14.36.5.102updateDDK()	228
14.36.5.103updateDDS()	228
14.36.5.104updateELL1()	229
14.36.5.105updateELL1H()	229
14.36.5.106updateELL1k()	229
14.36.5.107updateEpoch()	229
14.36.5.108updateEpoch_str()	229
14.36.5.109updateJV()	230
14.36.5.110updateMSS()	230
14.36.5.111updateT2()	230
14.36.5.112updateT2_PTA()	230

14.36.5.112	<a href="#">UseSelectFile()</a>	230
14.36.5.113	<a href="#">UseC2tai()</a>	231
14.36.5.114	<a href="#">SectorPulsar()</a>	231
14.36.5.115	<a href="#">Sectorscale()</a>	231
14.36.5.116	<a href="#">Sectorsum()</a>	231
14.36.5.117	<a href="#">WriteTim()</a>	231
14.36.5.118	<a href="#">Zoom_graphics()</a>	231
14.36.6	<a href="#">Variable Documentation</a>	232
14.36.6.1	<a href="#">covarFuncFile</a>	232
14.36.6.2	<a href="#">dcmFile</a>	232
14.36.6.3	<a href="#">displayCVSversion</a>	232
14.36.6.4	<a href="#">ECLIPTIC_OBLIQUITY</a>	232
14.36.6.5	<a href="#">forceGlobalFit</a>	232
14.36.6.6	<a href="#">MAX_OBSN</a>	232
14.36.6.7	<a href="#">MAX_PSR</a>	232
14.36.6.8	<a href="#">NEWFIT</a>	233
14.36.6.9	<a href="#">tempo2_clock_path</a>	233
14.36.6.10	<a href="#">TEMPO2_ENVIRON</a>	233
14.36.6.11	<a href="#">tempo2_plug_path</a>	233
14.36.6.12	<a href="#">tempo2_plug_path_len</a>	233
14.36.6.13	<a href="#">tempo2MachineType</a>	233
14.36.6.14	<a href="#">veryFast</a>	233
14.37	<a href="#">tempo2pred.h File Reference</a>	234
14.37.1	<a href="#">Enumeration Type Documentation</a>	234
14.37.1.1	<a href="#">T2PredictorKind</a>	234
14.37.2	<a href="#">Function Documentation</a>	235
14.37.2.1	<a href="#">T2Predictor_Copy()</a>	235
14.37.2.2	<a href="#">T2Predictor_Destroy()</a>	235
14.37.2.3	<a href="#">T2Predictor_FRead()</a>	235
14.37.2.4	<a href="#">T2Predictor_FWrite()</a>	235

14.37.2.5 T2Predictor_GetEndFreq()	235
14.37.2.6 T2Predictor_GetEndMJD()	236
14.37.2.7 T2Predictor_GetFrequency()	236
14.37.2.8 T2Predictor_GetPhase()	236
14.37.2.9 T2Predictor_GetPlan()	236
14.37.2.10T2Predictor_GetPlan_Ext()	236
14.37.2.11T2Predictor_GetPSRName()	237
14.37.2.12T2Predictor_GetSiteName()	237
14.37.2.13T2Predictor_GetStartFreq()	237
14.37.2.14T2Predictor_GetStartMJD()	237
14.37.2.15T2Predictor_Init()	237
14.37.2.16T2Predictor_Insert()	237
14.37.2.17T2Predictor_Keep()	237
14.37.2.18T2Predictor_Kind()	238
14.37.2.19T2Predictor_Read()	238
14.37.2.20T2Predictor_Write()	238
14.37.3 Variable Documentation	238
14.37.3.1 ChebyModelSet_OutOfRange	238
14.38tempo2pred_int.h File Reference	238
14.38.1 Function Documentation	239
14.38.1.1 Cheby2D_Construct()	239
14.38.1.2 Cheby2D_Construct_x_Derivative()	240
14.38.1.3 Cheby2D_Test()	240
14.38.1.4 ChebyModel_Construct()	240
14.38.1.5 ChebyModel_Copy()	240
14.38.1.6 ChebyModel_Destroy()	240
14.38.1.7 ChebyModel_GetFrequency()	240
14.38.1.8 ChebyModel_GetPhase()	241
14.38.1.9 ChebyModel_Init()	241
14.38.1.10ChebyModel_Read()	241

14.38.1.11ChebyModel_Test()	241
14.38.1.12ChebyModel_Write()	241
14.38.1.13ChebyModelSet_Construct()	242
14.38.1.14ChebyModelSet_Destroy()	242
14.38.1.15ChebyModelSet_GetFrequency()	242
14.38.1.16ChebyModelSet_GetNearest()	242
14.38.1.17ChebyModelSet_GetPhase()	242
14.38.1.18ChebyModelSet_Init()	243
14.38.1.19ChebyModelSet_Insert()	243
14.38.1.20ChebyModelSet_Keep()	243
14.38.1.21ChebyModelSet_Read()	243
14.38.1.22ChebyModelSet_Test()	243
14.38.1.23ChebyModelSet_Write()	243
14.38.1.24T1Polyco_GetFrequency()	244
14.38.1.25T1Polyco_GetPhase()	244
14.38.1.26T1Polyco_Read()	244
14.38.1.27T1Polyco_Write()	244
14.38.1.28T1PolycoSet_Destroy()	244
14.38.1.29T1PolycoSet_GetFrequency()	244
14.38.1.30T1PolycoSet_GetNearest()	245
14.38.1.31T1PolycoSet_GetPhase()	245
14.38.1.32T1PolycoSet_Read()	245
14.38.1.33T1PolycoSet_Write()	245
14.39tempo2Util.h File Reference	245
14.39.1 Function Documentation	245
14.39.1.1 dms_turn()	245
14.39.1.2 hms_turn()	246
14.39.1.3 turn_deg()	246
14.40TKcholesky.h File Reference	246
14.40.1 Function Documentation	246

14.40.1.1	cholesky_covarFunc2matrix()	246
14.40.1.2	cholesky_dmModel()	247
14.40.1.3	cholesky_dmModelCovarParam()	247
14.40.1.4	cholesky_ecm()	247
14.40.1.5	cholesky_formUinv()	247
14.40.1.6	cholesky_powerlawModel()	248
14.40.1.7	cholesky_powerlawModel_withBeta()	248
14.40.1.8	cholesky_readFromCovarianceFunction()	248
14.41	TKfit.h File Reference	248
14.41.1	Function Documentation	249
14.41.1.1	TKconstrainedLeastSquares()	249
14.41.1.2	TKfindPoly_d()	249
14.41.1.3	TKfitPoly()	250
14.41.1.4	TKleastSquares()	250
14.41.1.5	TKleastSquares_svd()	250
14.41.1.6	TKleastSquares_svd_noErr()	250
14.41.1.7	TKremovePoly_d()	251
14.41.1.8	TKremovePoly_f()	251
14.41.1.9	TKrobustConstrainedLeastSquares()	251
14.41.1.10	TKrobustDefConstrainedLeastSquares()	251
14.41.1.11	TKrobustLeastSquares()	252
14.42	TKlog.h File Reference	252
14.42.1	Macro Definition Documentation	253
14.42.1.1	_LOG	253
14.42.1.2	BOLDCOLOR	253
14.42.1.3	DEPRECATED	254
14.42.1.4	ENDERR	254
14.42.1.5	ENDL	254
14.42.1.6	ERRORCOLOR	254
14.42.1.7	LOG_OUTFILE	254

14.42.1.8 logall . . . . .	254
14.42.1.9 logdbg . . . . .	254
14.42.1.10logerr . . . . .	255
14.42.1.11logmsg . . . . .	255
14.42.1.12logtchk . . . . .	255
14.42.1.13logwarn . . . . .	255
14.42.1.14RESETCOLOR . . . . .	255
14.42.1.15TK_MAX_ERROR_LEN . . . . .	255
14.42.1.16TK_MAX_ERRORS . . . . .	256
14.42.1.17TK_STORE_ERROR . . . . .	256
14.42.1.18TK_STORE_WARNING . . . . .	256
14.42.1.19WARNCOLOR . . . . .	256
14.42.1.20WHEREARG . . . . .	256
14.42.1.21WHEREERR . . . . .	256
14.42.1.22WHERESTR . . . . .	256
14.42.1.23WHERECHK . . . . .	257
14.42.1.24WHEREWARN . . . . .	257
14.42.2 Function Documentation . . . . .	257
14.42.2.1 _TKchklog() . . . . .	257
14.42.2.2 logerr_check() . . . . .	257
14.42.3 Variable Documentation . . . . .	257
14.42.3.1 debugFlag . . . . .	257
14.42.3.2 quietFlag . . . . .	257
14.42.3.3 tcheck . . . . .	258
14.42.3.4 timer_clk . . . . .	258
14.42.3.5 TK_errorCount . . . . .	258
14.42.3.6 TK_errorlog . . . . .	258
14.42.3.7 TK_warnCount . . . . .	258
14.42.3.8 TK_warnlog . . . . .	258
14.42.3.9 writeResiduals . . . . .	258

14.43TKlongdouble.float128.h File Reference . . . . .	258
14.43.1 Macro Definition Documentation . . . . .	259
14.43.1.1 cosl . . . . .	259
14.43.1.2 fabsl . . . . .	259
14.43.1.3 floorl . . . . .	259
14.43.1.4 FMT_LD . . . . .	260
14.43.1.5 LD_PI . . . . .	260
14.43.1.6 longdouble . . . . .	260
14.43.1.7 LONGDOUBLE_IS_FLOAT128 . . . . .	260
14.43.1.8 LONGDOUBLE_ONE . . . . .	260
14.43.1.9 powl . . . . .	260
14.43.1.10sinl . . . . .	260
14.43.1.11USE_BUILTIN_LONGDOUBLE . . . . .	260
14.43.2 Typedef Documentation . . . . .	261
14.43.2.1 longdouble . . . . .	261
14.43.3 Function Documentation . . . . .	261
14.43.3.1 ld_fprintf() . . . . .	261
14.43.3.2 ld_printf() . . . . .	261
14.43.3.3 ld_sprintf() . . . . .	261
14.43.3.4 parse_longdouble() . . . . .	261
14.44TKlongdouble.h File Reference . . . . .	262
14.44.1 Macro Definition Documentation . . . . .	262
14.44.1.1 ld_fprintf . . . . .	262
14.44.1.2 LD_PI . . . . .	262
14.44.1.3 ld_printf . . . . .	262
14.44.1.4 ld_sprintf . . . . .	263
14.44.1.5 longdouble . . . . .	263
14.44.1.6 LONGDOUBLE_IS_IEEE754 . . . . .	263
14.44.1.7 LONGDOUBLE_ONE . . . . .	263
14.44.1.8 USE_BUILTIN_LONGDOUBLE . . . . .	263



14.44.2 Typedef Documentation . . . . .	263
14.44.2.1 longdouble . . . . .	263
14.44.3 Function Documentation . . . . .	263
14.44.3.1 parse_longdouble() . . . . .	263
14.45TKlongdouble.ld.h File Reference . . . . .	264
14.45.1 Macro Definition Documentation . . . . .	264
14.45.1.1 ld_fprintf . . . . .	264
14.45.1.2 LD_PI . . . . .	264
14.45.1.3 ld_printf . . . . .	264
14.45.1.4 ld_sprintf . . . . .	265
14.45.1.5 longdouble . . . . .	265
14.45.1.6 LONGDOUBLE_IS_IEEE754 . . . . .	265
14.45.1.7 LONGDOUBLE_ONE . . . . .	265
14.45.1.8 USE_BUILTIN_LONGDOUBLE . . . . .	265
14.45.2 Typedef Documentation . . . . .	265
14.45.2.1 longdouble . . . . .	265
14.45.3 Function Documentation . . . . .	265
14.45.3.1 parse_longdouble() . . . . .	265
14.46TKmatrix.h File Reference . . . . .	266
14.46.1 Function Documentation . . . . .	266
14.46.1.1 free_2df() . . . . .	266
14.46.1.2 free_blas() . . . . .	266
14.46.1.3 free_uinv() . . . . .	266
14.46.1.4 get_blas_cols() . . . . .	267
14.46.1.5 get_blas_rows() . . . . .	267
14.46.1.6 malloc_2df() . . . . .	267
14.46.1.7 malloc_blas() . . . . .	267
14.46.1.8 malloc_uinv() . . . . .	267
14.46.1.9 TKmultMatrix() . . . . .	267
14.46.1.10TKmultMatrix_sq() . . . . .	268

14.46.1.1TKmultMatrixVec()	268
14.46.1.12TKmultMatrixVec_sq()	268
14.47TKrobust.h File Reference	268
14.47.1 Function Documentation	268
14.47.1.1 TKrobust()	269
14.48TKspectrum.h File Reference	269
14.48.1 Macro Definition Documentation	270
14.48.1.1 ABS	270
14.48.1.2 MAX	270
14.48.1.3 MIN	270
14.48.2 Typedef Documentation	271
14.48.2.1 complexVal	271
14.48.3 Function Documentation	271
14.48.3.1 calcSpectra()	271
14.48.3.2 calcSpectraErr()	271
14.48.3.3 calcSpectraErr_complex()	271
14.48.3.4 fit4()	272
14.48.3.5 getprtj()	272
14.48.3.6 getweights()	272
14.48.3.7 indexx8()	272
14.48.3.8 mat20()	272
14.48.3.9 sineFunc()	273
14.48.3.10TK_dft()	273
14.48.3.11TK_fft()	273
14.48.3.12TK_fitSine()	273
14.48.3.13TK_fitSinusoids()	274
14.48.3.14TK_weightLS()	274
14.48.3.15TKaveragePts()	274
14.48.3.16TKboxcar()	274
14.48.3.17TKcmonot()	275

14.48.3.18TKfirstDifference()	275
14.48.3.19TKhann()	275
14.48.3.20TKinterpolateSplineSmoothFixedXPts()	275
14.48.3.21TKlomb_d()	275
14.48.3.22TKsortit()	276
14.48.3.23TKspectrum()	276
14.48.3.24TKspline_interpolate()	276
14.48.4 Variable Documentation	276
14.48.4.1 verbose_calc_spectra	277
14.49TKsvd.h File Reference	277
14.49.1 Function Documentation	277
14.49.1.1 TKbacksubstitution_svd()	277
14.49.1.2 TKbidiagonal()	277
14.49.1.3 TKpythag()	278
14.49.1.4 TKsingularValueDecomposition_lsqr()	278
<b>Index</b>	<b>279</b>



## 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](mailto:george.hobbs@csiro.au)
  - Core tempo2 development.
  - Gravitational wave codes.
  - Binary models.
- Michael Keith [MK][mkeith@pulsarastronomy.net](mailto:mkeith@pulsarastronomy.net)
  - C++ code maintenance.
  - Linear algebra and least-squares algorithms.
  - Build system maintenance.
  - 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.

    // use stdint 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 \$TEMPO2 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

# 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 dependancies 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 PGLOT (e.g. setenv F77 gfortran, if you used gfortran for PGLOT)

### 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-6 and gxx-6 or similar. Use this with:

```
export CC=gcc-6
export CXX=g++-6
```

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 . . . . .	<a href="#">27</a>
libtempo2 External API . . . . .	<a href="#">28</a>



## Chapter 10

# Class Index

### 10.1 Class List

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

<a href="#">Cheby2D</a>	29
<a href="#">ChebyModel</a>	30
<a href="#">ChebyModelSet</a>	31
<a href="#">clock_correction</a>	32
<a href="#">complexVal</a>	33
<a href="#">constraint_param_info</a>	33
<a href="#">DynamicArray</a>	34
<a href="#">FitInfo</a>	
Details of the fit	35
<a href="#">FitOutput</a>	37
<a href="#">gwgeneralSrc</a>	39
<a href="#">gwgenSpec</a>	43
<a href="#">gwSrc</a>	44
<a href="#">interpolation_info</a>	47
<a href="#">jpl_eph_data</a>	48
<a href="#">observation</a>	
A struct containing the details of a single obesrvation	51
<a href="#">observatory</a>	64
<a href="#">parameter</a>	
Holds the values for a parameter	66
<a href="#">pulsar</a>	
Details for a single pulsar	68
<a href="#">storePrecision</a>	111
<a href="#">T1Polyco</a>	112
<a href="#">T1PolycoSet</a>	114
<a href="#">T2Predictor</a>	115
<a href="#">TabulatedFunction</a>	116
<a href="#">TabulatedFunctionSample</a>	117



# Chapter 11

## File Index

### 11.1 File List

Here is a list of all files with brief descriptions:

<a href="#">cholesky.h</a>	119
<a href="#">choleskyRoutines.h</a>	121
<a href="#">config.h</a>	121
<a href="#">constraints.h</a>	127
<a href="#">constraints_covar.h</a>	131
<a href="#">constraints_nestlike.h</a>	132
<a href="#">constraints_param.h</a>	134
<a href="#">dynarr.h</a>	134
<a href="#">enum_str.h</a>	136
<a href="#">GWsim.h</a>	136
<a href="#">ifteph.h</a>	143
<a href="#">ifunc.h</a>	146
<a href="#">jpl_int.h</a>	147
<a href="#">jpleph.h</a>	148
<a href="#">read_fortran.h</a>	154
<a href="#">read_fortran2.h</a>	156
<a href="#">T2accel.h</a>	158
<a href="#">t2fit.h</a>	160
<a href="#">t2fit_dmmodel.h</a>	162
<a href="#">t2fit_dmother.h</a>	164
<a href="#">t2fit_fitwaves.h</a>	165
<a href="#">t2fit_glitch.h</a>	166
<a href="#">t2fit_gw.h</a>	167
<a href="#">t2fit_ifunc.h</a>	169
<a href="#">t2fit_nestlike.h</a>	170
<a href="#">t2fit_position.h</a>	173
<a href="#">t2fit_stdFitFuncs.h</a>	174
<a href="#">T2toolkit.h</a>	
Set of routines that are commonly used in tempo2 and/or its plugins	179
<a href="#">tabulatedfunction.h</a>	186
<a href="#">tempo2.h</a>	
Main interface to libtempo2	187
<a href="#">tempo2pred.h</a>	234
<a href="#">tempo2pred_int.h</a>	238
<a href="#">tempo2Util.h</a>	245

TKcholesky.h	246
TKfit.h	248
TKlog.h	252
TKlongdouble.float128.h	258
TKlongdouble.h	262
TKlongdouble.ld.h	264
TKmatrix.h	266
TKrobust.h	268
TKspectrum.h	269
TKsvd.h	277

## 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 [coeff](#)

```
long double* Cheby2D::coeff
```

##### 13.1.1.2 [nx](#)

```
int Cheby2D::nx
```

##### 13.1.1.3 [ny](#)

```
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 [cheby](#)

[Cheby2D](#) ChebyModel::cheby

#### 13.2.1.2 [dispersion\\_constant](#)

long double ChebyModel::dispersion\_constant

#### 13.2.1.3 [freq\\_end](#)

long double ChebyModel::freq\_end

#### 13.2.1.4 [freq\\_start](#)

long double ChebyModel::freq\_start

#### 13.2.1.5 frequency\_cheby

[Cheby2D](#) ChebyModel::frequency\_cheby

#### 13.2.1.6 mjd\_end

long double ChebyModel::mjd\_end

#### 13.2.1.7 mjd\_start

long double ChebyModel::mjd\_start

#### 13.2.1.8 psrname

char ChebyModel::psrname[64]

#### 13.2.1.9 sitename

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 nsegments

```
int ChebyModelSet::nsegments
```

#### 13.3.1.2 segments

```
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 correction

```
double clock_correction::correction
```

#### 13.4.2.2 corrects\_to

```
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 imag

```
double complexVal::imag
```

#### 13.5.1.2 real

```
double complexVal::real
```

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

- [TKspectrum.h](#)

## 13.6 constraint\_param\_info Struct Reference

```
#include <constraints_param.h>
```

### Public Attributes

- int [param](#)
- int [param\\_k](#)
- double [val](#)
- double [err](#)

### 13.6.1 Member Data Documentation

#### 13.6.1.1 err

```
double constraint_param_info::err
```

#### 13.6.1.2 param

```
int constraint_param_info::param
```

#### 13.6.1.3 param\_k

```
int constraint_param_info::param_k
```

#### 13.6.1.4 val

```
double constraint_param_info::val
```

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

- [constraints\\_param.h](#)

## 13.7 DynamicArray Struct Reference

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

### Public Attributes

- void \* [data](#)
- size\_t [nelem](#)
- size\_t [elem\\_size](#)
- size\_t [nallocated](#)

### 13.7.1 Member Data Documentation

#### 13.7.1.1 data

```
void* DynamicArray::data
```

#### 13.7.1.2 elem\_size

```
size_t DynamicArray::elem_size
```

#### 13.7.1.3 nallocated

```
size_t DynamicArray::nallocated
```

#### 13.7.1.4 nelem

```
size_t DynamicArray::nelem
```

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

- [dynarr.h](#)

## 13.8 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]
- void \* [constraintSpecial](#) [MAX\_FIT]
- double [constraintValue](#) [MAX\_FIT]
- [paramUpdateFunc](#) [updateFunctions](#) [MAX\_FIT]
- [FitOutput](#) [output](#)

### 13.8.1 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.

### 13.8.2 Member Data Documentation

#### 13.8.2.1 constraintCounters

```
int FitInfo::constraintCounters[MAX\_FIT]
```

#### 13.8.2.2 constraintDerivs

```
constraintDerivFunc FitInfo::constraintDerivs[MAX\_FIT]
```

#### 13.8.2.3 constraintIndex

```
constraint\_label FitInfo::constraintIndex[MAX\_FIT]
```

#### 13.8.2.4 constraintSpecial

```
void* FitInfo::constraintSpecial[MAX\_FIT]
```

#### 13.8.2.5 constraintValue

```
double FitInfo::constraintValue[MAX\_FIT]
```

#### 13.8.2.6 nConstraints

```
unsigned FitInfo::nConstraints
```



### 13.8.2.7 nParams

unsigned FitInfo::nParams

### 13.8.2.8 output

FitOutput FitInfo::output

### 13.8.2.9 paramCounters

int FitInfo::paramCounters[MAX\_FIT]

### 13.8.2.10 paramDerivs

paramDerivFunc FitInfo::paramDerivs[MAX\_FIT]

### 13.8.2.11 paramIndex

param\_label FitInfo::paramIndex[MAX\_FIT]

### 13.8.2.12 updateFunctions

paramUpdateFunc FitInfo::updateFunctions[MAX\_FIT]

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

- [tempo2.h](#)

## 13.9 FitOutput Struct Reference

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

## Public Attributes

- double [parameterEstimates](#) [[MAX\\_FIT](#)]
- double [errorEstimates](#) [[MAX\\_FIT](#)]
- int [indexPsr](#) [[MAX\\_FIT](#)]
- [param\\_label](#) [indexParam](#) [[MAX\\_FIT](#)]
- int [indexCounter](#) [[MAX\\_FIT](#)]
- int [totalNfit](#)

### 13.9.1 Member Data Documentation

#### 13.9.1.1 [errorEstimates](#)

```
double FitOutput::errorEstimates[MAX\_FIT]
```

#### 13.9.1.2 [indexCounter](#)

```
int FitOutput::indexCounter[MAX\_FIT]
```

#### 13.9.1.3 [indexParam](#)

```
param\_label FitOutput::indexParam[MAX\_FIT]
```

#### 13.9.1.4 [indexPsr](#)

```
int FitOutput::indexPsr[MAX\_FIT]
```

#### 13.9.1.5 [parameterEstimates](#)

```
double FitOutput::parameterEstimates[MAX\_FIT]
```

#### 13.9.1.6 totalNfit

```
int FitOutput::totalNfit
```

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

- [tempo2.h](#)

## 13.10 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 \[3\]](#)

### 13.10.1 Member Data Documentation

#### 13.10.1.1 across\_g

```
longdouble gwgeneralSrc::across_g
```

**13.10.1.2 across\_im\_g**

`longdouble` gwgeneralSrc::across\_im\_g

**13.10.1.3 aplus\_g**

`longdouble` gwgeneralSrc::aplug

**13.10.1.4 aplus\_im\_g**

`longdouble` gwgeneralSrc::aplug\_im\_g

**13.10.1.5 asl\_g**

`longdouble` gwgeneralSrc::asl\_g

**13.10.1.6 asl\_im\_g**

`longdouble` gwgeneralSrc::asl\_im\_g

**13.10.1.7 ast\_g**

`longdouble` gwgeneralSrc::ast\_g

**13.10.1.8 ast\_im\_g**

`longdouble` gwgeneralSrc::ast\_im\_g

**13.10.1.9 avx\_g**

`longdouble` gwgeneralSrc::avx\_g

**13.10.1.10 avx\_im\_g**

`longdouble` gwgeneralSrc::avx\_im\_g

**13.10.1.11 avy\_g**

`longdouble` gwgeneralSrc::avy\_g

**13.10.1.12 avy\_im\_g**

`longdouble` gwgeneralSrc::avy\_im\_g

**13.10.1.13 dist\_bin**

`longdouble` gwgeneralSrc::dist\_bin

**13.10.1.14 h**

`longdouble` gwgeneralSrc::h[3][3]

**13.10.1.15 h\_im**

`longdouble` gwgeneralSrc::h\_im[3][3]

**13.10.1.16 inc\_bin**

`longdouble` gwgeneralSrc::inc\_bin

**13.10.1.17 kg**

`longdouble` gwgeneralSrc::kg[3]

**13.10.1.18 omega\_g**

`longdouble` gwgeneralSrc::omega\_g

**13.10.1.19 phase\_g**

`longdouble` gwgeneralSrc::phase\_g

**13.10.1.20 phi\_bin**

`longdouble` gwgeneralSrc::phi\_bin

**13.10.1.21 phi\_g**

`longdouble` gwgeneralSrc::phi\_g

**13.10.1.22 phi\_polar\_g**

`longdouble` gwgeneralSrc::phi\_polar\_g

**13.10.1.23 theta\_bin**

`longdouble` gwgeneralSrc::theta\_bin

**13.10.1.24 theta\_g**

`longdouble` gwgeneralSrc::theta\_g

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

- [GWSim.h](#)

## 13.11 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](#)

### 13.11.1 Member Data Documentation

#### 13.11.1.1 sl\_alpha

```
double gwgenSpec::sl_alpha
```

#### 13.11.1.2 sl\_amp

```
double gwgenSpec::sl_amp
```

#### 13.11.1.3 st\_alpha

```
double gwgenSpec::st_alpha
```

#### 13.11.1.4 st\_amp

```
double gwgenSpec::st_amp
```

#### 13.11.1.5 tensor\_alpha

```
double gwgenSpec::tensor_alpha
```

#### 13.11.1.6 tensor\_amp

```
double gwgenSpec::tensor_amp
```

#### 13.11.1.7 vl\_alpha

```
double gwgenSpec::vl_alpha
```

#### 13.11.1.8 vl\_amp

```
double gwgenSpec::vl_amp
```

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

- [GWsim.h](#)

## 13.12 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 \[3\]](#)



### 13.12.1 Member Data Documentation

#### 13.12.1.1 across\_g

`longdouble gwSrc::across_g`

#### 13.12.1.2 across\_im\_g

`longdouble gwSrc::across_im_g`

#### 13.12.1.3 aplus\_g

`longdouble gwSrc::aplus_g`

#### 13.12.1.4 aplus\_im\_g

`longdouble gwSrc::aplus_im_g`

#### 13.12.1.5 dist\_bin

`longdouble gwSrc::dist_bin`

#### 13.12.1.6 h

`longdouble gwSrc::h[3][3]`

#### 13.12.1.7 h\_im

`longdouble gwSrc::h_im[3][3]`

**13.12.1.8 inc\_bin**

`longdouble` gwSrc::inc\_bin

**13.12.1.9 kg**

`longdouble` gwSrc::kg[3]

**13.12.1.10 omega\_g**

`longdouble` gwSrc::omega\_g

**13.12.1.11 phase\_g**

`longdouble` gwSrc::phase\_g

**13.12.1.12 phi\_bin**

`longdouble` gwSrc::phi\_bin

**13.12.1.13 phi\_g**

`longdouble` gwSrc::phi\_g

**13.12.1.14 phi\_polar\_g**

`longdouble` gwSrc::phi\_polar\_g

**13.12.1.15 theta\_bin**

`longdouble` gwSrc::theta\_bin

#### 13.12.1.16 theta\_g

```
longdouble gwSrc::theta_g
```

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

- [GWsim.h](#)

## 13.13 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](#)

### 13.13.1 Member Data Documentation

#### 13.13.1.1 n\_posn\_avail

```
unsigned interpolation_info::n_posn_avail
```

#### 13.13.1.2 n\_vel\_avail

```
unsigned interpolation_info::n_vel_avail
```

#### 13.13.1.3 posn\_coeff

```
double interpolation_info::posn_coeff [MAX\_CHEBY]
```

#### 13.13.1.4 twot

```
double interpolation_info::twot
```

#### 13.13.1.5 vel\_coeff

```
double interpolation_info::vel_coeff[MAX_CHEBY]
```

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

- [jpl\\_int.h](#)

### 13.14 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](#)

#### 13.14.1 Member Data Documentation

**13.14.1.1 au**

```
double jpl_eph_data::au
```

**13.14.1.2 cache**

```
double* jpl_eph_data::cache
```

**13.14.1.3 curr\_cache\_loc**

```
uint32_t jpl_eph_data::curr_cache_loc
```

**13.14.1.4 emrat**

```
double jpl_eph_data::emrat
```

**13.14.1.5 ephem\_end**

```
double jpl_eph_data::ephem_end
```

**13.14.1.6 ephem\_start**

```
double jpl_eph_data::ephem_start
```

**13.14.1.7 ephem\_step**

```
double jpl_eph_data::ephem_step
```

**13.14.1.8 ephemeris\_version**

```
uint32_t jpl_eph_data::ephemeris_version
```

#### 13.14.1.9 ifile

```
FILE* jpl_eph_data::ifile
```

#### 13.14.1.10 iinfo

```
struct interpolation_info jpl_eph_data::iinfo
```

#### 13.14.1.11 ipt

```
uint32_t jpl_eph_data::ipt[15][3]
```

#### 13.14.1.12 kernel\_size

```
uint32_t jpl_eph_data::kernel_size
```

#### 13.14.1.13 ncoeff

```
uint32_t jpl_eph_data::ncoeff
```

#### 13.14.1.14 ncon

```
uint32_t jpl_eph_data::ncon
```

#### 13.14.1.15 pvsun

```
double jpl_eph_data::pvsun[9]
```

#### 13.14.1.16 pvsun\_t

```
double jpl_eph_data::pvsun_t
```

#### 13.14.1.17 `resize`

```
uint32_t jpl_eph_data::resize
```

#### 13.14.1.18 `swap_bytes`

```
uint32_t jpl_eph_data::swap_bytes
```

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

- [jpl\\_int.h](#)

## 13.15 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](#)
- [longdouble residualtn](#)
- [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](#)
- double [snr](#)
- double [pnoise](#)
- double [tobs](#)
- double [chisq](#)
- double [bline](#)

### 13.15.1 Detailed Description

A struct containing the details of a single obesrvation.

### 13.15.2 Member Data Documentation

#### 13.15.2.1 addedNoise

```
double observation::addedNoise
```

#### 13.15.2.2 averagebat

```
double observation::averagebat
```

#### 13.15.2.3 averagedmbat

```
double observation::averagedmbat
```

#### 13.15.2.4 averagedmerr

```
double observation::averagedmerr
```

#### 13.15.2.5 averagedmres

```
double observation::averagedmres
```

**13.15.2.6 averageerr**

`double observation::averageerr`

**13.15.2.7 averageres**

`double observation::averageres`

**13.15.2.8 bat**

`longdouble observation::bat`

Infinite frequency barycentric arrival time

**13.15.2.9 batCorr**

`longdouble observation::batCorr`

**13.15.2.10 bbat**

`longdouble observation::bbat`

Arrival time at binary barycentre

**13.15.2.11 bline**

`double observation::bline`

**13.15.2.12 chisq**

`double observation::chisq`

**13.15.2.13 clockCorr**

`int observation::clockCorr`

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

**13.15.2.14 correctionsTT**

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

chain of corrections from site TOA to chosen realisation of TT

**13.15.2.15 correctionTT\_calcEph**

```
longdouble observation::correctionTT_calcEph
```

**13.15.2.16 correctionTT\_TB**

```
longdouble observation::correctionTT_TB
```

Correction to TDB/TCB

**13.15.2.17 correctionTT\_Teph**

```
longdouble observation::correctionTT_Teph
```

Correction to Teph

**13.15.2.18 correctionUT1**

```
longdouble observation::correctionUT1
```

Correction from site TOA to UT1

**13.15.2.19 delayCorr**

```
int observation::delayCorr
```

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

**13.15.2.20 deleted**

```
int observation::deleted
```

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

**13.15.2.21 earth\_ssb**

```
double observation::earth_ssb[6]
```

Centre of Earth w.r.t. SSB

**13.15.2.22 earthMoonBary\_earth**

```
double observation::earthMoonBary_earth[6]
```

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

**13.15.2.23 earthMoonBary\_ssb**

```
double observation::earthMoonBary_ssb[6]
```

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

**13.15.2.24 efac**

```
double observation::efac
```

Error multiplication factor

**13.15.2.25 einsteinRate**

```
double observation::einsteinRate
```

Derivative of correctionTT\_TB

**13.15.2.26 equad**

```
double observation::equad
```

Value to add in quadrature

**13.15.2.27 flagID**

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

Flags in .tim file

**13.15.2.28 flagVal**

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

**13.15.2.29 fname**

```
char observation::fname[MAX_FILELEN]
```

Name of data file giving TOA

**13.15.2.30 freq**

```
double observation::freq
```

Frequency of observation (in MHz)

**13.15.2.31 freqSSB**

```
double observation::freqSSB
```

Frequency of observation in barycentric frame (in Hz)

**13.15.2.32 jump**

```
int observation::jump[MAX_FLAGS]
```

Jump region

**13.15.2.33 jupiter\_earth**

```
double observation::jupiter_earth[6]
```

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

**13.15.2.34 nclock\_correction**

```
int observation::nclock_correction
```

**13.15.2.35 neptune\_earth**

```
double observation::neptune_earth[6]
```

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

**13.15.2.36 nFlags**

```
int observation::nFlags
```

**13.15.2.37 nphase**

```
longdouble observation::nphase
```

allows the pulse number to be determined

**13.15.2.38 nutations**

```
double observation::nutations[6]
```

**13.15.2.39 observatory\_earth**

```
double observation::observatory_earth[6]
```

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

**13.15.2.40 obsNjump**

```
int observation::obsNjump
```

Number of jumps for this observation

**13.15.2.41 origErr**

```
double observation::origErr
```

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

**13.15.2.42 origsat**

```
longdouble observation::origsat
```

**13.15.2.43 pet**

```
longdouble observation::pet
```

Pulsar emission time

**13.15.2.44 phase**

```
longdouble observation::phase
```

**13.15.2.45 phaseOffset**

```
double observation::phaseOffset
```

Phase offset

**13.15.2.46 planet\_ssb**

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

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

**13.15.2.47 planet\_ssb\_derv**

```
double observation::planet_ssb_derv[9][6]
```

**13.15.2.48 planet\_ssb\_tmr**

```
double observation::planet_ssb_tmr[9][6]
```

**13.15.2.49 pnoise**

```
double observation::pnoise
```

**13.15.2.50 prefitResidual**

```
longdouble observation::prefitResidual
```

Pre-fit residual

**13.15.2.51 psrPos**

```
double observation::psrPos[3]
```

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

**13.15.2.52 pulseN**

```
long long observation::pulseN
```

Pulse number

**13.15.2.53 residual**

```
longdouble observation::residual
```

residual

**13.15.2.54 residualtn**

`longdouble observation::residualtn`

**13.15.2.55 roemer**

`longdouble observation::roemer`

Roemer delay

**13.15.2.56 sat**

`longdouble observation::sat`

Site arrival time

**13.15.2.57 sat\_day**

`longdouble observation::sat_day`

**13.15.2.58 sat\_sec**

`longdouble observation::sat_sec`

**13.15.2.59 saturn\_earth**

`double observation::saturn_earth[6]`

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

**13.15.2.60 shapiroDelayJupiter**

`double observation::shapiroDelayJupiter`

Shapiro Delay due to Jupiter

**13.15.2.61 shapiroDelayNeptune**

`double observation::shapiroDelayNeptune`

Shapiro Delay due to Neptune



**13.15.2.62 shapiroDelaySaturn**

```
double observation::shapiroDelaySaturn
```

Shapiro Delay due to Saturn

**13.15.2.63 shapiroDelaySun**

```
double observation::shapiroDelaySun
```

Shapiro Delay due to the Sun

**13.15.2.64 shapiroDelayUranus**

```
double observation::shapiroDelayUranus
```

Shapiro Delay due to Uranus

**13.15.2.65 shapiroDelayVenus**

```
double observation::shapiroDelayVenus
```

Shapiro Delay due to Venus

**13.15.2.66 shklovskii**

```
longdouble observation::shklovskii
```

Shklovskii delay term

**13.15.2.67 siteVel**

```
double observation::siteVel[3]
```

Observatory velocity w.r.t. geocentre

**13.15.2.68 snr**

```
double observation::snr
```

**13.15.2.69 sun\_earth**

```
double observation::sun_earth[6]
```

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

**13.15.2.70 sun\_ssb**

```
double observation::sun_ssb[6]
```

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

**13.15.2.71 tdis1**

```
double observation::tdis1
```

Interstellar dispersion measure delay

**13.15.2.72 tdis2**

```
double observation::tdis2
```

Dispersion measure delay due to solar system

**13.15.2.73 telID**

```
char observation::telID[100]
```

Telescope ID

**13.15.2.74 TNDMErr**

```
double observation::TNDMErr
```

Error on Model DM signal from temponest fit

**13.15.2.75 TNDMSignal**

```
double observation::TNDMSignal
```

Model DM signal from temponest fit

**13.15.2.76 TNGroupErr**

```
double observation::TNGroupErr
```

Error on Model Group Noise signal from temponest fit

**13.15.2.77 TNGroupSignal**

```
double observation::TNGroupSignal
```

Model Group Noise signal from temponest fit

**13.15.2.78 TNRedErr**

```
double observation::TNRedErr
```

Error on Model red noise signal from temponest fit

**13.15.2.79 TNRedSignal**

```
double observation::TNRedSignal
```

Model red noise signal from temponest fit

**13.15.2.80 toaDMErr**

```
double observation::toaDMErr
```

Error on TOA due to DM (in us)

**13.15.2.81 toaErr**

```
double observation::toaErr
```

Error on TOA (in us)

**13.15.2.82 tobs**

```
double observation::tobs
```

**13.15.2.83 torb**

```
longdouble observation::torb
```

Combined binary delays

**13.15.2.84 troposphericDelay**

```
double observation::troposphericDelay
```

Delay due to neutral refraction in atmosphere

**13.15.2.85 uranus\_earth**

```
double observation::uranus_earth[6]
```

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

### 13.15.2.86 venus\_earth

```
double observation::venus_earth[6]
```

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

### 13.15.2.87 zenith

```
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](#)

## 13.16 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]

### 13.16.1 Member Data Documentation

#### 13.16.1.1 clock\_name

```
char observatory::clock_name[16]
```

#### 13.16.1.2 code

```
char observatory::code[16]
```

### 13.16.1.3 height\_grs80

```
double observatory::height_grs80
```

### 13.16.1.4 latitude\_grs80

```
double observatory::latitude_grs80
```

### 13.16.1.5 longitude\_grs80

```
double observatory::longitude_grs80
```

### 13.16.1.6 name

```
char observatory::name[32]
```

### 13.16.1.7 x

```
double observatory::x
```

### 13.16.1.8 y

```
double observatory::y
```

### 13.16.1.9 z

```
double observatory::z
```

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

- [tempo2.h](#)

## 13.17 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](#)

### 13.17.1 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

### 13.17.2 Member Data Documentation

#### 13.17.2.1 aSize

```
int parameter::aSize
```

Number of elements in the array for this parameter

#### 13.17.2.2 err

```
longdouble* parameter::err
```

Uncertainty on parameter value

### 13.17.2.3 fitFlag

```
int* parameter::fitFlag
```

= 1 if fitting required, = 2 for global fit

### 13.17.2.4 label

```
char** parameter::label
```

Label about this parameter

### 13.17.2.5 linkFrom

```
int parameter::linkFrom[5]
```

### 13.17.2.6 linkTo

```
int parameter::linkTo[5]
```

### 13.17.2.7 nLinkFrom

```
int parameter::nLinkFrom
```

### 13.17.2.8 nLinkTo

```
int parameter::nLinkTo
```

### 13.17.2.9 paramSet

```
int* parameter::paramSet
```

= 1 if parameter has been set

### 13.17.2.10 prefit

```
longdouble* parameter::prefit
```

Pre-fit value of the parameter

### 13.17.2.11 prefitErr

`longdouble* parameter::prefitErr`

Pre-fit value of the uncertainty

### 13.17.2.12 shortlabel

`char** parameter::shortlabel`

Label about this parameter without units

### 13.17.2.13 val

`longdouble* parameter::val`

Value of parameter

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

- [tempo2.h](#)

## 13.18 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 [dmoftsDMnum](#)
- int [dmoftsCMnum](#)
- double [dmoftsDM\\_mjd](#) [[MAX\\_IFUNC](#)]
- double [dmoftsDM](#) [[MAX\\_IFUNC](#)]
- double [dmoftsDM\\_error](#) [[MAX\\_IFUNC](#)]
- double [dmoftsDM\\_weight](#) [[MAX\\_IFUNC](#)]
- double [dmoftsCM\\_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 [gwcs\\_raj](#)
- double [gwcs\\_decj](#)
- double [gwcs\\_epoch](#)
- double [gwcs\\_width](#)
- double [gwcs\\_geom\\_p](#)
- double [gwcs\\_geom\\_c](#)
- 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]
- char [jumpSAT](#) [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](#)
- int [nSx](#)
- double [fitChisq](#)
- int [fitNfree](#)
- int [globalNfit](#)
- int [globalNoConstrain](#)
- int [nFit](#)
- int [nParam](#)
- int [nGlobal](#)
- 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](#)
- double [rmstn](#)
- 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 [TNGlobalIEF](#)
- char [TNEQFlagID](#) [MAX\_TNEQ][MAX\_FLAG\_LEN]
- char [TNEQFlagVal](#) [MAX\_TNEQ][MAX\_FLAG\_LEN]
- double [TNEQVal](#) [MAX\_TNEQ]
- double [TNGlobalIEQ](#)
- 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](#)
- double [detUinv](#)
- 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](#)
- double [constraint\\_efactor](#)
- enum [constraint constraints](#) [MAX\_PARAMS]
- char [auto\\_constraints](#)
- char \* [constraint\\_special](#) [MAX\_PARAMS]
- [FitInfo](#) [fitinfo](#)
- int [brace](#)

### 13.18.1 Detailed Description

contains the details for a single pulsar.

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

### 13.18.2 Member Data Documentation

#### 13.18.2.1 addTNGlobalEQ

```
double pulsar::addTNGlobalEQ
```

#### 13.18.2.2 auto\_constraints

```
char pulsar::auto_constraints
```

#### 13.18.2.3 AverageDMResiduals

```
int pulsar::AverageDMResiduals
```

#### 13.18.2.4 AverageEpochWidth

```
float pulsar::AverageEpochWidth
```

#### 13.18.2.5 AverageFlag

```
char pulsar::AverageFlag[MAX\_FLAG\_LEN]
```

#### 13.18.2.6 AverageResiduals

```
int pulsar::AverageResiduals
```

**13.18.2.7 binaryModel**

```
char pulsar::binaryModel[100]
```

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

**13.18.2.8 bootStrap**

```
int pulsar::bootStrap
```

0 if calculating errors using bootstrap Monte-Carlo method

**13.18.2.9 brace**

```
int pulsar::brace
```

**13.18.2.10 calcShapiro**

```
int pulsar::calcShapiro
```

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

**13.18.2.11 cgw\_angpol**

```
double pulsar::cgw_angpol
```

**13.18.2.12 cgw\_cosinc**

```
double pulsar::cgw_cosinc
```

**13.18.2.13 cgw\_h0**

```
double pulsar::cgw_h0
```

#### 13.18.2.14 cgw\_mc

```
double pulsar::cgw_mc
```

#### 13.18.2.15 clk\_offsE

```
double pulsar::clk_offsE[MAX_TEL_CLK_OFFS]
```

#### 13.18.2.16 clk\_offsT

```
double pulsar::clk_offsT[MAX_TEL_CLK_OFFS]
```

#### 13.18.2.17 clk\_offsV

```
double pulsar::clk_offsV[MAX_TEL_CLK_OFFS]
```

#### 13.18.2.18 clkOffsN

```
int pulsar::clkOffsN
```

#### 13.18.2.19 clock

```
char pulsar::clock[16]
```

Clock standard to use as "UTC"

#### 13.18.2.20 clockFromOverride

```
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)



**13.18.2.21 constraint\_efactor**

```
double pulsar::constraint_efactor
```

**13.18.2.22 constraint\_special**

```
char* pulsar::constraint_special[MAX\_PARAMS]
```

**13.18.2.23 constraints**

```
enum constraint pulsar::constraints[MAX\_PARAMS]
```

Which constraints are specified

**13.18.2.24 correctTroposphere**

```
int pulsar::correctTroposphere
```

whether or not do correct for tropospheric delay

**13.18.2.25 covar**

```
double** pulsar::covar
```

**13.18.2.26 decjStrPost**

```
char pulsar::decjStrPost[100]
```

String containing RAJ and DECJ (postfit)

**13.18.2.27 decjStrPre**

```
char pulsar::decjStrPre[100]
```

String containing RAJ and DECJ (prefit)

**13.18.2.28 decsim**

```
double pulsar::decsim
```

**13.18.2.29 deleteFileName**

```
char pulsar::deleteFileName[100]
```

File name containing deleted points

**13.18.2.30 detUinv**

```
double pulsar::detUinv
```

**13.18.2.31 dilateFreq**

```
int pulsar::dilateFreq
```

whether or not to apply SS time dilation to RFs

**13.18.2.32 dmooffsCM**

```
double pulsar::dmooffsCM[MAX\_IFUNC]
```

**13.18.2.33 dmooffsCM\_error**

```
double pulsar::dmooffsCM_error[MAX\_IFUNC]
```

**13.18.2.34 dmooffsCM\_mjd**

```
double pulsar::dmooffsCM_mjd[MAX\_IFUNC]
```

**13.18.2.35 dmooffsCM\_weight**

```
double pulsar::dmooffsCM_weight[MAX\_IFUNC]
```

**13.18.2.36 dmooffsCMnum**

```
int pulsar::dmooffsCMnum
```

**13.18.2.37 dmoffsDM**

```
double pulsar::dmoffsDM[MAX_IFUNC]
```

**13.18.2.38 dmoffsDM\_error**

```
double pulsar::dmoffsDM_error[MAX_IFUNC]
```

**13.18.2.39 dmoffsDM\_mjd**

```
double pulsar::dmoffsDM_mjd[MAX_IFUNC]
```

**13.18.2.40 dmoffsDM\_weight**

```
double pulsar::dmoffsDM_weight[MAX_IFUNC]
```

**13.18.2.41 dmoffsDMnum**

```
int pulsar::dmoffsDMnum
```

**13.18.2.42 dmOffset**

```
double pulsar::dmOffset
```

Value to add to DM flags

**13.18.2.43 eclCoord**

```
int pulsar::eclCoord
```

= 1 for ecliptic coords otherwise celestial coords

**13.18.2.44 eopc04\_file**

```
char pulsar::eopc04_file[MAX_FILELEN]
```

**13.18.2.45 ephemeris**

```
char pulsar::ephemeris[MAX_FILELEN]
```

**13.18.2.46 filterStr**

```
char pulsar::filterStr[MAX_STRLEN]
```

String describing filters

**13.18.2.47 fitChisq**

```
double pulsar::fitChisq
```

Chisq value from the fit

**13.18.2.48 fitFunc**

```
char pulsar::fitFunc[MAX_FILELEN]
```

**13.18.2.49 fitinfo**

```
FitInfo pulsar::fitinfo
```

**13.18.2.50 fitJump**

```
int pulsar::fitJump[MAX_JUMPS]
```

= 1 if fit for jump

**13.18.2.51 fitMode**

```
int pulsar::fitMode
```

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

**13.18.2.52 fitNfree**

```
int pulsar::fitNfree
```

Number of degrees of freedom in fit

**13.18.2.53 fixedFormat**

```
int pulsar::fixedFormat
```

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

**13.18.2.54 fjumpID**

```
char pulsar::fjumpID[16]
```

**13.18.2.55 globalNfit**

```
int pulsar::globalNfit
```

Total number of parameters in the fit

**13.18.2.56 globalNoConstrain**

```
int pulsar::globalNoConstrain
```

Total number of points without constraints

**13.18.2.57 gwb\_decj**

```
double pulsar::gwb_decj
```

**13.18.2.58 gwb\_epoch**

```
double pulsar::gwb_epoch
```

**13.18.2.59 gwb\_geom\_c**

```
double pulsar::gwb_geom_c
```

**13.18.2.60 gwb\_geom\_p**

```
double pulsar::gwb_geom_p
```

**13.18.2.61 gwb\_raj**

```
double pulsar::gwb_raj
```

**13.18.2.62 gwb\_width**

```
double pulsar::gwb_width
```

**13.18.2.63 gwcs\_decj**

```
double pulsar::gwcs_decj
```

**13.18.2.64 gwcs\_epoch**

```
double pulsar::gwcs_epoch
```

**13.18.2.65 gwcs\_geom\_c**

```
double pulsar::gwcs_geom_c
```

**13.18.2.66 gwcs\_geom\_p**

```
double pulsar::gwcs_geom_p
```

**13.18.2.67 gwcs\_raj**

```
double pulsar::gwcs_raj
```

**13.18.2.68 gwcs\_width**

```
double pulsar::gwcs_width
```

**13.18.2.69 gwecc\_dec**

```
double pulsar::gwecc_dec
```

**13.18.2.70 gwecc\_distance**

```
double pulsar::gwecc_distance
```

**13.18.2.71 gwecc\_e**

```
double pulsar::gwecc_e
```

**13.18.2.72 gwecc\_epoch**

```
double pulsar::gwecc_epoch
```

**13.18.2.73 gwecc\_inc**

```
double pulsar::gwecc_inc
```

**13.18.2.74 gwecc\_m1**

```
double pulsar::gwecc_m1
```

**13.18.2.75 gwecc\_m2**

```
double pulsar::gwecc_m2
```

**13.18.2.76 gwecc\_nodes\_orientation**

```
double pulsar::gwecc_nodes_orientation
```

**13.18.2.77 gwecc\_orbital\_period**

```
double pulsar::gwecc_orbital_period
```

**13.18.2.78 gwecc\_psrdist**

```
double pulsar::gwecc_psrdist
```

**13.18.2.79 gwecc\_pulsarTermOn**

```
int pulsar::gwecc_pulsarTermOn
```

**13.18.2.80 gwecc\_ra**

```
double pulsar::gwecc_ra
```

**13.18.2.81 gwecc\_redshift**

```
double pulsar::gwecc_redshift
```

**13.18.2.82 gwecc\_theta\_0**

```
double pulsar::gwecc_theta_0
```

**13.18.2.83 gwecc\_theta\_nodes**

```
double pulsar::gwecc_theta_nodes
```

**13.18.2.84 gwm\_decj**

```
double pulsar::gwm_decj
```



**13.18.2.85 gwm\_dphase**

```
double pulsar::gwm_dphase
```

**13.18.2.86 gwm\_epoch**

```
double pulsar::gwm_epoch
```

**13.18.2.87 gwm\_phi**

```
double pulsar::gwm_phi
```

**13.18.2.88 gwm\_raj**

```
double pulsar::gwm_raj
```

**13.18.2.89 gwsrc\_across\_i**

```
double pulsar::gwsrc_across_i
```

**13.18.2.90 gwsrc\_across\_i\_e**

```
double pulsar::gwsrc_across_i_e
```

**13.18.2.91 gwsrc\_across\_r**

```
double pulsar::gwsrc_across_r
```

**13.18.2.92 gwsrc\_across\_r\_e**

```
double pulsar::gwsrc_across_r_e
```

**13.18.2.93 gwsrc\_aplus\_i**

```
double pulsar::gwsrc_aplus_i
```

**13.18.2.94 gwsrc\_aplus\_i\_e**

```
double pulsar::gwsrc_aplus_i_e
```

**13.18.2.95 gwsrc\_aplus\_r**

```
double pulsar::gwsrc_aplus_r
```

**13.18.2.96 gwsrc\_aplus\_r\_e**

```
double pulsar::gwsrc_aplus_r_e
```

**13.18.2.97 gwsrc\_dec**

```
double pulsar::gwsrc_dec
```

**13.18.2.98 gwsrc\_epoch**

```
double pulsar::gwsrc_epoch
```

**13.18.2.99 gwsrc\_psrdist**

```
double pulsar::gwsrc_psrdist
```

**13.18.2.100 gwsrc\_ra**

```
double pulsar::gwsrc_ra
```

**13.18.2.101 ifunc\_weights**

```
double pulsar::ifunc_weights[MAX\_IFUNC]
```

**13.18.2.102 ifuncE**

```
double pulsar::ifuncE[MAX\_IFUNC]
```

**13.18.2.103 ifuncN**

```
int pulsar::ifuncN
```

**13.18.2.104 ifuncT**

```
double pulsar::ifuncT[MAX\_IFUNC]
```

**13.18.2.105 ifuncV**

```
double pulsar::ifuncV[MAX\_IFUNC]
```

**13.18.2.106 ipm**

```
int pulsar::ipm
```

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

**13.18.2.107 jboFormat**

```
int pulsar::jboFormat
```

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

**13.18.2.108 JPL\_EPHEMERIS**

```
char pulsar::JPL_EPHEMERIS[MAX\_FILELEN]
```

**13.18.2.109 jumpSAT**

```
char pulsar::jumpSAT[MAX_JUMPS]
```

This jump is in SAT rather than phase

**13.18.2.110 jumpStr**

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

String describing jump

**13.18.2.111 jumpVal**

```
double pulsar::jumpVal[MAX_JUMPS]
```

Value of jump

**13.18.2.112 jumpValErr**

```
double pulsar::jumpValErr[MAX_JUMPS]
```

Error on jump

**13.18.2.113 name**

```
char pulsar::name[100]
```

**13.18.2.114 nCompanion**

```
int pulsar::nCompanion
```

Number of binary companions

**13.18.2.115 nconstraints**

```
int pulsar::nconstraints
```

Number of fit constraints specified

**13.18.2.116 nDMEvents**

```
int pulsar::nDMEvents
```

**13.18.2.117 ndmx**

```
int pulsar::ndmx
```

Number of DM steps

**13.18.2.118 ne\_sw**

```
double pulsar::ne_sw
```

Electron density at 1AU due to the solar wind

**13.18.2.119 nFit**

```
int pulsar::nFit
```

Number of points in the fit

**13.18.2.120 nGlobal**

```
int pulsar::nGlobal
```

Number of global parameters in the fit

**13.18.2.121 nits**

```
int pulsar::nits
```

Number of iterations for the fit

**13.18.2.122 nJumps**

```
int pulsar::nJumps
```

Number of jumps

**13.18.2.123 nobs**

```
int pulsar::nobs
```

Number of observations in .tim file

**13.18.2.124 noWarnings**

```
int pulsar::noWarnings
```

= 1, do not display warning messages

**13.18.2.125 nParam**

```
int pulsar::nParam
```

Number of parameters in the fit

**13.18.2.126 nPhaseJump**

```
int pulsar::nPhaseJump
```

Number of phase jumps

**13.18.2.127 nQuad**

```
int pulsar::nQuad
```

**13.18.2.128 nStorePrecision**

```
int pulsar::nStorePrecision
```

**13.18.2.129 nSx**

```
int pulsar::nSx
```

Number of Scatter steps

**13.18.2.130 nT2efac**

```
int pulsar::nT2efac
```

**13.18.2.131 nT2equad**

```
int pulsar::nT2equad
```

**13.18.2.132 nTelDX**

```
int pulsar::nTelDX
```

**13.18.2.133 nTelDY**

```
int pulsar::nTelDY
```

**13.18.2.134 nTelDZ**

```
int pulsar::nTelDZ
```

**13.18.2.135 nTNBandNoise**

```
int pulsar::nTNBandNoise
```

**13.18.2.136 nTNECORR**

```
int pulsar::nTNECORR
```

**13.18.2.137 nTNEF**

```
int pulsar::nTNEF
```

**13.18.2.138 nTNEQ**

```
int pulsar::nTNEQ
```

**13.18.2.139 nTNGroupNoise**

```
int pulsar::nTNGroupNoise
```

**13.18.2.140 nTNShapeletEvents**

```
int pulsar::nTNShapeletEvents
```

**13.18.2.141 nTNSQ**

```
int pulsar::nTNSQ
```

**13.18.2.142 nToffset**

```
int pulsar::nToffset
```

**13.18.2.143 nWhite**

```
int pulsar::nWhite
```

**13.18.2.144 nWhite\_dm**

```
int pulsar::nWhite_dm
```

**13.18.2.145 obsn**

```
observation* pulsar::obsn
```

```
[MAX_OBSN_VAL];
```

**13.18.2.146 offset**

```
double pulsar::offset
```

Offset, always fitted for

**13.18.2.147 offset\_e**

```
double pulsar::offset_e
```

Error in the offset

**13.18.2.148 outputTMatrix**

```
int pulsar::outputTMatrix
```



**13.18.2.149 param**

```
parameter pulsar::param[MAX\_PARAMS]
```

**13.18.2.150 passStr**

```
char pulsar::passStr[MAX\_STRLEN]
```

String describing filters

**13.18.2.151 phaseJump**

```
longdouble pulsar::phaseJump[MAX\_JUMPS]
```

Time of phase jump

**13.18.2.152 phaseJumpDir**

```
int pulsar::phaseJumpDir[MAX\_JUMPS]
```

Size and direction of phase jump

**13.18.2.153 phaseJumpID**

```
int pulsar::phaseJumpID[MAX\_JUMPS]
```

ID of closest point to the phase jump

**13.18.2.154 planetShapiro**

```
int pulsar::planetShapiro
```

= 1 if included otherwise 0

**13.18.2.155 posPulsar**

```
double pulsar::posPulsar[3]
```

3-vector pointing at pulsar

**13.18.2.156 quad\_across\_i**

```
double pulsar::quad_across_i[MAX\_QUAD]
```

**13.18.2.157 quad\_across\_i\_e**

```
double pulsar::quad_across_i_e[MAX\_QUAD]
```

**13.18.2.158 quad\_across\_r**

```
double pulsar::quad_across_r[MAX\_QUAD]
```

**13.18.2.159 quad\_across\_r\_e**

```
double pulsar::quad_across_r_e[MAX\_QUAD]
```

**13.18.2.160 quad\_aplus\_i**

```
double pulsar::quad_aplus_i[MAX\_QUAD]
```

**13.18.2.161 quad\_aplus\_i\_e**

```
double pulsar::quad_aplus_i_e[MAX\_QUAD]
```

**13.18.2.162 quad\_aplus\_r**

```
double pulsar::quad_aplus_r[MAX\_QUAD]
```

**13.18.2.163 quad\_aplus\_r\_e**

```
double pulsar::quad_aplus_r_e[MAX\_QUAD]
```

**13.18.2.164 quad\_ifunc\_c\_DEC**

```
double pulsar::quad_ifunc_c_DEC
```

**13.18.2.165 quad\_ifunc\_c\_RA**

```
double pulsar::quad_ifunc_c_RA
```

**13.18.2.166 quad\_ifunc\_geom\_c**

```
double pulsar::quad_ifunc_geom_c
```

**13.18.2.167 quad\_ifunc\_geom\_p**

```
double pulsar::quad_ifunc_geom_p
```

**13.18.2.168 quad\_ifunc\_p\_DEC**

```
double pulsar::quad_ifunc_p_DEC
```

**13.18.2.169 quad\_ifunc\_p\_RA**

```
double pulsar::quad_ifunc_p_RA
```

**13.18.2.170 quad\_ifuncE\_c**

```
double pulsar::quad_ifuncE_c[MAX\_IFUNC]
```

**13.18.2.171 quad\_ifuncE\_p**

```
double pulsar::quad_ifuncE_p[MAX\_IFUNC]
```

**13.18.2.172 quad\_ifuncN\_c**

```
int pulsar::quad_ifuncN_c
```

**13.18.2.173 quad\_ifuncN\_p**

```
int pulsar::quad_ifuncN_p
```

**13.18.2.174 quad\_ifuncT\_c**

```
double pulsar::quad_ifuncT_c[MAX\_IFUNC]
```

**13.18.2.175 quad\_ifuncT\_p**

```
double pulsar::quad_ifuncT_p[MAX\_IFUNC]
```

**13.18.2.176 quad\_ifuncV\_c**

```
double pulsar::quad_ifuncV_c[MAX\_IFUNC]
```

**13.18.2.177 quad\_ifuncV\_p**

```
double pulsar::quad_ifuncV_p[MAX\_IFUNC]
```

**13.18.2.178 quadDEC**

```
double pulsar::quadDEC
```

**13.18.2.179 quadEpoch**

```
double pulsar::quadEpoch
```

**13.18.2.180 quadRA**

```
double pulsar::quadRA
```

**13.18.2.181 rajStrPost**

```
char pulsar::rajStrPost[100]
```

**13.18.2.182 rajStrPre**

```
char pulsar::rajStrPre[100]
```

**13.18.2.183 rasim**

```
double pulsar::rasim
```

**13.18.2.184 rescaleErrChisq**

```
int pulsar::rescaleErrChisq
```

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

**13.18.2.185 rmsPost**

```
double pulsar::rmsPost
```

**13.18.2.186 rmsPre**

```
double pulsar::rmsPre
```

**13.18.2.187 rmstn**

```
double pulsar::rmstn
```

**13.18.2.188 robust**

```
char pulsar::robust
```

**13.18.2.189 setTelVelX**

```
int pulsar::setTelVelX
```

**13.18.2.190 setTelVelY**

```
int pulsar::setTelVelY
```

**13.18.2.191 setTelVelZ**

```
int pulsar::setTelVelZ
```

**13.18.2.192 setUnits**

```
int pulsar::setUnits
```

**13.18.2.193 simflag**

```
int pulsar::simflag
```

Which fit function are we using

**13.18.2.194 sorted**

```
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];

**13.18.2.195 storePrec**

```
storePrecision pulsar::storePrec[MAX_STOREPRECISION]
```

**13.18.2.196 swm**

```
int pulsar::swm
```

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

**13.18.2.197 t2cMethod**

```
int pulsar::t2cMethod
```

How to transform from terrestrial to celestial coords

**13.18.2.198 T2efacFlagID**

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

**13.18.2.199 T2efacFlagVal**

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

**13.18.2.200 T2efacVal**

```
double pulsar::T2efacVal[MAX_T2EFAC]
```

**13.18.2.201 T2equadFlagID**

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

**13.18.2.202 T2equadFlagVal**

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

**13.18.2.203 T2equadVal**

```
double pulsar::T2equadVal[MAX_T2EQUAD]
```

**13.18.2.204 T2globalEfac**

```
double pulsar::T2globalEfac
```

**13.18.2.205 telDX\_e**

```
double pulsar::telDX_e[MAX\_TEL\_DX]
```

**13.18.2.206 telDX\_t**

```
double pulsar::telDX_t[MAX\_TEL\_DX]
```

**13.18.2.207 telDX\_v**

```
double pulsar::telDX_v[MAX\_TEL\_DX]
```

**13.18.2.208 telDX\_vel**

```
double pulsar::telDX_vel[MAX\_TEL\_DX]
```

**13.18.2.209 telDX\_vel\_e**

```
double pulsar::telDX_vel_e[MAX\_TEL\_DX]
```

**13.18.2.210 telDY\_e**

```
double pulsar::telDY_e[MAX\_TEL\_DY]
```

**13.18.2.211 telDY\_t**

```
double pulsar::telDY_t[MAX\_TEL\_DY]
```



**13.18.2.212 telDY\_v**

```
double pulsar::telDY_v[MAX\_TEL\_DY]
```

**13.18.2.213 telDY\_vel**

```
double pulsar::telDY_vel[MAX\_TEL\_DY]
```

**13.18.2.214 telDY\_vel\_e**

```
double pulsar::telDY_vel_e[MAX\_TEL\_DY]
```

**13.18.2.215 telDZ\_e**

```
double pulsar::telDZ_e[MAX\_TEL\_DZ]
```

**13.18.2.216 telDZ\_t**

```
double pulsar::telDZ_t[MAX\_TEL\_DZ]
```

**13.18.2.217 telDZ\_v**

```
double pulsar::telDZ_v[MAX\_TEL\_DZ]
```

**13.18.2.218 telDZ\_vel**

```
double pulsar::telDZ_vel[MAX\_TEL\_DZ]
```

**13.18.2.219 telDZ\_vel\_e**

```
double pulsar::telDZ_vel_e[MAX\_TEL\_DZ]
```

**13.18.2.220 tempo1**

```
int pulsar::tempo1
```

= 1 if tempo1 is emulated

**13.18.2.221 timeEphemeris**

```
int pulsar::timeEphemeris
```

Which code to use for Einstein delay

**13.18.2.222 TNBandDMAmp**

```
double pulsar::TNBandDMAmp
```

**13.18.2.223 TNBandDMC**

```
int pulsar::TNBandDMC
```

**13.18.2.224 TNBandDMGam**

```
double pulsar::TNBandDMGam
```

**13.18.2.225 TNBandNoiseAmp**

```
double pulsar::TNBandNoiseAmp[MAX\_TBNB]
```

**13.18.2.226 TNBandNoiseC**

```
int pulsar::TNBandNoiseC[MAX\_TBNB]
```

**13.18.2.227 TNBandNoiseGam**

```
double pulsar::TNBandNoiseGam[MAX\_TBNB]
```

**13.18.2.228 TNBandNoiseHF**

```
double pulsar::TNBandNoiseHF[MAX\_TBN]
```

**13.18.2.229 TNBandNoiseLF**

```
double pulsar::TNBandNoiseLF[MAX\_TBN]
```

**13.18.2.230 TNDMAmp**

```
double pulsar::TNDMAmp
```

**13.18.2.231 TNDMC**

```
int pulsar::TNDMC
```

**13.18.2.232 TNDMCoeffs**

```
double pulsar::TNDMCoeffs[200]
```

**13.18.2.233 TNDMEvAmp**

```
double pulsar::TNDMEvAmp[MAX\_TNDMEv]
```

**13.18.2.234 TNDMEvGam**

```
double pulsar::TNDMEvGam[MAX\_TNDMEv]
```

**13.18.2.235 TNDMEvLength**

```
double pulsar::TNDMEvLength[MAX\_TNDMEv]
```

**13.18.2.236 TNDMEvLin**

```
int pulsar::TNDMEvLin[MAX\_TNDMEv]
```

**13.18.2.237 TNDMEvOff**

```
int pulsar::TNDMEvOff[MAX\_TNDMEv]
```

**13.18.2.238 TNDMEvQuad**

```
int pulsar::TNDMEvQuad[MAX\_TNDMEv]
```

**13.18.2.239 TNDMEvStart**

```
double pulsar::TNDMEvStart[MAX\_TNDMEv]
```

**13.18.2.240 TNDMGam**

```
double pulsar::TNDMGam
```

**13.18.2.241 TNECORRFlagID**

```
char pulsar::TNECORRFlagID[MAX\_TNECORR][MAX\_FLAG\_LEN]
```

**13.18.2.242 TNECORRFlagVal**

```
char pulsar::TNECORRFlagVal[MAX\_TNECORR][MAX\_FLAG\_LEN]
```

**13.18.2.243 TNECORRVal**

```
double pulsar::TNECORRVal[MAX\_TNECORR]
```

**13.18.2.244 TNEFFlagID**

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

**13.18.2.245 TNEFFlagVal**

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

**13.18.2.246 TNEFVal**

```
double pulsar::TNEFVal[MAX_TNEF]
```

**13.18.2.247 TNEQFlagID**

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

**13.18.2.248 TNEQFlagVal**

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

**13.18.2.249 TNEQVal**

```
double pulsar::TNEQVal[MAX_TNEQ]
```

**13.18.2.250 TNGlobalEF**

```
double pulsar::TNGlobalEF
```

**13.18.2.251 TNGlobalEQ**

```
double pulsar::TNGlobalEQ
```

**13.18.2.252 TNGroupNoiseAmp**

```
double pulsar::TNGroupNoiseAmp[MAX\_TNGN]
```

**13.18.2.253 TNGroupNoiseC**

```
int pulsar::TNGroupNoiseC[MAX\_TNGN]
```

**13.18.2.254 TNGroupNoiseFlagID**

```
char pulsar::TNGroupNoiseFlagID[MAX\_TNGN][MAX\_FLAG\_LEN]
```

**13.18.2.255 TNGroupNoiseFlagVal**

```
char pulsar::TNGroupNoiseFlagVal[MAX\_TNGN][MAX\_FLAG\_LEN]
```

**13.18.2.256 TNGroupNoiseGam**

```
double pulsar::TNGroupNoiseGam[MAX\_TNGN]
```

**13.18.2.257 TNRedAmp**

```
double pulsar::TNRedAmp
```

**13.18.2.258 TNRedC**

```
int pulsar::TNRedC
```

**13.18.2.259 TNRedCoeffs**

```
double pulsar::TNRedCoeffs[200]
```

**13.18.2.260 TNRedCorner**

```
double pulsar::TNRedCorner
```

**13.18.2.261 TNRedFlow**

```
double pulsar::TNRedFlow
```

**13.18.2.262 TNRedGam**

```
double pulsar::TNRedGam
```

**13.18.2.263 TNShapeletEvFScale**

```
double pulsar::TNShapeletEvFScale[MAX\_TNDMEv]
```

**13.18.2.264 TNShapeletEvN**

```
int pulsar::TNShapeletEvN[MAX\_TNDMEv]
```

**13.18.2.265 TNShapeletEvPos**

```
double pulsar::TNShapeletEvPos[MAX\_TNDMEv]
```

**13.18.2.266 TNShapeletEvWidth**

```
double pulsar::TNShapeletEvWidth[MAX\_TNDMEv]
```

**13.18.2.267 TNSQFlagID**

```
char pulsar::TNSQFlagID[MAX\_TNSQ][MAX\_FLAG\_LEN]
```

**13.18.2.268 TNSQFlagVal**

```
char pulsar::TNSQFlagVal[MAX\_TNSQ][MAX\_FLAG\_LEN]
```

**13.18.2.269 TNSQVal**

```
double pulsar::TNSQVal[MAX\_TNSQ]
```

**13.18.2.270 TNsubtractDM**

```
int pulsar::TNsubtractDM
```

**13.18.2.271 TNsubtractRed**

```
int pulsar::TNsubtractRed
```

**13.18.2.272 ToAextraCovar**

```
double** pulsar::ToAextraCovar
```

**13.18.2.273 tOffset**

```
double pulsar::tOffset[MAX\_TOFFSET]
```

Offsets in TOAs in seconds

**13.18.2.274 tOffset\_f1**

```
double pulsar::tOffset_f1[MAX\_TOFFSET]
```

**13.18.2.275 tOffset\_f2**

```
double pulsar::tOffset_f2[MAX\_TOFFSET]
```

Range for offset to be applied



**13.18.2.276 tOffset\_t1**

```
double pulsar::tOffset_t1[MAX\_TOFFSET]
```

**13.18.2.277 tOffset\_t2**

```
double pulsar::tOffset_t2[MAX\_TOFFSET]
```

**13.18.2.278 tOffsetFlags**

```
char pulsar::tOffsetFlags[MAX\_TOFFSET][1000]
```

**13.18.2.279 tOffsetSite**

```
char pulsar::tOffsetSite[MAX\_TOFFSET][100]
```

**13.18.2.280 tzrsite**

```
char pulsar::tzrsite[100]
```

Site-code for polyco

**13.18.2.281 units**

```
int pulsar::units
```

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

**13.18.2.282 useCalceph**

```
int pulsar::useCalceph
```

**13.18.2.283 useTNorth**

```
int pulsar::useTNorth
```

**13.18.2.284 velPulsar**

```
double pulsar::velPulsar[3]
```

3-vector giving pulsar's velocity

**13.18.2.285 wave\_cos**

```
double pulsar::wave_cos[MAX_WHITE]
```

**13.18.2.286 wave\_cos\_dm**

```
double pulsar::wave_cos_dm[MAX_WHITE]
```

**13.18.2.287 wave\_cos\_dm\_err**

```
double pulsar::wave_cos_dm_err[MAX_WHITE]
```

**13.18.2.288 wave\_cos\_err**

```
double pulsar::wave_cos_err[MAX_WHITE]
```

**13.18.2.289 wave\_sine**

```
double pulsar::wave_sine[MAX_WHITE]
```

**13.18.2.290 wave\_sine\_dm**

```
double pulsar::wave_sine_dm[MAX_WHITE]
```

**13.18.2.291 wave\_sine\_dm\_err**

```
double pulsar::wave_sine_dm_err[MAX_WHITE]
```

#### 13.18.2.292 wave\_sine\_err

```
double pulsar::wave_sine_err[MAX_WHITE]
```

#### 13.18.2.293 waveScale

```
double pulsar::waveScale
```

#### 13.18.2.294 whiteNoiseModelFile

```
char pulsar::whiteNoiseModelFile[MAX_STRLEN]
```

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

- [tempo2.h](#)

## 13.19 storePrecision Struct Reference

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

### Public Attributes

- [longdouble minPrec](#)
- [char routine](#) [100]
- [char comment](#) [MAX\_STRLEN]

### 13.19.1 Member Data Documentation

#### 13.19.1.1 comment

```
char storePrecision::comment[MAX_STRLEN]
```

#### 13.19.1.2 minPrec

```
longdouble storePrecision::minPrec
```

### 13.19.1.3 routine

```
char storePrecision::routine[100]
```

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

- [tempo2.h](#)

## 13.20 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]

### 13.20.1 Member Data Documentation

#### 13.20.1.1 [binary\\_frequency](#)

```
double T1Polyco::binary_frequency
```

#### 13.20.1.2 [binary\\_phase](#)

```
double T1Polyco::binary_phase
```

**13.20.1.3 coeff**

```
long double T1Polyco::coeff[32]
```

**13.20.1.4 date\_string**

```
char T1Polyco::date_string[10]
```

**13.20.1.5 dm**

```
double T1Polyco::dm
```

**13.20.1.6 doppler**

```
double T1Polyco::doppler
```

**13.20.1.7 frequency\_obs**

```
double T1Polyco::frequency_obs
```

**13.20.1.8 frequency\_psr\_0**

```
long double T1Polyco::frequency_psr_0
```

**13.20.1.9 log10rms**

```
double T1Polyco::log10rms
```

**13.20.1.10 mjd\_mid**

```
long double T1Polyco::mjd_mid
```

#### 13.20.1.11 ncoeff

```
int T1Polyco::ncoeff
```

#### 13.20.1.12 psrname

```
char T1Polyco::psrname[64]
```

#### 13.20.1.13 reference\_phase

```
long double T1Polyco::reference_phase
```

#### 13.20.1.14 sitename

```
char T1Polyco::sitename[5]
```

#### 13.20.1.15 span

```
int T1Polyco::span
```

#### 13.20.1.16 utc\_string

```
char T1Polyco::utc_string[13]
```

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

- [tempo2pred.h](#)

## 13.21 T1PolycoSet Struct Reference

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

Collaboration diagram for T1PolycoSet:

## Public Attributes

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

### 13.21.1 Member Data Documentation

#### 13.21.1.1 nsegments

```
int T1PolycoSet::nsegments
```

#### 13.21.1.2 segments

```
T1Polyco* T1PolycoSet::segments
```

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

- [tempo2pred.h](#)

## 13.22 T2Predictor Struct Reference

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

Collaboration diagram for T2Predictor:

## Public Attributes

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

### 13.22.1 Member Data Documentation

#### 13.22.1.1 cheby

[ChebyModelSet](#) T2Predictor::cheby

#### 13.22.1.2 kind

[T2PredictorKind](#) T2Predictor::kind

#### 13.22.1.3 modelset

union { ... } T2Predictor::modelset

#### 13.22.1.4 t1

[T1PolycoSet](#) T2Predictor::t1

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

- [tempo2pred.h](#)

## 13.23 TabulatedFunction Struct Reference

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

Collaboration diagram for TabulatedFunction:

### Public Attributes

- char [fileName](#) [256]
- char [header\\_line](#) [256]
- [DynamicArray](#) [samples](#)

#### 13.23.1 Member Data Documentation



#### 13.23.1.1 fileName

```
char TabulatedFunction::fileName[256]
```

#### 13.23.1.2 header\_line

```
char TabulatedFunction::header_line[256]
```

#### 13.23.1.3 samples

```
DynamicArray TabulatedFunction::samples
```

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

- [tabulatedfunction.h](#)

## 13.24 TabulatedFunctionSample Struct Reference

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

### Public Attributes

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

#### 13.24.1 Member Data Documentation

##### 13.24.1.1 x

```
double TabulatedFunctionSample::x
```

##### 13.24.1.2 y

```
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 cholesky\_covarFunc2matrix()

```
void cholesky_covarFunc2matrix (  
    double ** m,  
    double * covarFunc,  
    int ndays,  
    double * resx,  
    double * resy,  
    double * rese,  
    int np,  
    int nc )
```

#### 14.1.1.2 cholesky\_dmModel()

```
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 cholesky\_dmModelCovarParam()

```
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 cholesky\_ecm()

```
void cholesky_ecm (
    double ** m,
    char * fileName,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc )
```

#### 14.1.1.5 cholesky\_formUinv()

```
int cholesky_formUinv (
    double ** uinv,
    double ** m,
    int np )
```

#### 14.1.1.6 cholesky\_powerlawModel()

```
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 cholesky\_powerlawModel\_withBeta()

```
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 cholesky\_readFromCovarianceFunction()

```
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 2018.09.1"`
- `#define PACKAGE_TARNAME "tempo2"`
- `#define PACKAGE_URL "http://www.bitbucket.org/psrsoft/tempo2"`
- `#define PACKAGE_VERSION "2018.09.1"`
- `#define QR_DEFAULT`
- `#define STDC_HEADERS 1`
- `#define TEMPO2_ARCH "darwin15"`
- `#define VERSION "2018.09.1"`
- `#define _DARWIN_USE_64_BIT_INODE 1`

### 14.3.1 Macro Definition Documentation

#### 14.3.1.1 `_DARWIN_USE_64_BIT_INODE`

```
#define _DARWIN_USE_64_BIT_INODE 1
```

#### 14.3.1.2 F77\_FUNC

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

#### 14.3.1.3 F77\_FUNC\_

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

#### 14.3.1.4 HAVE\_BLAS

```
#define HAVE_BLAS 1
```

#### 14.3.1.5 HAVE\_CFITSIO

```
#define HAVE_CFITSIO 1
```

#### 14.3.1.6 HAVE\_DLERROR

```
#define HAVE_DLERROR 1
```

#### 14.3.1.7 HAVE\_DLFCN\_H

```
#define HAVE_DLFCN_H 1
```

#### 14.3.1.8 HAVE\_FFTW3

```
#define HAVE_FFTW3 1
```

#### 14.3.1.9 HAVE\_INTTYPES\_H

```
#define HAVE_INTTYPES_H 1
```

#### 14.3.1.10 HAVE\_LAPACK

```
#define HAVE_LAPACK 1
```

#### 14.3.1.11 HAVE\_LIBDL

```
#define HAVE_LIBDL 1
```

#### 14.3.1.12 HAVE\_LIBDLLOADER

```
#define HAVE_LIBDLLOADER 1
```

#### 14.3.1.13 HAVE\_LIBM

```
#define HAVE_LIBM 1
```

#### 14.3.1.14 HAVE\_MEMORY\_H

```
#define HAVE_MEMORY_H 1
```

#### 14.3.1.15 HAVE\_PGPLOT

```
#define HAVE_PGPLOT 1
```

#### 14.3.1.16 HAVE\_PTHREAD

```
#define HAVE_PTHREAD 1
```



#### 14.3.1.17 HAVE\_STDINT\_H

```
#define HAVE_STDINT_H 1
```

#### 14.3.1.18 HAVE\_STDLIB\_H

```
#define HAVE_STDLIB_H 1
```

#### 14.3.1.19 HAVE\_STRING\_H

```
#define HAVE_STRING_H 1
```

#### 14.3.1.20 HAVE\_STRINGS\_H

```
#define HAVE_STRINGS_H 1
```

#### 14.3.1.21 HAVE\_SYS\_STAT\_H

```
#define HAVE_SYS_STAT_H 1
```

#### 14.3.1.22 HAVE\_SYS\_TYPES\_H

```
#define HAVE_SYS_TYPES_H 1
```

#### 14.3.1.23 HAVE\_UNISTD\_H

```
#define HAVE_UNISTD_H 1
```

#### 14.3.1.24 LT\_OBJDIR

```
#define LT_OBJDIR ".libs/"
```

**14.3.1.25 PACKAGE**

```
#define PACKAGE "tempo2"
```

**14.3.1.26 PACKAGE\_BUGREPORT**

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

**14.3.1.27 PACKAGE\_NAME**

```
#define PACKAGE_NAME "Tempo2"
```

**14.3.1.28 PACKAGE\_STRING**

```
#define PACKAGE_STRING "Tempo2 2018.09.1"
```

**14.3.1.29 PACKAGE\_TARNAME**

```
#define PACKAGE_TARNAME "tempo2"
```

**14.3.1.30 PACKAGE\_URL**

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

**14.3.1.31 PACKAGE\_VERSION**

```
#define PACKAGE_VERSION "2018.09.1"
```

**14.3.1.32 QR\_DEFAULT**

```
#define QR_DEFAULT
```

## 14.3.1.33 STDC\_HEADERS

```
#define STDC_HEADERS 1
```

## 14.3.1.34 TEMPO2\_ARCH

```
#define TEMPO2_ARCH "darwin15"
```

## 14.3.1.35 VERSION

```
#define VERSION "2018.09.1"
```

## 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, void \*special)

## 14.4.1 Function Documentation

#### 14.4.1.1 autosetDMCM()

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

#### 14.4.1.2 computeConstraintWeights()

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

#### 14.4.1.3 consFunc\_dmmodel\_cw()

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

#### 14.4.1.4 consFunc\_dmmodel\_cw\_year()

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

#### 14.4.1.5 consFunc\_dmmodel\_dm1()

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

#### 14.4.1.6 consFunc\_dmmodel\_mean()

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

#### 14.4.1.7 consFunc\_ifunc()

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

#### 14.4.1.8 consFunc\_ifunc\_year()

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

#### 14.4.1.9 consFunc\_qifunc\_c\_year()

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

#### 14.4.1.10 consFunc\_qifunc\_p\_year()

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

#### 14.4.1.11 consFunc\_quad\_ifunc\_c()

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

#### 14.4.1.12 consFunc\_quad\_ifunc\_p()

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

#### 14.4.1.13 consFunc\_tel\_dx()

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

#### 14.4.1.14 consFunc\_tel\_dy()

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

#### 14.4.1.15 consFunc\_tel\_dz()

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

#### 14.4.1.16 CONSTRAINTfuncs()

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

#### 14.4.1.17 get\_constraint\_name()

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

#### 14.4.1.18 standardConstraintFunctions()

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

## 14.5 constraints\_covar.h File Reference

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

Include dependency graph for constraints\_covar.h:

### Functions

- double [constraints\\_covar\\_ifunc](#) (pulsar \*psr, int ipsr, int iconstraint, int iparam, int constraintk, int k, void \*constraintSpecial)

#### 14.5.1 Function Documentation

#### 14.5.1.1 constraints\_covar\_ifunc()

```
double constraints_covar_ifunc (
    pulsar * psr,
    int ipsr,
    int iconstraint,
    int iparam,
    int constraintk,
    int k,
    void * constraintSpecial )
```

## 14.6 constraints\_nestlike.h File Reference

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

Include dependency graph for constraints\_nestlike.h:

### Functions

- double [constraints\\_nestlike\\_red](#) (pulsar \*psr, int ipsr, int iconstraint, int iparam, int constraintk, int k, void \*special)
- double [constraints\\_nestlike\\_red\\_dm](#) (pulsar \*psr, int ipsr, int iconstraint, int iparam, int constraintk, int k, void \*special)
- double [constraints\\_nestlike\\_jitter](#) (pulsar \*psr, int ipsr, int iconstraint, int iparam, int constraintk, int k, void \*special)
- double [constraints\\_nestlike\\_band](#) (pulsar \*psr, int ipsr, int iconstraint, int iparam, int constraintk, int k, void \*special)
- double [constraints\\_nestlike\\_group](#) (pulsar \*psr, int ipsr, int iconstraint, int iparam, int constraintk, int k, void \*special)

### 14.6.1 Function Documentation

#### 14.6.1.1 constraints\_nestlike\_band()

```
double constraints_nestlike_band (
    pulsar * psr,
    int ipsr,
    int iconstraint,
    int iparam,
    int constraintk,
    int k,
    void * special )
```



#### 14.6.1.2 constraints\_nestlike\_group()

```
double constraints_nestlike_group (  
    pulsar * psr,  
    int ipsr,  
    int iconstraint,  
    int iparam,  
    int constraintk,  
    int k,  
    void * special )
```

#### 14.6.1.3 constraints\_nestlike\_jitter()

```
double constraints_nestlike_jitter (  
    pulsar * psr,  
    int ipsr,  
    int iconstraint,  
    int iparam,  
    int constraintk,  
    int k,  
    void * special )
```

#### 14.6.1.4 constraints\_nestlike\_red()

```
double constraints_nestlike_red (  
    pulsar * psr,  
    int ipsr,  
    int iconstraint,  
    int iparam,  
    int constraintk,  
    int k,  
    void * special )
```

#### 14.6.1.5 constraints\_nestlike\_red\_dm()

```
double constraints_nestlike_red_dm (  
    pulsar * psr,  
    int ipsr,  
    int iconstraint,  
    int iparam,  
    int constraintk,  
    int k,  
    void * special )
```

## 14.7 constraints\_param.h File Reference

### Classes

- struct [constraint\\_param\\_info](#)

### Functions

- double [constraint\\_param\\_function](#) ([pulsar](#) \*psr, int ipsr, int iconstraint, int iparam, int constraintk, int k, void \*special)

### 14.7.1 Function Documentation

#### 14.7.1.1 constraint\_param\_function()

```
double constraint_param_function (
    pulsar * psr,
    int ipsr,
    int iconstraint,
    int iparam,
    int constraintk,
    int k,
    void * special )
```

## 14.8 documentation/1\_USER\_GUIDE.md File Reference

## 14.9 documentation/2\_developers.md File Reference

## 14.10 documentation/3\_DEVELOPER\_GUIDE.md File Reference

## 14.11 documentation/4\_directories.md File Reference

## 14.12 documentation/5\_plugins.md File Reference

## 14.13 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](#) \*)

### 14.13.1 Function Documentation

#### 14.13.1.1 [DynamicArray\\_free\(\)](#)

```
void DynamicArray_free (  
    DynamicArray * )
```

#### 14.13.1.2 [DynamicArray\\_init\(\)](#)

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

#### 14.13.1.3 [DynamicArray\\_push\\_back\(\)](#)

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

#### 14.13.1.4 [DynamicArray\\_resize\(\)](#)

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

## 14.14 enum\_str.h File Reference

### Variables

- const char \* [label\\_str](#) []
- const char \* [constraint\\_str](#) []

### 14.14.1 Variable Documentation

#### 14.14.1.1 [constraint\\_str](#)

```
const char* constraint_str[]
```

#### 14.14.1.2 [label\\_str](#)

```
const char* label_str[]
```

## 14.15 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 [dtdt](#) (double ec, double t, double p)
- double [Rs](#) (double m1)
- [longdouble eccRes](#) ([pulsar](#) \*psr, int i, int \*coalesceFlag, double \*prev\_p, double \*prev\_e, double \*prev\_a, double \*prev\_epoch, double \*prev\_theta)
- [longdouble eccResWithEnergy](#) ([pulsar](#) \*psr, int i, int \*coalesceFlag, double \*prev\_p, double \*prev\_e, double \*prev\_a, double \*prev\_epoch, double \*prev\_theta, float \*eOut)
- void [setupGW](#) ([gwSrc](#) \*gw)
- void [matrixMult](#) ([longdouble](#) m1[3][3], [longdouble](#) m2[3][3], [longdouble](#) out[3][3])
- [longdouble dotProduct](#) ([longdouble](#) \*m1, [longdouble](#) \*m2)
- void [GWbackground](#) ([gwSrc](#) \*gw, int numberGW, long \*idum, [longdouble](#) flo, [longdouble](#) fhi, double gwAmp, double alpha, int loglin)
- [longdouble calculateResidualGW](#) ([longdouble](#) \*kp, [gwSrc](#) \*gw, [longdouble](#) time, [longdouble](#) dist)
- void [setupPulsar\\_GWsim](#) ([longdouble](#) ra\_p, [longdouble](#) dec\_p, [longdouble](#) \*kp)
- int [GWbackground\\_read](#) ([gwSrc](#) \*gw, FILE \*file, int ireal)
- void [GWbackground\\_write](#) ([gwSrc](#) \*gw, FILE \*file, int ngw, int ireal)
- double [psrangle](#) (double centre\_long, double centre\_lat, double psr\_long, double psr\_lat)
- double [sphharm](#) (int l, int m, double x)
- double [Findphi](#) (double prob, double amp, double phase)
- void [setupgeneralGW](#) ([gwgeneralSrc](#) \*gw)
- void [GWgeneralbackground](#) ([gwgeneralSrc](#) \*gw, int \*numberGW, long \*idum, [longdouble](#) flo, [longdouble](#) fhi, [gwgenSpec](#) gwAmps, int loglin)
- void [GWgeneralanisotropicbackground](#) ([gwgeneralSrc](#) \*gw, int \*numberGW, long \*idum, [longdouble](#) flo, [longdouble](#) fhi, [gwgenSpec](#) gwAmps, int loglin, double \*\*\*harmlist, int \*nharms)
- void [GWanisotropicbackground](#) ([gwSrc](#) \*gw, int numberGW, long \*idum, [longdouble](#) flo, [longdouble](#) fhi, double gwAmp, double alpha, int loglin, double \*\*harmlist, int nharms)
- void [GWdipolebackground](#) ([gwSrc](#) \*gw, int numberGW, long \*idum, [longdouble](#) flo, [longdouble](#) fhi, double gwAmp, double alpha, int loglin, double \*dipoleamps)
- [longdouble calculateResidualgeneralGW](#) ([longdouble](#) \*kp, [gwgeneralSrc](#) \*gw, [longdouble](#) time, [longdouble](#) dist)
- int [GWgeneralbackground\\_read](#) ([gwgeneralSrc](#) \*gw, FILE \*file, int ireal)
- void [GWgeneralbackground\\_write](#) ([gwgeneralSrc](#) \*gw, FILE \*file, int ngw, int ireal)

### 14.15.1 Typedef Documentation

#### 14.15.1.1 gwgeneralSrc

```
typedef struct gwgeneralSrc gwgeneralSrc
```

#### 14.15.1.2 gwgenSpec

```
typedef struct gwgenSpec gwgenSpec
```

### 14.15.1.3 gwSrc

```
typedef struct gwSrc gwSrc
```

## 14.15.2 Function Documentation

### 14.15.2.1 calculateResidualgeneralGW()

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

### 14.15.2.2 calculateResidualGW()

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

### 14.15.2.3 dadt()

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

### 14.15.2.4 dedt()

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

#### 14.15.2.5 dotProduct()

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

#### 14.15.2.6 dtdt()

```
double dtdt (
    double ec,
    double t,
    double p )
```

#### 14.15.2.7 eccRes()

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

#### 14.15.2.8 eccResWithEnergy()

```
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 )
```

#### 14.15.2.9 Fe()

```
double Fe (
    double ec )
```

**14.15.2.10 Findphi()**

```
double Findphi (
    double prob,
    double amp,
    double phase )
```

**14.15.2.11 GWanisotropicbackground()**

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

**14.15.2.12 GWbackground()**

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

**14.15.2.13 GWbackground\_read()**

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



#### 14.15.2.14 GWbackground\_write()

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

#### 14.15.2.15 GWdipolebackground()

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

#### 14.15.2.16 GWgeneralanisotropicbackground()

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

#### 14.15.2.17 GWgeneralbackground()

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

**14.15.2.18 GWgeneralbackground\_read()**

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

**14.15.2.19 GWgeneralbackground\_write()**

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

**14.15.2.20 matrixMult()**

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

**14.15.2.21 psrangle()**

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

**14.15.2.22 Rs()**

```
double Rs (
    double m1 )
```

**14.15.2.23 setupgeneralGW()**

```
void setupgeneralGW (
    gwgeneralSrc * gw )
```

## 14.15.2.24 setupGW()

```
void setupGW (
    gwSrc * gw )
```

## 14.15.2.25 setupPulsar\_GWsim()

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

## 14.15.2.26 sphharm()

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

## 14.16 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)

## 14.16.1 Macro Definition Documentation

### 14.16.1.1 IFTE\_JD0

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

### 14.16.1.2 IFTE\_K

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

### 14.16.1.3 IFTE\_KM1

```
#define IFTE_KM1 1.55051979176e-8
```

### 14.16.1.4 IFTE\_LC

```
#define IFTE_LC 1.48082686742e-8
```

### 14.16.1.5 IFTE\_MJD0

```
#define IFTE_MJD0 43144.0003725
```

### 14.16.1.6 IFTE\_TEPH0

```
#define IFTE_TEPH0 -65.564518e-6
```

## 14.16.2 Function Documentation

#### 14.16.2.1 IFTE\_close\_file()

```
void IFTE_close_file ( )
```

#### 14.16.2.2 IFTE\_DeltaT()

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

#### 14.16.2.3 IFTE\_DeltaTDot()

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

#### 14.16.2.4 IFTE\_get\_DeltaT\_DeltaTDot()

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

#### 14.16.2.5 IFTE\_get\_vE()

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

#### 14.16.2.6 IFTE\_get\_vE\_vEDot()

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

### 14.16.2.7 IFTE\_get\_vEDot()

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

### 14.16.2.8 IFTE\_init()

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

## 14.17 ifunc.h File Reference

### Functions

- double [ifunc](#) (const double \*mjd, const double t, const int N, const int k)
- double [ifunc](#) (const double \*mjd, const double \*yoffs, const double t, const int N)
- double [sinfunc](#) (const double \*T, const double t, const int k)

### 14.17.1 Function Documentation

#### 14.17.1.1 ifunc() [1/2]

```
double ifunc (
    const double * mjd,
    const double t,
    const int N,
    const int k )
```

Compute an ifunc gradient for a given 'k'

#### 14.17.1.2 ifunc() [2/2]

```
double ifunc (
    const double * mjd,
    const double * yoffs,
    const double t,
    const int N )
```

Compute an ifunc summed over all elements.

### 14.17.1.3 sinfunc()

```
double sinfunc (
    const double * T,
    const double t,
    const int k )
```

## 14.18 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

## 14.18.1 Macro Definition Documentation

### 14.18.1.1 JPL\_HEADER\_SIZE

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

### 14.18.1.2 MAX\_CHEBY

```
#define MAX_CHEBY 18
```

## 14.19 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)`

#### 14.19.1 Macro Definition Documentation



#### 14.19.1.1 DLL\_FUNC

```
#define DLL_FUNC
```

#### 14.19.1.2 JPL\_EPH\_FSEEK\_ERROR

```
#define JPL_EPH_FSEEK_ERROR (-6)
```

#### 14.19.1.3 JPL\_EPH\_INVALID\_INDEX

```
#define JPL_EPH_INVALID_INDEX (-5)
```

#### 14.19.1.4 JPL\_EPH\_OUTSIDE\_RANGE

```
#define JPL_EPH_OUTSIDE_RANGE (-1)
```

#### 14.19.1.5 JPL\_EPH\_QUANTITY\_NOT\_IN\_EPHEMERIS

```
#define JPL_EPH_QUANTITY_NOT_IN_EPHEMERIS (-3)
```

#### 14.19.1.6 JPL\_EPH\_READ\_ERROR

```
#define JPL_EPH_READ_ERROR (-2)
```

#### 14.19.1.7 JPL\_EPHEM\_AU\_IN\_KM

```
#define JPL_EPHEM_AU_IN_KM 28
```

#### 14.19.1.8 JPL\_EPHEM\_EARTH\_MOON\_RATIO

```
#define JPL_EPHEM_EARTH_MOON_RATIO 36
```

**14.19.1.9 JPL\_EPHEM\_END\_JD**

```
#define JPL_EPHEM_END_JD 8
```

**14.19.1.10 JPL\_EPHEM\_EPHEMERIS\_VERSION**

```
#define JPL_EPHEM_EPHEMERIS_VERSION 224
```

**14.19.1.11 JPL\_EPHEM\_IPT\_ARRAY**

```
#define JPL_EPHEM_IPT_ARRAY 44
```

**14.19.1.12 JPL\_EPHEM\_KERNEL\_NCOEFF**

```
#define JPL_EPHEM_KERNEL_NCOEFF 236
```

**14.19.1.13 JPL\_EPHEM\_KERNEL\_RECORD\_SIZE**

```
#define JPL_EPHEM_KERNEL_RECORD_SIZE 232
```

**14.19.1.14 JPL\_EPHEM\_KERNEL\_SIZE**

```
#define JPL_EPHEM_KERNEL_SIZE 228
```

**14.19.1.15 JPL\_EPHEM\_KERNEL\_SWAP\_BYTES**

```
#define JPL_EPHEM_KERNEL_SWAP_BYTES 240
```

**14.19.1.16 JPL\_EPHEM\_N\_CONSTANTS**

```
#define JPL_EPHEM_N_CONSTANTS 24
```

**14.19.1.17 JPL\_EPHEM\_START\_JD**

```
#define JPL_EPHEM_START_JD 0
```

**14.19.1.18 JPL\_EPHEM\_STEP**

```
#define JPL_EPHEM_STEP 16
```

**14.19.1.19 jpl\_get\_pvsun**

```
#define jpl_get_pvsun(  
    ephem ) ((double *) ((char *) ephem + 248))
```

**14.19.1.20 JPL\_INIT\_FILE\_CORRUPT**

```
#define JPL_INIT_FILE_CORRUPT -5
```

**14.19.1.21 JPL\_INIT\_FILE\_NOT\_FOUND**

```
#define JPL_INIT_FILE_NOT_FOUND -1
```

**14.19.1.22 JPL\_INIT\_FREAD2\_FAILED**

```
#define JPL_INIT_FREAD2_FAILED -4
```

**14.19.1.23 JPL\_INIT\_FREAD3\_FAILED**

```
#define JPL_INIT_FREAD3_FAILED -7
```

**14.19.1.24 JPL\_INIT\_FREAD4\_FAILED**

```
#define JPL_INIT_FREAD4_FAILED -8
```

**14.19.1.25 JPL\_INIT\_FREAD5\_FAILED**

```
#define JPL_INIT_FREAD5_FAILED -10
```

**14.19.1.26 JPL\_INIT\_FREAD\_FAILED**

```
#define JPL_INIT_FREAD_FAILED -3
```

**14.19.1.27 JPL\_INIT\_FSEEK\_FAILED**

```
#define JPL_INIT_FSEEK_FAILED -2
```

**14.19.1.28 JPL\_INIT\_MEMORY\_FAILURE**

```
#define JPL_INIT_MEMORY_FAILURE -6
```

**14.19.1.29 JPL\_INIT\_NO\_ERROR**

```
#define JPL_INIT_NO_ERROR 0
```

**14.19.1.30 JPL\_INIT\_NOT\_CALLED**

```
#define JPL_INIT_NOT_CALLED -9
```

**14.19.2 Function Documentation**

#### 14.19.2.1 jpl\_close\_ephemeris()

```
void DLL_FUNC jpl_close_ephemeris (
    void * ephem )
```

#### 14.19.2.2 jpl\_get\_constant()

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

#### 14.19.2.3 jpl\_get\_double()

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

#### 14.19.2.4 jpl\_get\_long()

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

#### 14.19.2.5 jpl\_init\_ephemeris()

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

#### 14.19.2.6 jpl\_init\_error\_code()

```
int DLL_FUNC jpl_init_error_code (
    void )
```

#### 14.19.2.7 jpl\_pleph()

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

#### 14.19.2.8 jpl\_state()

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

#### 14.19.2.9 make\_sub\_ephem()

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

## 14.20 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](#)

## 14.20.1 Function Documentation

### 14.20.1.1 close\_file()

```
void close_file ( )
```

### 14.20.1.2 open\_file()

```
int open_file (
    char * fname )
```

### 14.20.1.3 read\_char()

```
char read_char ( )
```

### 14.20.1.4 read\_character()

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

### 14.20.1.5 read\_double()

```
double read_double ( )
```

### 14.20.1.6 read\_float()

```
float read_float ( )
```

#### 14.20.1.7 read\_int()

```
int read_int ( )
```

#### 14.20.1.8 read\_record\_int()

```
int read_record_int ( )
```

### 14.20.2 Variable Documentation

#### 14.20.2.1 c\_fileptr

```
FILE* c_fileptr
```

#### 14.20.2.2 swapByte

```
int swapByte
```

## 14.21 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](#)



### 14.21.1 Function Documentation

#### 14.21.1.1 close\_file2()

```
void close_file2 ( )
```

#### 14.21.1.2 open\_file2()

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

#### 14.21.1.3 read\_character2()

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

#### 14.21.1.4 read\_double2()

```
double read_double2 ( )
```

#### 14.21.1.5 read\_float2()

```
float read_float2 ( )
```

#### 14.21.1.6 read\_int2()

```
int read_int2 ( )
```

#### 14.21.1.7 read\_record\_int2()

```
int read_record_int2 ( )
```

### 14.21.2 Variable Documentation

#### 14.21.2.1 c\_fileptr2

```
FILE* c_fileptr2
```

#### 14.21.2.2 swapByte2

```
int swapByte2
```

## 14.22 README.md File Reference

## 14.23 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 \*\*Ocvm, char rescale\_errors)
- 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](#)

## 14.23.1 Macro Definition Documentation

### 14.23.1.1 ACCEL\_LSQ

```
#define ACCEL_LSQ
```

### 14.23.1.2 ACCEL\_MULTMATRIX

```
#define ACCEL_MULTMATRIX
```

### 14.23.1.3 ACCEL\_UINV

```
#define ACCEL_UINV
```

## 14.23.2 Function Documentation

### 14.23.2.1 accel\_lsqr()

```
double accel_lsqr (
    double ** dm,
    double * data,
    double * oparm,
    int ndata,
    int nparam,
    double ** Ocvr,
    char rescale_errors )
```

### 14.23.2.2 accel\_multMatrix()

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

### 14.23.2.3 accel\_multMatrixVec()

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

### 14.23.2.4 accel\_uinv()

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

## 14.23.3 Variable Documentation

### 14.23.3.1 useT2accel

```
char useT2accel
```

## 14.24 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, const double \*psr\_x, const int \*psr\_↔  
toaidx, const int psr\_ndata)
- 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)
- double [t2Fit\\_getParamDeriv](#) ([pulsar](#) \*psr, const [param\\_label](#) fit\_param, const double x, const int i, const int k)
- int [t2Fit\\_getParamMatrixRow](#) (const [FitInfo](#) &fitinfo, const int ipsr, const [param\\_label](#) fit\_param, const int k)

### 14.24.1 Function Documentation

#### 14.24.1.1 t2Fit()

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

#### 14.24.1.2 t2Fit\_buildConstraintsMatrix()

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

#### 14.24.1.3 t2Fit\_buildDesignMatrix()

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

#### 14.24.1.4 t2Fit\_fillFitInfo()

```
void t2Fit_fillFitInfo (
    pulsar * psr,
    FitInfo & OUT,
    const FitInfo & globals,
    const double * psr_x,
    const int * psr_toaidx,
    const int psr_ndata )
```

#### 14.24.1.5 t2Fit\_fillGlobalFitInfo()

```
void t2Fit_fillGlobalFitInfo (
    pulsar * psr,
    unsigned int npsr,
    FitInfo & OUT )
```

#### 14.24.1.6 t2Fit\_getFitData()

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

#### 14.24.1.7 t2Fit\_getParamDeriv()

```
double t2Fit_getParamDeriv (
    pulsar * psr,
    const param_label fit_param,
    const double x,
    const int i,
    const int k )
```

#### 14.24.1.8 t2Fit\_getParamMatrixRow()

```
int t2Fit_getParamMatrixRow (
    const FitInfo & fitinfo,
    const int ipsr,
    const param_label fit_param,
    const int k )
```

#### 14.24.1.9 t2Fit\_updateParameters()

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

### 14.25 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)

## 14.25.1 Function Documentation

### 14.25.1.1 t2FitFunc\_dmmodelCM()

```
double t2FitFunc_dmmodelCM (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

### 14.25.1.2 t2FitFunc\_dmmodelDM()

```
double t2FitFunc_dmmodelDM (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

### 14.25.1.3 t2UpdateFunc\_dmmodelCM()

```
void t2UpdateFunc_dmmodelCM (  
    pulsar * psr,  
    int ipsr,  
    param_label label,  
    int k,  
    double val,  
    double err )
```

### 14.25.1.4 t2UpdateFunc\_dmmodelDM()

```
void t2UpdateFunc_dmmodelDM (  
    pulsar * psr,  
    int ipsr,  
    param_label label,  
    int k,  
    double val,  
    double err )
```

## 14.26 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 ipsr, double x, int ipos, [param\\_label](#) label, int k)
- double [t2FitFunc\\_dmsinusoids](#) ([pulsar](#) \*psr, int ipsr, double x, int ipos, [param\\_label](#) label, int k)
- double [t2FitFunc\\_fd](#) ([pulsar](#) \*psr, int ipsr, double x, int ipos, [param\\_label](#) label, int k)
- double [t2FitFunc\\_fddc](#) ([pulsar](#) \*psr, int ipsr, double x, int ipos, [param\\_label](#) label, int k)
- double [t2FitFunc\\_ne\\_sw](#) ([pulsar](#) \*psr, int ipsr, double x, int ipos, [param\\_label](#) label, int k)
- void [t2UpdateFunc\\_ne\\_sw](#) ([pulsar](#) \*psr, int ipsr, [param\\_label](#) label, int k, double val, double error)

### 14.26.1 Function Documentation

#### 14.26.1.1 t2FitFunc\_dmsinusoids()

```
double t2FitFunc_dmsinusoids (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param\_label label,
    int k )
```

#### 14.26.1.2 t2FitFunc\_dmx()

```
double t2FitFunc_dmx (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param\_label label,
    int k )
```

#### 14.26.1.3 t2FitFunc\_fd()

```
double t2FitFunc_fd (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param\_label label,
    int k )
```



#### 14.26.1.4 t2FitFunc\_fddc()

```
double t2FitFunc_fddc (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

#### 14.26.1.5 t2FitFunc\_ne\_sw()

```
double t2FitFunc_ne_sw (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

#### 14.26.1.6 t2UpdateFunc\_ne\_sw()

```
void t2UpdateFunc_ne_sw (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double error )
```

## 14.27 t2fit\_fitwaves.h File Reference

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

### Functions

- double [t2FitFunc\\_fitwaves](#) (pulsar \*psr, int ipsr, double x, int ipos, param\_label label, int k)
- void [t2UpdateFunc\\_fitwaves](#) (pulsar \*psr, int ipsr, param\_label label, int k, double val, double err)

#### 14.27.1 Function Documentation

#### 14.27.1.1 t2FitFunc\_fitwaves()

```
double t2FitFunc_fitwaves (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

#### 14.27.1.2 t2UpdateFunc\_fitwaves()

```
void t2UpdateFunc_fitwaves (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.28 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)

### 14.28.1 Function Documentation

#### 14.28.1.1 t2FitFunc\_stdGlitch()

```
double t2FitFunc_stdGlitch (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

## 14.28.1.2 t2UpdateFunc\_stdGlitch()

```
void t2UpdateFunc_stdGlitch (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.29 t2fit\_gw.h File Reference

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

Include dependency graph for t2fit\_gw.h: This graph shows which files directly or indirectly include this file:

## Functions

- double [t2FitFunc\\_gwm\\_amp](#) (pulsar \*psr, int ipsr, double x, int ipos, param\_label label, int k)
- double [t2FitFunc\\_gwb\\_amp](#) (pulsar \*psr, int ipsr, double x, int ipos, param\_label label, int k)
- double [t2FitFunc\\_gwcs\\_amp](#) (pulsar \*psr, int ipsr, double x, int ipos, param\_label label, int k)
- double [t2FitFunc\\_quad\\_om](#) (pulsar \*psr, int ipsr, double x, int ipos, param\_label label, int k)
- double [t2FitFunc\\_gwsingle](#) (pulsar \*psr, int ipsr, double x, int ipos, param\_label label, int k)
- void [t2UpdateFunc\\_gwsingle](#) (pulsar \*psr, int ipsr, param\_label label, int k, double val, double err)
- void [t2UpdateFunc\\_quad\\_om](#) (pulsar \*psr, int ipsr, param\_label label, int k, double val, double err)

## 14.29.1 Function Documentation

## 14.29.1.1 t2FitFunc\_gwb\_amp()

```
double t2FitFunc_gwb_amp (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

## 14.29.1.2 t2FitFunc\_gwcs\_amp()

```
double t2FitFunc_gwcs_amp (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

#### 14.29.1.3 t2FitFunc\_gwm\_amp()

```
double t2FitFunc_gwm_amp (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.29.1.4 t2FitFunc\_gwsingle()

```
double t2FitFunc_gwsingle (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.29.1.5 t2FitFunc\_quad\_om()

```
double t2FitFunc_quad_om (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.29.1.6 t2UpdateFunc\_gwsingle()

```
void t2UpdateFunc_gwsingle (  
    pulsar * psr,  
    int ipsr,  
    param_label label,  
    int k,  
    double val,  
    double err )
```

#### 14.29.1.7 t2UpdateFunc\_quad\_om()

```
void t2UpdateFunc_quad_om (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.30 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 [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)

### 14.30.1 Function Documentation

#### 14.30.1.1 t2FitFunc\_ifunc()

```
double t2FitFunc_ifunc (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

#### 14.30.1.2 t2FitFunc\_sifunc()

```
double t2FitFunc_sifunc (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

### 14.30.1.3 t2UpdateFunc\_ifunc()

```
void t2UpdateFunc_ifunc (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.31 t2fit\_nestlike.h File Reference

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

Include dependency graph for t2fit\_nestlike.h:

### Functions

- double [t2FitFunc\\_nestlike\\_red](#) (pulsar \*psr, int ipsr, double x, int ipos, [param\\_label label](#), int k)
- void [t2UpdateFunc\\_nestlike\\_red](#) (pulsar \*psr, int ipsr, [param\\_label label](#), int k, double val, double err)
- double [t2FitFunc\\_nestlike\\_red\\_dm](#) (pulsar \*psr, int ipsr, double x, int ipos, [param\\_label label](#), int k)
- void [t2UpdateFunc\\_nestlike\\_red\\_dm](#) (pulsar \*psr, int ipsr, [param\\_label label](#), int k, double val, double err)
- double [t2FitFunc\\_nestlike\\_jitter](#) (pulsar \*psr, int ipsr, double x, int ipos, [param\\_label label](#), int k)
- void [t2UpdateFunc\\_nestlike\\_jitter](#) (pulsar \*psr, int ipsr, [param\\_label label](#), int k, double val, double err)
- double [t2FitFunc\\_nestlike\\_band](#) (pulsar \*psr, int ipsr, double x, int ipos, [param\\_label label](#), int k)
- void [t2UpdateFunc\\_nestlike\\_band](#) (pulsar \*psr, int ipsr, [param\\_label label](#), int k, double val, double err)
- double [t2FitFunc\\_nestlike\\_group](#) (pulsar \*psr, int ipsr, double x, int ipos, [param\\_label label](#), int k)
- void [t2UpdateFunc\\_nestlike\\_group](#) (pulsar \*psr, int ipsr, [param\\_label label](#), int k, double val, double err)

### 14.31.1 Function Documentation

#### 14.31.1.1 t2FitFunc\_nestlike\_band()

```
double t2FitFunc_nestlike_band (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

#### 14.31.1.2 t2FitFunc\_nestlike\_group()

```
double t2FitFunc_nestlike_group (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.31.1.3 t2FitFunc\_nestlike\_jitter()

```
double t2FitFunc_nestlike_jitter (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.31.1.4 t2FitFunc\_nestlike\_red()

```
double t2FitFunc_nestlike_red (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.31.1.5 t2FitFunc\_nestlike\_red\_dm()

```
double t2FitFunc_nestlike_red_dm (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.31.1.6 t2UpdateFunc\_nestlike\_band()

```
void t2UpdateFunc_nestlike_band (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

#### 14.31.1.7 t2UpdateFunc\_nestlike\_group()

```
void t2UpdateFunc_nestlike_group (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

#### 14.31.1.8 t2UpdateFunc\_nestlike\_jitter()

```
void t2UpdateFunc_nestlike_jitter (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

#### 14.31.1.9 t2UpdateFunc\_nestlike\_red()

```
void t2UpdateFunc_nestlike_red (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```



## 14.31.1.10 t2UpdateFunc\_nestlike\_red\_dm()

```
void t2UpdateFunc_nestlike_red_dm (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.32 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)

## 14.32.1 Function Documentation

## 14.32.1.1 t2FitFunc\_stdPosition()

```
double t2FitFunc_stdPosition (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

## 14.32.1.2 t2UpdateFunc\_stdPosition()

```
void t2UpdateFunc_stdPosition (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.33 t2fit\_stdFitFuncs.h File Reference

```
#include <tempo2.h>
#include "t2fit_position.h"
#include "t2fit_fitwaves.h"
#include "t2fit_glitch.h"
#include "t2fit_ifunc.h"
#include "t2fit_dmmodel.h"
#include "t2fit_dmother.h"
#include "t2fit_gw.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)
- double [t2FitFunc\\_stdDm](#) ([pulsar](#) \*psr, int ipsr, double x, int ipos, [param\\_label](#) label, int k)
- double [t2FitFunc\\_telPos\\_delta](#) ([pulsar](#) \*psr, int ipsr, double x, int ipos, [param\\_label](#) label, int k)
- void [t2UpdateFunc\\_telPos\\_delta](#) ([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)
- 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)

### 14.33.1 Function Documentation

#### 14.33.1.1 t2FitFunc\_binaryModels()

```
double t2FitFunc_binaryModels (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param\_label label,
    int k )
```

#### 14.33.1.2 t2FitFunc\_ifunc()

```
double t2FitFunc_ifunc (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.3 t2FitFunc\_jump()

```
double t2FitFunc_jump (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.4 t2FitFunc\_notImplemented()

```
double t2FitFunc_notImplemented (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.5 t2FitFunc\_planet()

```
double t2FitFunc_planet (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.6 t2FitFunc\_stdDm()

```
double t2FitFunc_stdDm (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.7 t2FitFunc\_stdFreq()

```
double t2FitFunc_stdFreq (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.8 t2FitFunc\_telPos()

```
double t2FitFunc_telPos (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.9 t2FitFunc\_telPos\_delta()

```
double t2FitFunc_telPos_delta (  
    pulsar * psr,  
    int ipsr,  
    double x,  
    int ipos,  
    param_label label,  
    int k )
```

#### 14.33.1.10 t2FitFunc\_zero()

```
double t2FitFunc_zero (
    pulsar * psr,
    int ipsr,
    double x,
    int ipos,
    param_label label,
    int k )
```

#### 14.33.1.11 t2UpdateFunc\_binaryModels()

```
void t2UpdateFunc_binaryModels (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

#### 14.33.1.12 t2UpdateFunc\_ifunc()

```
void t2UpdateFunc_ifunc (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

#### 14.33.1.13 t2UpdateFunc\_jump()

```
void t2UpdateFunc_jump (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

#### 14.33.1.14 t2UpdateFunc\_notImplemented()

```
void t2UpdateFunc_notImplemented (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

#### 14.33.1.15 t2UpdateFunc\_simpleAdd()

```
void t2UpdateFunc_simpleAdd (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double error )
```

#### 14.33.1.16 t2UpdateFunc\_simpleMinus()

```
void t2UpdateFunc_simpleMinus (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double error )
```

#### 14.33.1.17 t2UpdateFunc\_stdFreq()

```
void t2UpdateFunc_stdFreq (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.33.1.18 t2UpdateFunc\_telPos\_delta()

```
void t2UpdateFunc_telPos_delta (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.33.1.19 t2UpdateFunc\_zero()

```
void t2UpdateFunc_zero (
    pulsar * psr,
    int ipsr,
    param_label label,
    int k,
    double val,
    double err )
```

## 14.34 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 nobS)
- void [TKsort\\_3d](#) (double \*val, double \*val2, double \*val3, int nobS)
- void [TKzeromean\\_d](#) (int n, double \*y)
- double [TKranDev](#) (long \*seed)
- double [TKgaussDev](#) (long \*seed)
- long [TKsetSeed](#) ()
- void [init\\_genrand](#) (unsigned long s)
- unsigned long [genrand\\_int32](#) (void)
- double [genrand\\_real1](#) (void)

### 14.34.1 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

### 14.34.2 Function Documentation

#### 14.34.2.1 [genrand\\_int32\(\)](#)

```
unsigned long genrand_int32 (  
    void )
```

#### 14.34.2.2 [genrand\\_real1\(\)](#)

```
double genrand_reall (  
    void )
```

#### 14.34.2.3 [init\\_genrand\(\)](#)

```
void init_genrand (  
    unsigned long s )
```



#### 14.34.2.4 TKconvertFloat1()

```
void TKconvertFloat1 (
    double * x,
    float * ox,
    int n )
```

#### 14.34.2.5 TKconvertFloat2()

```
void TKconvertFloat2 (
    double * x,
    double * y,
    float * ox,
    float * oy,
    int n )
```

#### 14.34.2.6 TKfindMax\_d()

```
double TKfindMax_d (
    double * x,
    int n )
```

#### 14.34.2.7 TKfindMax\_f()

```
float TKfindMax_f (
    float * x,
    int n )
```

#### 14.34.2.8 TKfindMedian\_d()

```
double TKfindMedian_d (
    double * val,
    int count )
```

#### 14.34.2.9 TKfindMedian\_f()

```
float TKfindMedian_f (
    float * val,
    int count )
```

**14.34.2.10 TKfindMin\_d()**

```
double TKfindMin_d (
    double * x,
    int n )
```

**14.34.2.11 TKfindMin\_f()**

```
float TKfindMin_f (
    float * x,
    int n )
```

**14.34.2.12 TKfindRMS\_d()**

```
double TKfindRMS_d (
    double * x,
    int n )
```

**14.34.2.13 TKfindRMS\_f()**

```
float TKfindRMS_f (
    float * x,
    int n )
```

**14.34.2.14 TKfindRMSweight\_d()**

```
float TKfindRMSweight_d (
    double * x,
    double * e,
    int n )
```

**14.34.2.15 TKgaussDev()**

```
double TKgaussDev (
    long * seed )
```

**14.34.2.16 TKmean\_d()**

```
double TKmean_d (  
    double * x,  
    int n )
```

**14.34.2.17 TKmean\_f()**

```
float TKmean_f (  
    float * x,  
    int n )
```

**14.34.2.18 TKranDev()**

```
double TKranDev (  
    long * seed )
```

**14.34.2.19 TKrange\_d()**

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

**14.34.2.20 TKrange\_f()**

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

**14.34.2.21 TKretMax\_d()**

```
double TKretMax_d (  
    double a,  
    double b )
```

**14.34.2.22 TKretMax\_f()**

```
float TKretMax_f (
    float a,
    float b )
```

**14.34.2.23 TKretMin\_d()**

```
double TKretMin_d (
    double a,
    double b )
```

**14.34.2.24 TKretMin\_f()**

```
float TKretMin_f (
    float a,
    float b )
```

**14.34.2.25 TKretMin\_i()**

```
int TKretMin_i (
    int a,
    int b )
```

**14.34.2.26 TKsetSeed()**

```
long TKsetSeed ( )
```

**14.34.2.27 TKsign\_d()**

```
double TKsign_d (
    double a,
    double b )
```

**14.34.2.28 TKsort\_2f()**

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

**14.34.2.29 TKsort\_3d()**

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

**14.34.2.30 TKsort\_d()**

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

**14.34.2.31 TKsort\_f()**

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

**14.34.2.32 TKvariance\_d()**

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

**14.34.2.33 TKzeromean\_d()**

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

## 14.35 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)

### 14.35.1 Function Documentation

#### 14.35.1.1 [TabulatedFunction\\_getEndX\(\)](#)

```
double TabulatedFunction_getEndX (  
    TabulatedFunction * func )
```

#### 14.35.1.2 [TabulatedFunction\\_getStartX\(\)](#)

```
double TabulatedFunction_getStartX (  
    TabulatedFunction * func )
```

#### 14.35.1.3 [TabulatedFunction\\_getValue\(\)](#)

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

## 14.35.1.4 TabulatedFunction\_load()

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

## 14.36 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 [FitOutput](#)
- 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 observation.*
- struct [pulsar](#)
  - contains the details for a single pulsar.*
- struct [observatory](#)

## Macros

- #define [TEMPO2\\_h\\_HASH](#) "\$Id: 0dc79e3a4c8b3cac82106bc030d9b8145e15d6f1 \$"
- #define [TEMPO2\\_h\\_VER](#) "2018.09.1"
- #define [TEMPO2\\_h\\_MAJOR\\_VER](#) 2018.09
- #define [TEMPO2\\_h\\_MINOR\\_VER](#) 1
- #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\_SX 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 MASR2RADS 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, void \*)  
*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 `FitOutput` `FitOutput`
- typedef struct `FitInfo` `FitInfo`  
*contains details of the fit*
- typedef struct `storePrecision` `storePrecision`
- typedef struct `parameter` `parameter`  
*Holds the values for a parameter.*
- typedef struct `observation` `observation`  
*A struct containing the details of a single obesrvation.*
- typedef struct `pulsar` `pulsar`  
*contains the details for a single pulsar.*

## Enumerations

- enum `label` {  
`param_raj`, `param_decj`, `param_f`, `param_pepoch`,  
`param_posepoch`, `param_dmepoch`, `param_dm`, `param_pmra`,  
`param_pmdec`, `param_px`, `param_sini`, `param_pb`,  
`param_fb`, `param_t0`, `param_a1`, `param_om`,  
`param_pmr`, `param_ecc`, `param_edot`, `param_e2dot`,  
`param_xpbdot`, `param_pbdot`, `param_a1dot`, `param_a2dot`,  
`param_omdot`, `param_om2dot`, `param_orbpx`, `param_tasc`,  
`param_eps1`, `param_eps2`, `param_m2`, `param_gamma`,  
`param_mtot`, `param_glep`, `param_glph`, `param_glf0`,  
`param_glf1`, `param_glf2`, `param_glf0d`, `param_gltd`,  
`param_start`, `param_finish`, `param_track`, `param_bp`,  
`param_bpp`, `param_tzrmjd`, `param_tzrfreq`, `param_fddc`,  
`param_fddi`, `param_fd`, `param_dr`, `param_dtheta`,  
`param_tspan`, `param_bpjep`, `param_bpjph`, `param_bpja1`,  
`param_bpjec`, `param_bpjom`, `param_bpjpb`, `param_wave_om`,  
`param_kom`, `param_kin`, `param_shapmax`, `param_dth`,  
`param_a0`, `param_b0`, `param_xomdot`, `param_afac`,  
`param_eps1dot`, `param_eps2dot`, `param_tres`, `param_trestn`,  
`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_gwcs_amp, param_gwecc,
param_gwb_amp, param_dm_sin1yr, param_dm_cos1yr, param_brake,
param_stateSwitchT, param_df1, param_red_sin, param_red_cos,
param_jitter, param_red_dm_sin, param_red_dm_cos, param_band_red_sin,
param_band_red_cos, param_sx, param_sxr1, param_sxr2,
param_sxer, param_group_red_sin, param_group_red_cos, param_ne_sw,
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_cov, constraint_ifunc_x0,
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_red_sin,
constraint_red_cos, constraint_band_red_sin, constraint_band_red_cos, constraint_red_dm_sin,
constraint_red_dm_cos, constraint_group_red_sin, constraint_group_red_cos, constraint_jitter,
constraint_param, 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 [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)
- 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)
- void [updateEpoch\\_str](#) ([pulsar](#) \*psr, int p, const char \*newEpoch)
- void [updateEpoch](#) ([pulsar](#) \*psr, int p, [longdouble](#) nMJD)
- [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, int arr)
- void [updateELL1](#) ([pulsar](#) \*psr, double val, double err, int pos, int arr)
- [longdouble](#) [DDmodel](#) ([pulsar](#) \*psr, int p, int obs, int param)
- void [updateDD](#) ([pulsar](#) \*psr, double val, double err, int pos)
- double [T2model](#) ([pulsar](#) \*psr, int p, int obs, int param, int arr)
- void [updateT2](#) ([pulsar](#) \*psr, double val, double err, int pos, int arr)
- double [T2\\_PTAmodel](#) ([pulsar](#) \*psr, int p, int obs, int param, int arr)
- void [updateT2\\_PTA](#) ([pulsar](#) \*psr, double val, double err, int pos, int arr)
- double [JVmodel](#) ([pulsar](#) \*psr, int p, int obs, int param, int arr)
- void [updateJV](#) ([pulsar](#) \*psr, double val, double err, int pos, int arr)
- double [DDKmodel](#) ([pulsar](#) \*psr, int p, int obs, int param)
- void [updateDDK](#) ([pulsar](#) \*psr, double val, double err, int pos)
- double [DDSmodel](#) ([pulsar](#) \*psr, int p, int obs, int param)
- void [updateDDS](#) ([pulsar](#) \*psr, double val, double err, int pos)
- double [DDGRmodel](#) ([pulsar](#) \*psr, int p, int obs, int param)
- void [updateDDGR](#) ([pulsar](#) \*psr, double val, double err, int pos)
- double [DDHmodel](#) ([pulsar](#) \*psr, int p, int obs, int param)
- void [updateDDH](#) ([pulsar](#) \*psr, double val, double err, int pos)
- double [ELL1Hmodel](#) ([pulsar](#) \*psr, int p, int obs, int param)
- void [updateELL1H](#) ([pulsar](#) \*psr, double val, double err, int pos)
- double [ELL1kmodel](#) ([pulsar](#) \*psr, int p, int obs, int param)
- void [updateELL1k](#) ([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 [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 [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\\_clock\\_path](#) [MAX\_STRLEN]
- char [tempo2\\_plug\\_path](#) [32][MAX\_STRLEN]
- int [tempo2\\_plug\\_path\\_len](#)

### 14.36.1 Detailed Description

contains the main interface to libtempo2.

#### Note

some parts of this to be moved to an internal interface

### 14.36.2 Macro Definition Documentation

#### 14.36.2.1 AU\_DIST

```
#define AU_DIST 1.49598e11
```

1 AU in m

#### 14.36.2.2 AULTSC

```
#define AULTSC 499.00478364
```

Number of light seconds in 1 AU

#### 14.36.2.3 BIG\_G

```
#define BIG_G 6.673e-11
```

Gravitational constant

#### 14.36.2.4 DM\_CONST

```
#define DM_CONST 2.41e-4
```

#### 14.36.2.5 DM\_CONST\_SI

```
#define DM_CONST_SI 7.436e6
```

Dispersion constant in SI units

#### 14.36.2.6 ECLIPTIC\_OBLIQUITY\_VAL

```
#define ECLIPTIC_OBLIQUITY_VAL 84381.4059
```

mean obliquity of ecliptic in arcsec

#### 14.36.2.7 FB90\_TIMEEPH

```
#define FB90_TIMEEPH 2
```

Fairhead & Bretagnon time ephemeris

#### 14.36.2.8 GM

```
#define GM 1.3271243999e20
```

Gravitational constant \* mass sun

#### 14.36.2.9 GM\_C3

```
#define GM_C3 4.925490947e-6
```

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

#### 14.36.2.10 GMJ\_C3

```
#define GMJ_C3 4.70255e-9
```

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

#### 14.36.2.11 GMN\_C3

```
#define GMN_C3 2.54488e-10
```

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

**14.36.2.12 GMS\_C3**

```
#define GMS_C3 1.40797e-9
```

GM\_saturn/c<sup>3</sup> (in seconds)

**14.36.2.13 GMU\_C3**

```
#define GMU_C3 2.14539e-10
```

GM\_uranus/c<sup>3</sup> (in seconds)

**14.36.2.14 GMV\_C3**

```
#define GMV_C3 1.2061e-11
```

GM\_venus/c<sup>3</sup> (in seconds)

**14.36.2.15 HAVE\_GWSIM\_H**

```
#define HAVE_GWSIM_H
```

**14.36.2.16 IF99\_TIMEEPH**

```
#define IF99_TIMEEPH 1
```

Irwin & Fukushima time ephemeris

**14.36.2.17 IFTEPH\_FILE**

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

**14.36.2.18 LEAPSECOND\_FILE**

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

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

**14.36.2.19 MASR2RADS**

```
#define MASR2RADS 1.53628185e-16
```

Converts from mas/yr to rad/s

**14.36.2.20 MAX\_BPJ\_JUMPS**

```
#define MAX_BPJ_JUMPS 5
```

Maximum number of jumps in binary params - for BPJ model

**14.36.2.21 MAX\_CLK\_CORR**

```
#define MAX_CLK_CORR 30
```

Maximum number of steps in the correction to TT

**14.36.2.22 MAX\_CLKCORR**

```
#define MAX_CLKCORR 5000
```

Maximum number of lines in time.dat file

**14.36.2.23 MAX\_COEFF**

```
#define MAX_COEFF 5000
```

Maximum number of coefficients in polyco

**14.36.2.24 MAX\_COMPANIONS**

```
#define MAX_COMPANIONS 4
```

Maximum number of binary companions

**14.36.2.25 MAX\_DM\_DERIVATIVES**

```
#define MAX_DM_DERIVATIVES 10
```

DM0 -> DMn where n=10

**14.36.2.26 MAX\_DMx**

```
#define MAX_DMx 512
```

Max number of DM steps allowed

**14.36.2.27 MAX\_FILELEN**

```
#define MAX_FILELEN 500
```

Maximum filename length



**14.36.2.28 MAX\_FIT**

```
#define MAX_FIT 10000
```

Maximum number of parameters to fit for

**14.36.2.29 MAX\_FLAG\_LEN**

```
#define MAX_FLAG_LEN 32
```

Maximum number of characters in each flag

**14.36.2.30 MAX\_FLAGS**

```
#define MAX_FLAGS 40
```

Maximum number of flags in .tim file/observation

**14.36.2.31 MAX\_FREQ\_DERIVATIVES**

```
#define MAX_FREQ_DERIVATIVES 13
```

F0 -> Fn where n=10

**14.36.2.32 MAX\_IFUNC**

```
#define MAX_IFUNC 1000
```

Maximum number of parameters for interpolation function

**14.36.2.33 MAX\_JUMPS**

```
#define MAX_JUMPS 2000
```

Maximum number of phase jumps

**14.36.2.34 MAX\_LEAPSEC**

```
#define MAX_LEAPSEC 100
```

Maximum number of line in the leap second file

**14.36.2.35 MAX\_MSG**

```
#define MAX_MSG 50
```

Maximum number of different warnings

**14.36.2.36 MAX\_OBSN\_VAL**

```
#define MAX_OBSN_VAL 20000
```

Maximum number of TOAs

**14.36.2.37 MAX\_PARAMS**

```
#define MAX_PARAMS 2000
```

Maximum number of parameters

**14.36.2.38 MAX\_PSR\_VAL**

```
#define MAX_PSR_VAL 40
```

Maximum number of pulsars

**14.36.2.39 MAX\_QUAD**

```
#define MAX_QUAD 150
```

Maximum number of frequency channels in quadrupolar function

**14.36.2.40 MAX\_SITE**

```
#define MAX_SITE 100
```

Maximum number of observatory sites

**14.36.2.41 MAX\_STOREPRECISION**

```
#define MAX_STOREPRECISION 50
```

How many routines in TEMPO2 store precision information

**14.36.2.42 MAX\_STRLEN**

```
#define MAX_STRLEN 1000
```

Maximum length for strings

**14.36.2.43 MAX\_SX**

```
#define MAX_SX 512
```

Max number of Scatter steps allowed

**14.36.2.44 MAX\_T2EFAC**

```
#define MAX_T2EFAC 100
```

Maximum number of T2EFACs allowed

**14.36.2.45 MAX\_T2EQUAD**

```
#define MAX_T2EQUAD 100
```

Maximum number of T2EQUADs allowed

**14.36.2.46 MAX\_TEL\_CLK\_OFFS**

```
#define MAX_TEL_CLK_OFFS 500
```

Maximum number of parameters for telescope clock offset

**14.36.2.47 MAX\_TEL\_DX**

```
#define MAX_TEL_DX 500
```

Maximum number of parameters for interpolation function

**14.36.2.48 MAX\_TEL\_DY**

```
#define MAX_TEL_DY 500
```

Maximum number of parameters for interpolation function

**14.36.2.49 MAX\_TEL\_DZ**

```
#define MAX_TEL_DZ 500
```

Maximum number of parameters for interpolation function

**14.36.2.50 MAX\_TNBN**

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

**14.36.2.51 MAX\_TNDMEv**

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

**14.36.2.52 MAX\_TNECORR**

```
#define MAX_TNECORR 50
```

Maximum number of TNECORRss allowed

**14.36.2.53 MAX\_TNEF**

```
#define MAX_TNEF 50
```

Maximum number of TNEFACs allowed

**14.36.2.54 MAX\_TNEQ**

```
#define MAX_TNEQ 50
```

Maximum number of TNEQUADs allowed

**14.36.2.55 MAX\_TNGN**

```
#define MAX_TNGN 50
```

maximum number of TNGroupNoise parameters allowed

**14.36.2.56 MAX\_TNSQ**

```
#define MAX_TNSQ 50
```

Maximum number of TNEQUADs allowed

**14.36.2.57 MAX\_TOFFSET**

```
#define MAX_TOFFSET 10
```

Number of time jumps allowed in .par file

**14.36.2.58 MAX\_WHITE**

```
#define MAX_WHITE 100
```

Maximum number of parameters for whitening

**14.36.2.59 NE\_SW\_DEFAULT**

```
#define NE_SW_DEFAULT 4
```

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

**14.36.2.60 OBLQ**

```
#define OBLQ 23.445833333333333
```

Obliquity of the ecliptic

**14.36.2.61 OBSSYS\_FILE**

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

Path for file containing Observatory data (obsys.dat)

**14.36.2.62 PCM**

```
#define PCM 3.08568025e16
```

one parsec in meters

**14.36.2.63 SECDAY**

```
#define SECDAY 86400.0
```

Number of seconds in 1 day

**14.36.2.64 SECDAYl**

```
#define SECDAYl longdouble(86400.0)
```

Number of seconds in 1 day

**14.36.2.65 SI\_UNITS**

```
#define SI_UNITS 1
```

New tempo2 mode

**14.36.2.66 SOLAR\_MASS**

```
#define SOLAR_MASS 1.98892e30
```

Mass of Sun (kg)

**14.36.2.67 SOLAR\_RADIUS**

```
#define SOLAR_RADIUS 6.96e8
```

Radius of the Sun (in meters)

**14.36.2.68 SPEED\_LIGHT**

```
#define SPEED_LIGHT 299792458.0
```

Speed of light (m/s)

**14.36.2.69 T2C\_IAU2000B**

```
#define T2C_IAU2000B 1
```

**14.36.2.70 T2C\_TEMPO**

```
#define T2C_TEMPO 2
```

**14.36.2.71 TDB\_UNITS**

```
#define TDB_UNITS 2
```

original tempo mode

**14.36.2.72 TDBTDT\_FILE**

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

Path for file containing TDB-TDT ephemeris

**14.36.2.73 TEMPO2\_h\_HASH**

```
#define TEMPO2_h_HASH "$Id: 0dc79e3a4c8b3cac82106bc030d9b8145e15d6f1 $"
```

**14.36.2.74 TEMPO2\_h\_MAJOR\_VER**

```
#define TEMPO2_h_MAJOR_VER 2018.09
```

**14.36.2.75 TEMPO2\_h\_MINOR\_VER**

```
#define TEMPO2_h_MINOR_VER 1
```

#### 14.36.2.76 TEMPO2\_h\_VER

```
#define TEMPO2_h_VER "2018.09.1"
```

#### 14.36.2.77 TSUN

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

Solar constant for mass calculations.

#### 14.36.2.78 UT1\_FILE

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

Path for the file containing TAI-UT1

### 14.36.3 Typedef Documentation

#### 14.36.3.1 constraint\_label

```
typedef int constraint_label
```

for 'strong typing' - type for enum constraint

#### 14.36.3.2 constraintDerivFunc

```
typedef double(* constraintDerivFunc) (struct pulsar *, int, constraint_label, param_label,  
int, int, void *)
```

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.

#### 14.36.3.3 FitInfo

```
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.

#### 14.36.3.4 FitOutput

```
typedef struct FitOutput FitOutput
```

#### 14.36.3.5 observation

```
typedef struct observation observation
```

A struct containing the details of a single obesrvation.

#### 14.36.3.6 param\_label

```
typedef int param_label
```

for 'strong typing' - type for enum label

#### 14.36.3.7 paramDerivFunc

```
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.

#### 14.36.3.8 parameter

```
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

#### 14.36.3.9 paramUpdateFunc

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

a function used to update the parameters after a fit.



## 14.36.3.10 pulsar

```
typedef struct pulsar pulsar
```

contains the details for a single pulsar.

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

## 14.36.3.11 storePrecision

```
typedef struct storePrecision storePrecision
```

## 14.36.4 Enumeration Type Documentation

## 14.36.4.1 constraint

```
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_cov	
constraint_ifunc_x0	
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	

## Enumerator

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_red_sin	
constraint_red_cos	
constraint_band_red_sin	
constraint_band_red_cos	
constraint_red_dm_sin	
constraint_red_dm_cos	
constraint_group_red_sin	
constraint_group_red_cos	
constraint_jitter	
constraint_param	
constraint_LAST	marker for the last constraint

## 14.36.4.2 label

```
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	

## Enumerator

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_trestn	
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	

## Enumerator

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_gwcs_amp	
param_gwecc	
param_gwb_amp	
param_dm_sin1yr	
param_dm_cos1yr	
param_brake	
param_stateSwitchT	
param_df1	
param_red_sin	
param_red_cos	
param_jitter	
param_red_dm_sin	
param_red_dm_cos	
param_band_red_sin	
param_band_red_cos	
param_sx	
param_sxr1	
param_sxr2	
param_sxer	
param_group_red_sin	
param_group_red_cos	
param_ne_sw	
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

### 14.36.5 Function Documentation

#### 14.36.5.1 allocateMemory()

```
void allocateMemory (
    pulsar * psr,
    int realloc )
```

#### 14.36.5.2 autoConstraints()

```
void autoConstraints (
    pulsar * psr,
    int ipsr,
    int npsr )
```

#### 14.36.5.3 bootstrap()

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

#### 14.36.5.4 BTJmodel()

```
double BTJmodel (
    pulsar * psr,
    int p,
    int obs,
    int param,
    int arr )
```

#### 14.36.5.5 BTmodel()

```
double BTmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

#### 14.36.5.6 BTXmodel()

```
double BTXmodel (
    pulsar * psr,
    int p,
    int obs,
    int param,
    int arr )
```

#### 14.36.5.7 calcRMS()

```
double calcRMS (
    pulsar * psr,
    int p )
```

#### 14.36.5.8 calculate\_bclt()

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

#### 14.36.5.9 compute\_tropospheric\_delays()

```
void compute_tropospheric_delays (
    pulsar * psr,
    int npsr )
```

#### 14.36.5.10 copyParam()

```
void copyParam (
    parameter p1,
    parameter * p2 )
```

#### 14.36.5.11 copyPSR()

```
void copyPSR (
    pulsar * p,
    int p1,
    int p2 )
```

#### 14.36.5.12 CVSdisplayVersion()

```
void CVSdisplayVersion (
    const char * file,
    const char * func,
    const char * verNum )
```

#### 14.36.5.13 DDGRmodel()

```
double DDGRmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

#### 14.36.5.14 DDHmodel()

```
double DDHmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

#### 14.36.5.15 DDKmodel()

```
double DDKmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

#### 14.36.5.16 DDmodel()

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



**14.36.5.17 DDSmodel()**

```
double DDSmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

**14.36.5.18 defineClockCorrectionSequence()**

```
void defineClockCorrectionSequence (
    char * fileList,
    int dispWarnings )
```

**14.36.5.19 destroyMemory()**

```
void destroyMemory (
    pulsar * psr )
```

**14.36.5.20 destroyOne()**

```
void destroyOne (
    pulsar * psr )
```

**14.36.5.21 displayMsg()**

```
void displayMsg (
    int type,
    const char * key,
    const char * searchStr,
    const char * variableStr,
    int noWarnings )
```

**14.36.5.22 displayParameters()**

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

#### 14.36.5.23 dm\_delays()

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

#### 14.36.5.24 dms\_turn()

```
double dms_turn (
    char * line )
```

#### 14.36.5.25 doFitAll()

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

#### 14.36.5.26 dotproduct()

```
double dotproduct (
    double * v1,
    double * v2 )
```

#### 14.36.5.27 ELL1Hmodel()

```
double ELL1Hmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

**14.36.5.28 ELL1kmodel()**

```
double ELL1kmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

**14.36.5.29 ELL1model()**

```
double ELL1model (
    pulsar * psr,
    int p,
    int obs,
    int param,
    int arr )
```

**14.36.5.30 equ2ec1()**

```
void equ2ec1 (
    double * x )
```

**14.36.5.31 formBats()**

```
void formBats (
    pulsar * psr,
    int npsr )
```

**14.36.5.32 formBatsAll()**

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

**14.36.5.33 formResiduals()**

```
void formResiduals (
    pulsar * psr,
    int npsr,
    int removeMean )
```

**14.36.5.34 fortran\_mod()**

```
longdouble fortran_mod (
    longdouble a,
    longdouble p )
```

**14.36.5.35 fortran\_nint()**

```
int fortran_nint (
    double x )
```

**14.36.5.36 fortran\_nlong()**

```
long fortran_nlong (
    longdouble x )
```

**14.36.5.37 get\_EOP()**

```
void get_EOP (
    double mjd,
    double * xp,
    double * yp,
    double * dut1,
    double * dut1dot,
    int dispWarnings,
    char * eopcFile )
```

**14.36.5.38 get\_obsCoord()**

```
void get_obsCoord (
    pulsar * psr,
    int npsr )
```

**14.36.5.39 get\_obsCoord\_IAU2000B()**

```
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] )
```

#### 14.36.5.40 get\_OneobsCoord()

```
void get_OneobsCoord (
    pulsar * psr,
    int npsr,
    int obs )
```

#### 14.36.5.41 getCholeskyMatrix()

```
void getCholeskyMatrix (
    double ** uinv,
    const char * fname,
    pulsar * psr,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc,
    int * ip )
```

#### 14.36.5.42 getClockCorrections()

```
void getClockCorrections (
    observation * obs,
    const char * clockFrom,
    const char * clockTo,
    int warnings )
```

#### 14.36.5.43 getCorrection()

```
double getCorrection (
    observation * obs,
    const char * clockFrom,
    const char * clockTo,
    int warnings )
```

#### 14.36.5.44 getCorrectionTT()

```
double getCorrectionTT (
    observation * obs )
```

#### 14.36.5.45 getInputs()

```
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 )
```

#### 14.36.5.46 getObservatory()

```
observatory* getObservatory (
    char * code )
```

#### 14.36.5.47 getParamDeriv()

```
double getParamDeriv (
    pulsar * psr,
    int ipos,
    double x,
    int i,
    int k )
```

#### 14.36.5.48 getParameterValue()

```
longdouble getParameterValue (
    pulsar * psr,
    int param,
    int arr )
```

**14.36.5.49 hms\_turn()**

```
double hms_turn (
    char * line )
```

**14.36.5.50 id\_residual()**

```
int id_residual (
    float xcurs,
    float ycurs )
```

**14.36.5.51 initialise()**

```
void initialise (
    pulsar * psr,
    int noWarnings )
```

**14.36.5.52 initialiseOne()**

```
void initialiseOne (
    pulsar * psr,
    int noWarnings,
    int fullSetup )
```

**14.36.5.53 JVmodel()**

```
double JVmodel (
    pulsar * psr,
    int p,
    int obs,
    int param,
    int arr )
```

**14.36.5.54 logicFlag()**

```
void logicFlag (
    char * line,
    pulsar * psr,
    int npsr )
```

#### 14.36.5.55 lookup\_observatory\_alias()

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

#### 14.36.5.56 MSSmodel()

```
double MSSmodel (
    pulsar * psr,
    int p,
    int obs,
    int param )
```

#### 14.36.5.57 polyco()

```
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 )
```

#### 14.36.5.58 preProcess()

```
void preProcess (
    pulsar * psr,
    int npsr,
    int argc,
    char * argv[] )
```

#### 14.36.5.59 preProcessSimple()

```
void preProcessSimple (
    pulsar * psr )
```



**14.36.5.60 preProcessSimple1()**

```
void preProcessSimple1 (
    pulsar * psr,
    int tempol,
    double thelast )
```

**14.36.5.61 preProcessSimple2()**

```
void preProcessSimple2 (
    pulsar * psr,
    float startdmmjd,
    int ndm,
    float * dmvals,
    int trimonly )
```

**14.36.5.62 preProcessSimple3()**

```
void preProcessSimple3 (
    pulsar * psr )
```

**14.36.5.63 processFlag()**

```
void processFlag (
    char * line,
    pulsar * psr,
    int npsr )
```

**14.36.5.64 processSimultaneous()**

```
void processSimultaneous (
    char * line,
    pulsar * psr,
    int npsr )
```

**14.36.5.65 readEphemeris()**

```
void readEphemeris (
    pulsar * psr,
    int npsr,
    int addEphemNoise )
```

#### 14.36.5.66 readEphemeris\_calceph()

```
void readEphemeris_calceph (
    pulsar * psr,
    int npsr )
```

#### 14.36.5.67 readJBO\_bat()

```
void readJBO_bat (
    char * fname,
    pulsar * psr,
    int p )
```

#### 14.36.5.68 readObsFile()

```
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] )
```

#### 14.36.5.69 readOneEphemeris()

```
void readOneEphemeris (
    pulsar * psr,
    int npsr,
    int addEphemNoise,
    int obsNumber )
```

#### 14.36.5.70 readParfile()

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

**14.36.5.71 readParfileGlobal()**

```
void readParfileGlobal (
    pulsar * psr,
    int npsr,
    char tpar[MAX_STRLEN][MAX_FILELEN],
    char ttim[MAX_STRLEN][MAX_FILELEN] )
```

**14.36.5.72 readSimpleParfile()**

```
int readSimpleParfile (
    FILE * fin,
    pulsar * p )
```

**14.36.5.73 readTimfile()**

```
void readTimfile (
    pulsar * psr,
    char timFile[][MAX_FILELEN],
    int npsr )
```

**14.36.5.74 recordPrecision()**

```
void recordPrecision (
    pulsar * psr,
    longdouble prec,
    const char * routine,
    const char * comment )
```

**14.36.5.75 secularMotion()**

```
void secularMotion (
    pulsar * psr,
    int npsr )
```

**14.36.5.76 setPlugPath()**

```
void setPlugPath ( )
```

**14.36.5.77 setStart()**

```
float setStart (
    float xcurs,
    float ycurs,
    int flag )
```

**14.36.5.78 setupParameterFileDefaults()**

```
int setupParameterFileDefaults (
    pulsar * p )
```

**14.36.5.79 shapiro\_delay()**

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

**14.36.5.80 simplePlot()**

```
void simplePlot (
    pulsar * psr,
    double unitFlag )
```

**14.36.5.81 solarWindModel()**

```
double solarWindModel (
    pulsar psr,
    int iobs )
```

**14.36.5.82 sortToAs()**

```
void sortToAs (
    pulsar * psr )
```

**14.36.5.83 T2\_PTAmode()**

```
double T2_PTAmode (
    pulsar * psr,
    int p,
    int obs,
    int param,
    int arr )
```

**14.36.5.84 T2model()**

```
double T2model (
    pulsar * psr,
    int p,
    int obs,
    int param,
    int arr )
```

**14.36.5.85 tai2tt()**

```
void tai2tt (
    pulsar * psr,
    int npsr )
```

**14.36.5.86 tai2ut1()**

```
void tai2ut1 (
    pulsar * psr,
    int npsr )
```

**14.36.5.87 textOutput()**

```
void textOutput (
    pulsar * psr,
    int npsr,
    double globalParameter,
    int nGlobal,
    int outRes,
    int newpar,
    const char * fname )
```

**14.36.5.88 toa2utc()**

```
void toa2utc (
    pulsar * psr,
    int npsr )
```

**14.36.5.89 transform\_units()**

```
void transform_units (
    struct pulsar * psr,
    int from,
    int to )
```

**14.36.5.90 tt2tb()**

```
void tt2tb (
    pulsar * psr,
    int npsr )
```

**14.36.5.91 tt2tb\_calceph()**

```
void tt2tb_calceph (
    pulsar * psr,
    int npsr )
```

**14.36.5.92 turn\_deg()**

```
double turn_deg (
    double turn )
```

**14.36.5.93 turn\_dms()**

```
int turn_dms (
    double turn,
    char * dms )
```

**14.36.5.94 turn\_hms()**

```
int turn_hms (
    double turn,
    char * hms )
```

**14.36.5.95 updateBatsAll()**

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

**14.36.5.96 updateBT()**

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

**14.36.5.97 updateBTJ()**

```
void updateBTJ (
    pulsar * psr,
    double val,
    double err,
    int pos,
    int arr )
```

**14.36.5.98 updateBTX()**

```
void updateBTX (
    pulsar * psr,
    double val,
    double err,
    int pos,
    int arr )
```

**14.36.5.99 updateDD()**

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

**14.36.5.100 updateDDGR()**

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

**14.36.5.101 updateDDH()**

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

**14.36.5.102 updateDDK()**

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

**14.36.5.103 updateDDS()**

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



**14.36.5.104 updateELL1()**

```
void updateELL1 (
    pulsar * psr,
    double val,
    double err,
    int pos,
    int arr )
```

**14.36.5.105 updateELL1H()**

```
void updateELL1H (
    pulsar * psr,
    double val,
    double err,
    int pos )
```

**14.36.5.106 updateELL1k()**

```
void updateELL1k (
    pulsar * psr,
    double val,
    double err,
    int pos )
```

**14.36.5.107 updateEpoch()**

```
void updateEpoch (
    pulsar * psr,
    int p,
    longdouble nMJD )
```

**14.36.5.108 updateEpoch\_str()**

```
void updateEpoch_str (
    pulsar * psr,
    int p,
    const char * newEpoch )
```

**14.36.5.109 updateJV()**

```
void updateJV (
    pulsar * psr,
    double val,
    double err,
    int pos,
    int arr )
```

**14.36.5.110 updateMSS()**

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

**14.36.5.111 updateT2()**

```
void updateT2 (
    pulsar * psr,
    double val,
    double err,
    int pos,
    int arr )
```

**14.36.5.112 updateT2\_PTA()**

```
void updateT2_PTA (
    pulsar * psr,
    double val,
    double err,
    int pos,
    int arr )
```

**14.36.5.113 useSelectFile()**

```
void useSelectFile (
    char * fname,
    pulsar * psr,
    int npsr )
```

**14.36.5.114 utc2tai()**

```
void utc2tai (
    pulsar * psr,
    int npsr )
```

**14.36.5.115 vectorPulsar()**

```
void vectorPulsar (
    pulsar * psr,
    int npsr )
```

**14.36.5.116 vectorscale()**

```
void vectorscale (
    double * v,
    double k )
```

**14.36.5.117 vectorsum()**

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

**14.36.5.118 writeTim()**

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

**14.36.5.119 zoom\_graphics()**

```
int zoom_graphics (
    float xcurs2,
    float ycurs2,
    int flag )
```

### 14.36.6 Variable Documentation

#### 14.36.6.1 covarFuncFile

```
char covarFuncFile[MAX_FILELEN]
```

#### 14.36.6.2 dcmFile

```
char dcmFile[MAX_FILELEN]
```

#### 14.36.6.3 displayCVSversion

```
int displayCVSversion
```

Display CVS version

#### 14.36.6.4 ECLIPTIC\_OBLIQUITY

```
double ECLIPTIC_OBLIQUITY
```

#### 14.36.6.5 forceGlobalFit

```
int forceGlobalFit
```

Global = 1 if we are forcing a global fit

#### 14.36.6.6 MAX\_OBSN

```
int MAX_OBSN
```

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

#### 14.36.6.7 MAX\_PSR

```
int MAX_PSR
```

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

#### 14.36.6.8 NEWFIT

```
char NEWFIT
```

global boolean used to enable new fit.

#### Warning

this will be removed in future.

#### 14.36.6.9 tempo2\_clock\_path

```
char tempo2_clock_path[MAX_STRLEN]
```

paths to search for clock files

#### 14.36.6.10 TEMPO2\_ENVIRON

```
char TEMPO2_ENVIRON[ ]
```

TEMPO2 environment variable

#### 14.36.6.11 tempo2\_plug\_path

```
char tempo2_plug_path[32][MAX_STRLEN]
```

paths to search for plugins

#### 14.36.6.12 tempo2\_plug\_path\_len

```
int tempo2_plug_path_len
```

#### 14.36.6.13 tempo2MachineType

```
char tempo2MachineType[MAX_FILELEN]
```

#### 14.36.6.14 veryFast

```
int veryFast
```

Global to run the code fast

## 14.37 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](#)

## 14.37.1 Enumeration Type Documentation

### 14.37.1.1 T2PredictorKind

enum [T2PredictorKind](#)

## Enumerator

NonePredType	
Cheby	
T1	

## 14.37.2 Function Documentation

### 14.37.2.1 T2Predictor\_Copy()

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

### 14.37.2.2 T2Predictor\_Destroy()

```
void T2Predictor_Destroy (
    T2Predictor * t2p )
```

### 14.37.2.3 T2Predictor\_FRead()

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

### 14.37.2.4 T2Predictor\_FWrite()

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

### 14.37.2.5 T2Predictor\_GetEndFreq()

```
long double T2Predictor_GetEndFreq (
    T2Predictor * t2p )
```

#### 14.37.2.6 T2Predictor\_GetEndMJD()

```
long double T2Predictor_GetEndMJD (
    T2Predictor * t2p )
```

#### 14.37.2.7 T2Predictor\_GetFrequency()

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

#### 14.37.2.8 T2Predictor\_GetPhase()

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

#### 14.37.2.9 T2Predictor\_GetPlan()

```
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 )
```

#### 14.37.2.10 T2Predictor\_GetPlan\_Ext()

```
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 )
```



**14.37.2.11 T2Predictor\_GetPSRName()**

```
char* T2Predictor_GetPSRName (
    T2Predictor * t2p )
```

**14.37.2.12 T2Predictor\_GetSiteName()**

```
char* T2Predictor_GetSiteName (
    T2Predictor * t2p )
```

**14.37.2.13 T2Predictor\_GetStartFreq()**

```
long double T2Predictor_GetStartFreq (
    T2Predictor * t2p )
```

**14.37.2.14 T2Predictor\_GetStartMJD()**

```
long double T2Predictor_GetStartMJD (
    T2Predictor * t2p )
```

**14.37.2.15 T2Predictor\_Init()**

```
void T2Predictor_Init (
    T2Predictor * t2p )
```

**14.37.2.16 T2Predictor\_Insert()**

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

**14.37.2.17 T2Predictor\_Keep()**

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

#### 14.37.2.18 T2Predictor\_Kind()

```
T2PredictorKind T2Predictor_Kind (
    T2Predictor * t2p )
```

#### 14.37.2.19 T2Predictor\_Read()

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

#### 14.37.2.20 T2Predictor\_Write()

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

### 14.37.3 Variable Documentation

#### 14.37.3.1 ChebyModelSet\_OutOfRange

```
int ChebyModelSet_OutOfRange
```

## 14.38 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)

### 14.38.1 Function Documentation

#### 14.38.1.1 Cheby2D\_Construct()

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

#### 14.38.1.2 Cheby2D\_Construct\_x\_Derivative()

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

#### 14.38.1.3 Cheby2D\_Test()

```
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 )
```

#### 14.38.1.4 ChebyModel\_Construct()

```
void ChebyModel_Construct (
    ChebyModel * cm,
    const pulsar * psr )
```

#### 14.38.1.5 ChebyModel\_Copy()

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

#### 14.38.1.6 ChebyModel\_Destroy()

```
void ChebyModel_Destroy (
    ChebyModel * cm )
```

#### 14.38.1.7 ChebyModel\_GetFrequency()

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

#### 14.38.1.8 ChebyModel\_GetPhase()

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

#### 14.38.1.9 ChebyModel\_Init()

```
void ChebyModel_Init (
    ChebyModel * cmodel,
    int nmjdcoeff,
    int nfreqcoeff )
```

#### 14.38.1.10 ChebyModel\_Read()

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

#### 14.38.1.11 ChebyModel\_Test()

```
void ChebyModel_Test (
    ChebyModel * cm,
    const pulsar * psr,
    int nmjd,
    int nfreq,
    long double * residualRMS,
    long double * residualMAV )
```

#### 14.38.1.12 ChebyModel\_Write()

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

#### 14.38.1.13 ChebyModelSet\_Construct()

```
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 )
```

#### 14.38.1.14 ChebyModelSet\_Destroy()

```
void ChebyModelSet_Destroy (
    ChebyModelSet * cms )
```

#### 14.38.1.15 ChebyModelSet\_GetFrequency()

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

#### 14.38.1.16 ChebyModelSet\_GetNearest()

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

#### 14.38.1.17 ChebyModelSet\_GetPhase()

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

#### 14.38.1.18 ChebyModelSet\_Init()

```
void ChebyModelSet_Init (
    ChebyModelSet * cms )
```

#### 14.38.1.19 ChebyModelSet\_Insert()

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

#### 14.38.1.20 ChebyModelSet\_Keep()

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

#### 14.38.1.21 ChebyModelSet\_Read()

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

#### 14.38.1.22 ChebyModelSet\_Test()

```
void ChebyModelSet_Test (
    ChebyModelSet * cms,
    const pulsar * psr,
    int nmjd,
    int nfreq,
    long double * residualRMS,
    long double * residualMAV )
```

#### 14.38.1.23 ChebyModelSet\_Write()

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

**14.38.1.24 T1Polyco\_GetFrequency()**

```
long double T1Polyco_GetFrequency (
    const T1Polyco * t1p,
    long double mjd,
    long double freq )
```

**14.38.1.25 T1Polyco\_GetPhase()**

```
long double T1Polyco_GetPhase (
    const T1Polyco * t1p,
    long double mjd,
    long double freq )
```

**14.38.1.26 T1Polyco\_Read()**

```
int T1Polyco_Read (
    T1Polyco * t1p,
    FILE * f )
```

**14.38.1.27 T1Polyco\_Write()**

```
void T1Polyco_Write (
    const T1Polyco * t1p,
    FILE * f )
```

**14.38.1.28 T1PolycoSet\_Destroy()**

```
void T1PolycoSet_Destroy (
    T1PolycoSet * t1ps )
```

**14.38.1.29 T1PolycoSet\_GetFrequency()**

```
long double T1PolycoSet_GetFrequency (
    const T1PolycoSet * t1ps,
    long double mjd,
    long double freq )
```



#### 14.38.1.30 T1PolycoSet\_GetNearest()

```
T1Polyco* T1PolycoSet_GetNearest (
    long double mjd )
```

#### 14.38.1.31 T1PolycoSet\_GetPhase()

```
long double T1PolycoSet_GetPhase (
    const T1PolycoSet * tlps,
    long double mjd,
    long double freq )
```

#### 14.38.1.32 T1PolycoSet\_Read()

```
int T1PolycoSet_Read (
    T1PolycoSet * tlps,
    FILE * f )
```

#### 14.38.1.33 T1PolycoSet\_Write()

```
void T1PolycoSet_Write (
    const T1PolycoSet * tlps,
    FILE * f )
```

## 14.39 tempo2Util.h File Reference

### Functions

- double [turn\\_deg](#) (double turn)
- double [dms\\_turn](#) (char \*line)
- double [hms\\_turn](#) (char \*line)

### 14.39.1 Function Documentation

#### 14.39.1.1 dms\_turn()

```
double dms_turn (
    char * line )
```

### 14.39.1.2 hms\_turn()

```
double hms_turn (
    char * line )
```

### 14.39.1.3 turn\_deg()

```
double turn_deg (
    double turn )
```

## 14.40 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)

### 14.40.1 Function Documentation

#### 14.40.1.1 cholesky\_covarFunc2matrix()

```
void cholesky_covarFunc2matrix (
    double ** m,
    double * covarFunc,
    int ndays,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc )
```

#### 14.40.1.2 cholesky\_dmModel()

```
void cholesky_dmModel (
    double ** m,
    double D,
    double d,
    double ref_freq,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc )
```

#### 14.40.1.3 cholesky\_dmModelCovarParam()

```
void cholesky_dmModelCovarParam (
    double ** m,
    double alpha,
    double a,
    double b,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc )
```

#### 14.40.1.4 cholesky\_ecm()

```
void cholesky_ecm (
    double ** m,
    char * fileName,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc )
```

#### 14.40.1.5 cholesky\_formUinv()

```
int cholesky_formUinv (
    double ** uinv,
    double ** m,
    int np )
```

#### 14.40.1.6 cholesky\_powerlawModel()

```
void cholesky_powerlawModel (
    double ** m,
    double modelAlpha,
    double modelFc,
    double modelA,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc )
```

#### 14.40.1.7 cholesky\_powerlawModel\_withBeta()

```
void cholesky_powerlawModel_withBeta (
    double ** m,
    double modelAlpha,
    double beta,
    double modelFc,
    double modelA,
    double * resx,
    double * resy,
    double * rese,
    int np,
    int nc )
```

#### 14.40.1.8 cholesky\_readFromCovarianceFunction()

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

### 14.41 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)
- double [TKrobustDefConstrainedLeastSquares](#) (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, double \*constraint\_vals)
- 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)

### 14.41.1 Function Documentation

#### 14.41.1.1 TKconstrainedLeastSquares()

```
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 )
```

#### 14.41.1.2 TKfindPoly\_d()

```
void TKfindPoly_d (
    double * px,
    double * py,
    int n,
    int m,
    double * p )
```

#### 14.41.1.3 TKfitPoly()

```
void TKfitPoly (
    double x,
    double * v,
    int m )
```

#### 14.41.1.4 TKleastSquares()

```
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 )
```

#### 14.41.1.5 TKleastSquares\_svd()

```
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 )
```

#### 14.41.1.6 TKleastSquares\_svd\_noErr()

```
void TKleastSquares_svd_noErr (
    double * x,
    double * y,
    int n,
    double * p,
    int nf,
    void(*) (double, double [], int) fitFuncs )
```

#### 14.41.1.7 TKremovePoly\_d()

```
void TKremovePoly_d (
    double * px,
    double * py,
    int n,
    int m )
```

#### 14.41.1.8 TKremovePoly\_f()

```
void TKremovePoly_f (
    float * px,
    float * py,
    int n,
    int m )
```

#### 14.41.1.9 TKrobustConstrainedLeastSquares()

```
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 )
```

#### 14.41.1.10 TKrobustDefConstrainedLeastSquares()

```
double TKrobustDefConstrainedLeastSquares (
    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,
double * constraint_vals )
```

#### 14.41.1.11 TKrobustLeastSquares()

```
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 )
```

## 14.42 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 24`
- `#define TK_MAX_ERROR_LEN 200`
- `#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 logall(_fmt, ...) _LOG(WHERESTR _fmt ENDL, WHEREARG, ## __VA_ARGS__)`
- `#define logmsg(_fmt, ...) if(!quietFlag)logall(_fmt, ## __VA_ARGS__)`
- `#define logdbg(_fmt, ...) if(debugFlag)logall(_fmt, ## __VA_ARGS__)`



- `#define logerr(_fmt, ...) do{TK_STORE_ERROR(_fmt,##__VA_ARGS__); _LOG(WHEREERR _fmt 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)snprintf(TK_errorlog[TK_errorCount],TK_MAX_ERROR_LEN,_fmt,##__VA_ARGS__); ++TK_errorCount`
- `#define TK_STORE_WARNING(_fmt, ...) if(TK_warnCount < TK_MAX_ERRORS)snprintf(TK_warnlog[TK_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 quietFlag`
- `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]`

### 14.42.1 Macro Definition Documentation

#### 14.42.1.1 \_LOG

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

#### 14.42.1.2 BOLDCOLOR

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

#### 14.42.1.3 DEPRECATED

```
#define DEPRECATED
```

#### 14.42.1.4 ENDERR

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

#### 14.42.1.5 ENDL

```
#define ENDL "\n"
```

#### 14.42.1.6 ERRORCOLOR

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

#### 14.42.1.7 LOG\_OUTFILE

```
#define LOG_OUTFILE stdout
```

#### 14.42.1.8 logall

```
#define logall(  
    _fmt,  
    ... ) __LOG(WHERESTR _fmt ENDL, WHEREARG, ##__VA_ARGS__)
```

#### 14.42.1.9 logdbg

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

#### 14.42.1.10 logerr

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

#### 14.42.1.11 logmsg

```
#define logmsg(  
    _fmt,  
    ... ) if(!quietFlag) logall(_fmt,##__VA_ARGS__)
```

#### 14.42.1.12 logtchk

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

#### 14.42.1.13 logwarn

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

#### 14.42.1.14 RESETCOLOR

```
#define RESETCOLOR "\033[0m"
```

#### 14.42.1.15 TK\_MAX\_ERROR\_LEN

```
#define TK_MAX_ERROR_LEN 200
```

#### 14.42.1.16 TK\_MAX\_ERRORS

```
#define TK_MAX_ERRORS 24
```

#### 14.42.1.17 TK\_STORE\_ERROR

```
#define TK_STORE_ERROR(  
    _fmt,  
    ... ) if (TK_errorCount < TK_MAX_ERRORS) snprintf(TK_errorlog[TK_errorCount], TK_MAX_ERROR_LEN,  
    _fmt, ##__VA_ARGS__); ++TK_errorCount
```

#### 14.42.1.18 TK\_STORE\_WARNING

```
#define TK_STORE_WARNING(  
    _fmt,  
    ... ) if (TK_warnCount < TK_MAX_ERRORS) snprintf(TK_warnlog[TK_warnCount], TK_MAX_ERROR_LEN,  
    _fmt, ##__VA_ARGS__); ++TK_warnCount
```

#### 14.42.1.19 WARNCOLOR

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

#### 14.42.1.20 WHEREARG

```
#define WHEREARG __FILE__, __LINE__
```

#### 14.42.1.21 WHEREERR

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

#### 14.42.1.22 WHERESTR

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

#### 14.42.1.23 WHERECHK

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

#### 14.42.1.24 WHEREWARN

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

### 14.42.2 Function Documentation

#### 14.42.2.1 \_TKchklog()

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

#### 14.42.2.2 logerr\_check()

```
int logerr_check ( )
```

### 14.42.3 Variable Documentation

#### 14.42.3.1 debugFlag

```
int debugFlag
```

#### 14.42.3.2 quietFlag

```
int quietFlag
```

#### 14.42.3.3 tcheck

```
int tcheck
```

#### 14.42.3.4 timer\_clk

```
clock_t timer_clk
```

#### 14.42.3.5 TK\_errorCount

```
unsigned TK_errorCount
```

#### 14.42.3.6 TK\_errorlog

```
char TK_errorlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]
```

#### 14.42.3.7 TK\_warnCount

```
unsigned TK_warnCount
```

#### 14.42.3.8 TK\_warnlog

```
char TK_warnlog[TK_MAX_ERRORS][TK_MAX_ERROR_LEN]
```

#### 14.42.3.9 writeResiduals

```
int writeResiduals
```

### 14.43 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,...)

### 14.43.1 Macro Definition Documentation

#### 14.43.1.1 cosl

```
#define cosl cosq
```

#### 14.43.1.2 fabsl

```
#define fabsl fabsq
```

#### 14.43.1.3 floorl

```
#define floorl floorq
```

#### 14.43.1.4 FMT\_LD

```
#define FMT_LD "Q"
```

#### 14.43.1.5 LD\_PI

```
#define LD_PI M_PIq
```

#### 14.43.1.6 longdouble

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

#### 14.43.1.7 LONGDOUBLE\_IS\_FLOAT128

```
#define LONGDOUBLE_IS_FLOAT128
```

#### 14.43.1.8 LONGDOUBLE\_ONE

```
#define LONGDOUBLE_ONE 1.0Q
```

#### 14.43.1.9 powl

```
#define powl powq
```

#### 14.43.1.10 sinl

```
#define sinl sinq
```

#### 14.43.1.11 USE\_BUILTIN\_LONGDOUBLE

```
#define USE_BUILTIN_LONGDOUBLE
```



## 14.43.2 Typedef Documentation

### 14.43.2.1 longdouble

```
typedef __float128 longdouble
```

## 14.43.3 Function Documentation

### 14.43.3.1 ld\_fprintf()

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

### 14.43.3.2 ld\_printf()

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

### 14.43.3.3 ld\_sprintf()

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

### 14.43.3.4 parse\_longdouble()

```
longdouble parse_longdouble (
    const char * str )
```

## 14.44 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)`

#### 14.44.1 Macro Definition Documentation

##### 14.44.1.1 ld\_fprintf

```
#define ld_fprintf fprintf
```

##### 14.44.1.2 LD\_PI

```
#define LD_PI M_PI
```

##### 14.44.1.3 ld\_printf

```
#define ld_printf printf
```

#### 14.44.1.4 ld\_sprintf

```
#define ld_sprintf sprintf
```

#### 14.44.1.5 longdouble

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

#### 14.44.1.6 LONGDOUBLE\_IS\_IEEE754

```
#define LONGDOUBLE_IS_IEEE754
```

#### 14.44.1.7 LONGDOUBLE\_ONE

```
#define LONGDOUBLE_ONE 1.0L
```

#### 14.44.1.8 USE\_BUILTIN\_LONGDOUBLE

```
#define USE_BUILTIN_LONGDOUBLE
```

### 14.44.2 Typedef Documentation

#### 14.44.2.1 longdouble

```
typedef long double longdouble
```

### 14.44.3 Function Documentation

#### 14.44.3.1 parse\_longdouble()

```
longdouble parse_longdouble (  
    const char * str )
```

## 14.45 TKlongdouble.ld.h File Reference

```
#include <math.h>
```

Include dependency graph for TKlongdouble.ld.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)`

#### 14.45.1 Macro Definition Documentation

##### 14.45.1.1 ld\_fprintf

```
#define ld_fprintf fprintf
```

##### 14.45.1.2 LD\_PI

```
#define LD_PI M_PI
```

##### 14.45.1.3 ld\_printf

```
#define ld_printf printf
```

#### 14.45.1.4 ld\_sprintf

```
#define ld_sprintf sprintf
```

#### 14.45.1.5 longdouble

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

#### 14.45.1.6 LONGDOUBLE\_IS\_IEEE754

```
#define LONGDOUBLE_IS_IEEE754
```

#### 14.45.1.7 LONGDOUBLE\_ONE

```
#define LONGDOUBLE_ONE 1.0L
```

#### 14.45.1.8 USE\_BUILTIN\_LONGDOUBLE

```
#define USE_BUILTIN_LONGDOUBLE
```

### 14.45.2 Typedef Documentation

#### 14.45.2.1 longdouble

```
typedef long double longdouble
```

### 14.45.3 Function Documentation

#### 14.45.3.1 parse\_longdouble()

```
longdouble parse_longdouble (  
    const char * str )
```

## 14.46 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)

### 14.46.1 Function Documentation

#### 14.46.1.1 [free\\_2df\(\)](#)

```
void free_2df (
    float ** uinv )
```

#### 14.46.1.2 [free\\_blas\(\)](#)

```
void free_blas (
    double ** matrix )
```

#### 14.46.1.3 [free\\_uinv\(\)](#)

```
void free_uinv (
    double ** uinv )
```

#### 14.46.1.4 get\_blas\_cols()

```
int get_blas_cols (
    double ** uinv )
```

#### 14.46.1.5 get\_blas\_rows()

```
int get_blas_rows (
    double ** uinv )
```

#### 14.46.1.6 malloc\_2df()

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

#### 14.46.1.7 malloc\_blas()

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

#### 14.46.1.8 malloc\_uinv()

```
double** malloc_uinv (
    int n )
```

#### 14.46.1.9 TKmultMatrix()

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

#### 14.46.1.10 TKmultMatrix\_sq()

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

#### 14.46.1.11 TKmultMatrixVec()

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

#### 14.46.1.12 TKmultMatrixVec\_sq()

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

## 14.47 TKrobust.h File Reference

### Functions

- double [TKrobust](#) (double \*data, double \*white\_data, double \*\*designMatrix, double \*\*white\_designMatrix, double \*\*constraintsMatrix, int ndata, int nparams, int nconstraints, double tol, char rescale\_errors, double \*outP, double \*e, double \*\*Ocvm, char robust, double \*constraint\_vals)

#### 14.47.1 Function Documentation



## 14.47.1.1 TKrobust()

```
double TKrobust (
    double * data,
    double * white_data,
    double ** designMatrix,
    double ** white_designMatrix,
    double ** constraintsMatrix,
    int ndata,
    int nparams,
    int nconstraints,
    double tol,
    char rescale_errors,
    double * outP,
    double * e,
    double ** OcvM,
    char robust,
    double * constraint_vals )
```

## 14.48 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)
- int [calcSpectraErr\\_complex](#) (double \*\*uinv, double \*resx, double \*resy, int nres, double \*specX, double \*specR, double \*specI, 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](#)

## 14.48.1 Macro Definition Documentation

### 14.48.1.1 ABS

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

### 14.48.1.2 MAX

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

### 14.48.1.3 MIN

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

## 14.48.2 Typedef Documentation

### 14.48.2.1 complexVal

```
typedef struct complexVal complexVal
```

## 14.48.3 Function Documentation

### 14.48.3.1 calcSpectra()

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

### 14.48.3.2 calcSpectraErr()

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

### 14.48.3.3 calcSpectraErr\_complex()

```
int calcSpectraErr_complex (  
    double ** uinv,  
    double * resx,  
    double * resy,  
    int nres,  
    double * specX,  
    double * specR,  
    double * specI,  
    double * specE,  
    int nfit )
```

#### 14.48.3.4 fit4()

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

#### 14.48.3.5 getprtj()

```
void getprtj (
    int n )
```

#### 14.48.3.6 getweights()

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

#### 14.48.3.7 indexx8()

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

#### 14.48.3.8 mat20()

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

#### 14.48.3.9 sineFunc()

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

#### 14.48.3.10 TK\_dft()

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

#### 14.48.3.11 TK\_fft()

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

#### 14.48.3.12 TK\_fitSine()

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

#### 14.48.3.13 TK\_fitSinusoids()

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

#### 14.48.3.14 TK\_weightLS()

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

#### 14.48.3.15 TKaveragePts()

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

#### 14.48.3.16 TKboxcar()

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

**14.48.3.17 TKcmonot()**

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

**14.48.3.18 TKfirstDifference()**

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

**14.48.3.19 TKhann()**

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

**14.48.3.20 TKinterpolateSplineSmoothFixedXPts()**

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

**14.48.3.21 TKlomb\_d()**

```
void TKlomb_d (
    double * x,
    double * y,
    int n,
    double ofac,
    double hifac,
    double * ox,
    double * oy,
    int * outN,
    double * var )
```

#### 14.48.3.22 TKsortit()

```
void TKsortit (
    double * x,
    double * y,
    int n )
```

#### 14.48.3.23 TKspectrum()

```
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 )
```

#### 14.48.3.24 TKspline\_interpolate()

```
void TKspline_interpolate (
    int n,
    double * x,
    double * y,
    double yd[ ][4],
    double * interpX,
    double * interpY,
    int nInterp )
```

### 14.48.4 Variable Documentation



## 14.48.4.1 verbose\_calc\_spectra

```
bool verbose_calc_spectra
```

## 14.49 TKsvd.h File Reference

## Functions

- void [TKsingularValueDecomposition\\_lsq](#) (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)

## 14.49.1 Function Documentation

## 14.49.1.1 TKbacksubstitution\_svd()

```
void TKbacksubstitution_svd (
    longdouble ** V,
    longdouble * w,
    longdouble ** U,
    longdouble * b,
    longdouble * x,
    int n,
    int nf )
```

## 14.49.1.2 TKbidiagonal()

```
void TKbidiagonal (
    longdouble ** a,
    longdouble * anorm,
    int ndata,
    int nfit,
    longdouble ** v,
    longdouble * w,
    longdouble ** u,
    longdouble * rv1 )
```

#### 14.49.1.3 TKpythag()

```
longdouble TKpythag (
    longdouble a,
    longdouble b )
```

#### 14.49.1.4 TKsingularValueDecomposition\_lsqr()

```
void TKsingularValueDecomposition_lsqr (
    longdouble ** designMatrix,
    int n,
    int nf,
    longdouble ** v,
    longdouble * w,
    longdouble ** u )
```

# Index

- `_DARWIN_USE_64_BIT_INODE`  
config.h, 122
  - `_LOG`  
TKlog.h, 253
  - `_TKchklog`  
TKlog.h, 257
- ABS
  - TKspectrum.h, 270
- ACCEL\_LSQ
  - T2accel.h, 159
- ACCEL\_MULTMATRIX
  - T2accel.h, 159
- ACCEL\_UINV
  - T2accel.h, 159
- aSize
  - parameter, 66
- AU\_DIST
  - tempo2.h, 193
- AULTSC
  - tempo2.h, 193
- accel\_lsqr
  - T2accel.h, 159
- accel\_multMatrix
  - T2accel.h, 159
- accel\_multMatrixVec
  - T2accel.h, 159
- accel\_uinv
  - T2accel.h, 160
- across\_g
  - gwSrc, 45
  - gwgeneralSrc, 39
- across\_im\_g
  - gwSrc, 45
  - gwgeneralSrc, 39
- addTNGlobalEQ
  - pulsar, 74
- addedNoise
  - observation, 53
- allocateMemory
  - tempo2.h, 210
- aplus\_g
  - gwSrc, 45
  - gwgeneralSrc, 40
- aplus\_im\_g
  - gwSrc, 45
  - gwgeneralSrc, 40
- asl\_g
  - gwgeneralSrc, 40
- asl\_im\_g
  - gwgeneralSrc, 40
- ast\_g
  - gwgeneralSrc, 40
- ast\_im\_g
  - gwgeneralSrc, 40
- au
  - jpl\_eph\_data, 48
- auto\_constraints
  - pulsar, 74
- autoConstraints
  - tempo2.h, 210
- autosetDMCM
  - constraints.h, 127
- AverageDMResiduals
  - pulsar, 74
- AverageEpochWidth
  - pulsar, 74
- AverageFlag
  - pulsar, 74
- AverageResiduals
  - pulsar, 74
- averagebat
  - observation, 53
- averagedmbat
  - observation, 53
- averagedmerr
  - observation, 53
- averagedmres
  - observation, 53
- averageerr
  - observation, 53
- averageres
  - observation, 54
- avx\_g
  - gwgeneralSrc, 40
- avx\_im\_g
  - gwgeneralSrc, 40
- avy\_g
  - gwgeneralSrc, 41
- avy\_im\_g
  - gwgeneralSrc, 41
- BIG\_G
  - tempo2.h, 193
- BOLDCOLOR
  - TKlog.h, 253
- BTJmodel
  - tempo2.h, 210
- BTXmodel
  - tempo2.h, 210

- BTmodel
  - tempo2.h, [210](#)
- bat
  - observation, [54](#)
- batCorr
  - observation, [54](#)
- bbat
  - observation, [54](#)
- binary\_frequency
  - T1Polyco, [112](#)
- binary\_phase
  - T1Polyco, [112](#)
- binaryModel
  - pulsar, [74](#)
- bline
  - observation, [54](#)
- bootStrap
  - pulsar, [75](#)
- bootstrap
  - tempo2.h, [210](#)
- brace
  - pulsar, [75](#)
- c\_fileptr
  - read\_fortran.h, [156](#)
- c\_fileptr2
  - read\_fortran2.h, [158](#)
- CONSTRAINTfuncs
  - constraints.h, [130](#)
- CVSdisplayVersion
  - tempo2.h, [211](#)
- cache
  - jpl\_eph\_data, [49](#)
- calcRMS
  - tempo2.h, [211](#)
- calcShapiro
  - pulsar, [75](#)
- calcSpectra
  - TKspectrum.h, [271](#)
- calcSpectraErr
  - TKspectrum.h, [271](#)
- calcSpectraErr\_complex
  - TKspectrum.h, [271](#)
- calculate\_bclt
  - tempo2.h, [211](#)
- calculateResidualGW
  - GWsim.h, [138](#)
- calculateResidualgeneralGW
  - GWsim.h, [138](#)
- cgw\_angpol
  - pulsar, [75](#)
- cgw\_cosinc
  - pulsar, [75](#)
- cgw\_h0
  - pulsar, [75](#)
- cgw\_mc
  - pulsar, [75](#)
- cheby
  - ChebyModel, [30](#)
  - T2Predictor, [115](#)
- Cheby2D\_Construct
  - tempo2pred\_int.h, [239](#)
- Cheby2D\_Construct\_x\_Derivative
  - tempo2pred\_int.h, [239](#)
- Cheby2D\_Test
  - tempo2pred\_int.h, [240](#)
- Cheby2D, [29](#)
  - coeff, [29](#)
  - nx, [29](#)
  - ny, [29](#)
- ChebyModel, [30](#)
  - cheby, [30](#)
  - dispersion\_constant, [30](#)
  - freq\_end, [30](#)
  - freq\_start, [30](#)
  - frequency\_cheby, [30](#)
  - mjd\_end, [31](#)
  - mjd\_start, [31](#)
  - psrname, [31](#)
  - sitename, [31](#)
- ChebyModel\_Construct
  - tempo2pred\_int.h, [240](#)
- ChebyModel\_Copy
  - tempo2pred\_int.h, [240](#)
- ChebyModel\_Destroy
  - tempo2pred\_int.h, [240](#)
- ChebyModel\_GetFrequency
  - tempo2pred\_int.h, [240](#)
- ChebyModel\_GetPhase
  - tempo2pred\_int.h, [240](#)
- ChebyModel\_Init
  - tempo2pred\_int.h, [241](#)
- ChebyModel\_Read
  - tempo2pred\_int.h, [241](#)
- ChebyModel\_Test
  - tempo2pred\_int.h, [241](#)
- ChebyModel\_Write
  - tempo2pred\_int.h, [241](#)
- ChebyModelSet, [31](#)
  - nsegments, [32](#)
  - segments, [32](#)
- ChebyModelSet\_Construct
  - tempo2pred\_int.h, [241](#)
- ChebyModelSet\_Destroy
  - tempo2pred\_int.h, [242](#)
- ChebyModelSet\_GetFrequency
  - tempo2pred\_int.h, [242](#)
- ChebyModelSet\_GetNearest
  - tempo2pred\_int.h, [242](#)
- ChebyModelSet\_GetPhase
  - tempo2pred\_int.h, [242](#)
- ChebyModelSet\_Init
  - tempo2pred\_int.h, [242](#)
- ChebyModelSet\_Insert
  - tempo2pred\_int.h, [243](#)
- ChebyModelSet\_Keep
  - tempo2pred\_int.h, [243](#)

- ChebyModelSet\_OutOfRange
  - tempo2pred.h, [238](#)
- ChebyModelSet\_Read
  - tempo2pred\_int.h, [243](#)
- ChebyModelSet\_Test
  - tempo2pred\_int.h, [243](#)
- ChebyModelSet\_Write
  - tempo2pred\_int.h, [243](#)
- chisq
  - observation, [54](#)
- cholesky.h, [119](#)
  - cholesky\_covarFunc2matrix, [119](#)
  - cholesky\_dmModel, [119](#)
  - cholesky\_dmModelCovarParam, [120](#)
  - cholesky\_ecm, [120](#)
  - cholesky\_formUinv, [120](#)
  - cholesky\_powerlawModel, [120](#)
  - cholesky\_powerlawModel\_withBeta, [121](#)
  - cholesky\_readFromCovarianceFunction, [121](#)
- cholesky\_covarFunc2matrix
  - cholesky.h, [119](#)
  - TKcholesky.h, [246](#)
- cholesky\_dmModel
  - cholesky.h, [119](#)
  - TKcholesky.h, [246](#)
- cholesky\_dmModelCovarParam
  - cholesky.h, [120](#)
  - TKcholesky.h, [247](#)
- cholesky\_ecm
  - cholesky.h, [120](#)
  - TKcholesky.h, [247](#)
- cholesky\_formUinv
  - cholesky.h, [120](#)
  - TKcholesky.h, [247](#)
- cholesky\_powerlawModel
  - cholesky.h, [120](#)
  - TKcholesky.h, [247](#)
- cholesky\_powerlawModel\_withBeta
  - cholesky.h, [121](#)
  - TKcholesky.h, [248](#)
- cholesky\_readFromCovarianceFunction
  - cholesky.h, [121](#)
  - TKcholesky.h, [248](#)
- choleskyRoutines.h, [121](#)
- clk\_offsE
  - pulsar, [76](#)
- clk\_offsT
  - pulsar, [76](#)
- clk\_offsV
  - pulsar, [76](#)
- clkOffsN
  - pulsar, [76](#)
- clock
  - pulsar, [76](#)
- clock\_correction, [32](#)
  - correction, [32](#)
  - corrects\_to, [32](#)
- clock\_name
  - observatory, [64](#)
- clockCorr
  - observation, [54](#)
- clockFromOverride
  - pulsar, [76](#)
- close\_file
  - read\_fortran.h, [155](#)
- close\_file2
  - read\_fortran2.h, [157](#)
- code
  - observatory, [64](#)
- coeff
  - Cheby2D, [29](#)
  - T1Polyco, [112](#)
- comment
  - storePrecision, [111](#)
- complexVal, [33](#)
  - imag, [33](#)
  - real, [33](#)
  - TKspectrum.h, [271](#)
- compute\_tropospheric\_delays
  - tempo2.h, [211](#)
- computeConstraintWeights
  - constraints.h, [128](#)
- config.h, [121](#)
  - \_DARWIN\_USE\_64\_BIT\_INODE, [122](#)
  - F77\_FUNC\_, [123](#)
  - F77\_FUNC, [122](#)
  - HAVE\_BLAS, [123](#)
  - HAVE\_CFITSIO, [123](#)
  - HAVE\_DLERROR, [123](#)
  - HAVE\_DLFCN\_H, [123](#)
  - HAVE\_FFTW3, [123](#)
  - HAVE\_INTTYPES\_H, [123](#)
  - HAVE\_LAPACK, [124](#)
  - HAVE\_LIBDLLOADER, [124](#)
  - HAVE\_LIBDL, [124](#)
  - HAVE\_LIBM, [124](#)
  - HAVE\_MEMORY\_H, [124](#)
  - HAVE\_PGPLOT, [124](#)
  - HAVE\_PTHREAD, [124](#)
  - HAVE\_STDINT\_H, [124](#)
  - HAVE\_STDLIB\_H, [125](#)
  - HAVE\_STRING\_H, [125](#)
  - HAVE\_STRINGS\_H, [125](#)
  - HAVE\_SYS\_STAT\_H, [125](#)
  - HAVE\_SYS\_TYPES\_H, [125](#)
  - HAVE\_UNISTD\_H, [125](#)
  - LT\_OBJDIR, [125](#)
  - PACKAGE\_BUGREPORT, [126](#)
  - PACKAGE\_NAME, [126](#)
  - PACKAGE\_STRING, [126](#)
  - PACKAGE\_TARNAME, [126](#)
  - PACKAGE\_URL, [126](#)
  - PACKAGE\_VERSION, [126](#)
  - PACKAGE, [125](#)
  - QR\_DEFAULT, [126](#)
  - STDC\_HEADERS, [126](#)

- TEMPO2\_ARCH, 127
- VERSION, 127
- consFunc\_dmmodel\_cw
  - constraints.h, 128
- consFunc\_dmmodel\_cw\_year
  - constraints.h, 128
- consFunc\_dmmodel\_dm1
  - constraints.h, 128
- consFunc\_dmmodel\_mean
  - constraints.h, 128
- consFunc\_ifunc
  - constraints.h, 129
- consFunc\_ifunc\_year
  - constraints.h, 129
- consFunc\_qifunc\_c\_year
  - constraints.h, 129
- consFunc\_qifunc\_p\_year
  - constraints.h, 129
- consFunc\_quad\_ifunc\_c
  - constraints.h, 129
- consFunc\_quad\_ifunc\_p
  - constraints.h, 130
- consFunc\_tel\_dx
  - constraints.h, 130
- consFunc\_tel\_dy
  - constraints.h, 130
- consFunc\_tel\_dz
  - constraints.h, 130
- constraint
  - tempo2.h, 205
- constraint\_efactor
  - pulsar, 76
- constraint\_label
  - tempo2.h, 203
- constraint\_param\_function
  - constraints\_param.h, 134
- constraint\_param\_info, 33
  - err, 34
  - param, 34
  - param\_k, 34
  - val, 34
- constraint\_special
  - pulsar, 77
- constraint\_str
  - enum\_str.h, 136
- constraintCounters
  - FitInfo, 36
- constraintDerivFunc
  - tempo2.h, 203
- constraintDerivs
  - FitInfo, 36
- constraintIndex
  - FitInfo, 36
- constraintSpecial
  - FitInfo, 36
- constraintValue
  - FitInfo, 36
- constraints
  - pulsar, 77
- constraints.h, 127
  - autosetDMCM, 127
  - CONSTRAINTfuncs, 130
  - computeConstraintWeights, 128
  - consFunc\_dmmodel\_cw, 128
  - consFunc\_dmmodel\_cw\_year, 128
  - consFunc\_dmmodel\_dm1, 128
  - consFunc\_dmmodel\_mean, 128
  - consFunc\_ifunc, 129
  - consFunc\_ifunc\_year, 129
  - consFunc\_qifunc\_c\_year, 129
  - consFunc\_qifunc\_p\_year, 129
  - consFunc\_quad\_ifunc\_c, 129
  - consFunc\_quad\_ifunc\_p, 130
  - consFunc\_tel\_dx, 130
  - consFunc\_tel\_dy, 130
  - consFunc\_tel\_dz, 130
  - get\_constraint\_name, 131
  - standardConstraintFunctions, 131
- constraints\_covar.h, 131
  - constraints\_covar\_ifunc, 131
- constraints\_covar\_ifunc
  - constraints\_covar.h, 131
- constraints\_nestlike.h, 132
  - constraints\_nestlike\_band, 132
  - constraints\_nestlike\_group, 132
  - constraints\_nestlike\_jitter, 133
  - constraints\_nestlike\_red, 133
  - constraints\_nestlike\_red\_dm, 133
- constraints\_nestlike\_band
  - constraints\_nestlike.h, 132
- constraints\_nestlike\_group
  - constraints\_nestlike.h, 132
- constraints\_nestlike\_jitter
  - constraints\_nestlike.h, 133
- constraints\_nestlike\_red
  - constraints\_nestlike.h, 133
- constraints\_nestlike\_red\_dm
  - constraints\_nestlike.h, 133
- constraints\_param.h, 134
  - constraint\_param\_function, 134
- copyPSR
  - tempo2.h, 211
- copyParam
  - tempo2.h, 211
- correctTroposphere
  - pulsar, 77
- correction
  - clock\_correction, 32
- correctionTT\_TB
  - observation, 55
- correctionTT\_Teph
  - observation, 55
- correctionTT\_calcEph
  - observation, 55
- correctionUT1
  - observation, 55

- correctionsTT
  - observation, [54](#)
- corrects\_to
  - clock\_correction, [32](#)
- cosl
  - TKlongdouble.float128.h, [259](#)
- covar
  - pulsar, [77](#)
- covarFuncFile
  - tempo2.h, [232](#)
- curr\_cache\_loc
  - jpl\_eph\_data, [49](#)
- DDGRmodel
  - tempo2.h, [212](#)
- DDHmodel
  - tempo2.h, [212](#)
- DDKmodel
  - tempo2.h, [212](#)
- DDSmodel
  - tempo2.h, [212](#)
- DDmodel
  - tempo2.h, [212](#)
- DEPRECATED
  - TKlog.h, [253](#)
- DLL\_FUNC
  - jpleph.h, [148](#)
- DM\_CONST\_SI
  - tempo2.h, [194](#)
- DM\_CONST
  - tempo2.h, [193](#)
- dadt
  - GWsim.h, [138](#)
- data
  - DynamicArray, [34](#)
- date\_string
  - T1Polyco, [113](#)
- dcmFile
  - tempo2.h, [232](#)
- debugFlag
  - TKlog.h, [257](#)
- decjStrPost
  - pulsar, [77](#)
- decjStrPre
  - pulsar, [77](#)
- decsim
  - pulsar, [77](#)
- dedt
  - GWsim.h, [138](#)
- defineClockCorrectionSequence
  - tempo2.h, [213](#)
- delayCorr
  - observation, [55](#)
- deleteFileName
  - pulsar, [77](#)
- deleted
  - observation, [55](#)
- destroyMemory
  - tempo2.h, [213](#)
- destroyOne
  - tempo2.h, [213](#)
- detUinv
  - pulsar, [78](#)
- dilateFreq
  - pulsar, [78](#)
- dispersion\_constant
  - ChebyModel, [30](#)
- displayCVSversion
  - tempo2.h, [232](#)
- displayMsg
  - tempo2.h, [213](#)
- displayParameters
  - tempo2.h, [213](#)
- dist\_bin
  - gwSrc, [45](#)
  - gwgeneralSrc, [41](#)
- dm
  - T1Polyco, [113](#)
- dm\_delays
  - tempo2.h, [213](#)
- dmOffset
  - pulsar, [79](#)
- dmoftsCM\_error
  - pulsar, [78](#)
- dmoftsCM\_mjd
  - pulsar, [78](#)
- dmoftsCM\_weight
  - pulsar, [78](#)
- dmoftsCMnum
  - pulsar, [78](#)
- dmoftsCM
  - pulsar, [78](#)
- dmoftsDM\_error
  - pulsar, [79](#)
- dmoftsDM\_mjd
  - pulsar, [79](#)
- dmoftsDM\_weight
  - pulsar, [79](#)
- dmoftsDMnum
  - pulsar, [79](#)
- dmoftsDM
  - pulsar, [78](#)
- dms\_turn
  - tempo2.h, [214](#)
  - tempo2Util.h, [245](#)
- doFitAll
  - tempo2.h, [214](#)
- documentation/1\_USER\_GUIDE.md, [134](#)
- documentation/2\_developers.md, [134](#)
- documentation/3\_DEVELOPER\_GUIDE.md, [134](#)
- documentation/4\_directories.md, [134](#)
- documentation/5\_plugins.md, [134](#)
- doppler
  - T1Polyco, [113](#)
- dotProduct
  - GWsim.h, [138](#)
- dotproduct

- tempo2.h, 214
- dttd
  - GWsim.h, 139
- DynamicArray, 34
  - data, 34
  - elem\_size, 35
  - nallocated, 35
  - nelem, 35
- DynamicArray\_free
  - dynarr.h, 135
- DynamicArray\_init
  - dynarr.h, 135
- DynamicArray\_push\_back
  - dynarr.h, 135
- DynamicArray\_resize
  - dynarr.h, 135
- dynarr.h, 134
  - DynamicArray\_free, 135
  - DynamicArray\_init, 135
  - DynamicArray\_push\_back, 135
  - DynamicArray\_resize, 135
- ECLIPTIC\_OBLIQUITY\_VAL
  - tempo2.h, 194
- ECLIPTIC\_OBLIQUITY
  - tempo2.h, 232
- ELL1Hmodel
  - tempo2.h, 214
- ELL1kmodel
  - tempo2.h, 214
- ELL1model
  - tempo2.h, 215
- ENDERR
  - TKlog.h, 254
- ENDL
  - TKlog.h, 254
- ERRORCOLOR
  - TKlog.h, 254
- earth\_ssb
  - observation, 55
- earthMoonBary\_earth
  - observation, 55
- earthMoonBary\_ssb
  - observation, 56
- eccRes
  - GWsim.h, 139
- eccResWithEnergy
  - GWsim.h, 139
- eclCoord
  - pulsar, 79
- efac
  - observation, 56
- einsteinRate
  - observation, 56
- elem\_size
  - DynamicArray, 35
- emrat
  - jpl\_eph\_data, 49
- enum\_str.h, 136
  - constraint\_str, 136
  - label\_str, 136
- eopc04\_file
  - pulsar, 79
- ephem\_end
  - jpl\_eph\_data, 49
- ephem\_start
  - jpl\_eph\_data, 49
- ephem\_step
  - jpl\_eph\_data, 49
- ephemeris
  - pulsar, 79
- ephemeris\_version
  - jpl\_eph\_data, 49
- equ2ecl
  - tempo2.h, 215
- equad
  - observation, 56
- err
  - constraint\_param\_info, 34
  - parameter, 66
- errorEstimates
  - FitOutput, 38
- F77\_FUNC\_
  - config.h, 123
- F77\_FUNC
  - config.h, 122
- FB90\_TIMEEPH
  - tempo2.h, 194
- FMT\_LD
  - TKlongdouble.float128.h, 259
- fabsl
  - TKlongdouble.float128.h, 259
- Fe
  - GWsim.h, 139
- fileName
  - TabulatedFunction, 116
- filterStr
  - pulsar, 80
- Findphi
  - GWsim.h, 139
- fit4
  - TKspectrum.h, 271
- fitChisq
  - pulsar, 80
- fitFlag
  - parameter, 66
- fitFunc
  - pulsar, 80
- FitInfo, 35
  - constraintCounters, 36
  - constraintDerivs, 36
  - constraintIndex, 36
  - constraintSpecial, 36
  - constraintValue, 36
  - nConstraints, 36
  - nParams, 36
  - output, 37



- paramCounters, [37](#)
- paramDerivs, [37](#)
- paramIndex, [37](#)
- tempo2.h, [203](#)
- updateFunctions, [37](#)
- fitJump
  - pulsar, [80](#)
- fitMode
  - pulsar, [80](#)
- fitNfree
  - pulsar, [80](#)
- FitOutput, [37](#)
  - errorEstimates, [38](#)
  - indexCounter, [38](#)
  - indexParam, [38](#)
  - indexPsr, [38](#)
  - parameterEstimates, [38](#)
  - tempo2.h, [203](#)
  - totalNfit, [38](#)
- fitinfo
  - pulsar, [80](#)
- fixedFormat
  - pulsar, [80](#)
- fjumpID
  - pulsar, [81](#)
- flagID
  - observation, [56](#)
- flagVal
  - observation, [56](#)
- floorl
  - TKlongdouble.float128.h, [259](#)
- fname
  - observation, [56](#)
- forceGlobalFit
  - tempo2.h, [232](#)
- formBats
  - tempo2.h, [215](#)
- formBatsAll
  - tempo2.h, [215](#)
- formResiduals
  - tempo2.h, [215](#)
- fortran\_mod
  - tempo2.h, [215](#)
- fortran\_nint
  - tempo2.h, [216](#)
- fortran\_nlong
  - tempo2.h, [216](#)
- free\_2df
  - TKmatrix.h, [266](#)
- free\_blas
  - TKmatrix.h, [266](#)
- free\_uinv
  - TKmatrix.h, [266](#)
- freq
  - observation, [56](#)
- freq\_end
  - ChebyModel, [30](#)
- freq\_start
  - ChebyModel, [30](#)
- freqSSB
  - observation, [57](#)
- frequency\_cheby
  - ChebyModel, [30](#)
- frequency\_obs
  - T1Polyco, [113](#)
- frequency\_psr\_0
  - T1Polyco, [113](#)
- GM\_C3
  - tempo2.h, [194](#)
- GMJ\_C3
  - tempo2.h, [194](#)
- GMN\_C3
  - tempo2.h, [194](#)
- GMS\_C3
  - tempo2.h, [194](#)
- GMU\_C3
  - tempo2.h, [195](#)
- GMV\_C3
  - tempo2.h, [195](#)
- GWanisotropicbackground
  - GWsim.h, [140](#)
- GWbackground
  - GWsim.h, [140](#)
- GWbackground\_read
  - GWsim.h, [140](#)
- GWbackground\_write
  - GWsim.h, [140](#)
- GWdipolebackground
  - GWsim.h, [141](#)
- GWgeneralanisotropicbackground
  - GWsim.h, [141](#)
- GWgeneralbackground
  - GWsim.h, [141](#)
- GWgeneralbackground\_read
  - GWsim.h, [141](#)
- GWgeneralbackground\_write
  - GWsim.h, [142](#)
- GWsim.h, [136](#)
  - calculateResidualGW, [138](#)
  - calculateResidualgeneralGW, [138](#)
  - dadt, [138](#)
  - dedt, [138](#)
  - dotProduct, [138](#)
  - dtdt, [139](#)
  - eccRes, [139](#)
  - eccResWithEnergy, [139](#)
  - Fe, [139](#)
  - Findphi, [139](#)
  - GWanisotropicbackground, [140](#)
  - GWbackground, [140](#)
  - GWbackground\_read, [140](#)
  - GWbackground\_write, [140](#)
  - GWdipolebackground, [141](#)
  - GWgeneralanisotropicbackground, [141](#)
  - GWgeneralbackground, [141](#)
  - GWgeneralbackground\_read, [141](#)

- GWgeneralbackground\_write, 142
- gwSrc, 137
- gwgenSpec, 137
- gwgeneralSrc, 137
- matrixMult, 142
- psrange, 142
- Rs, 142
- setupGW, 142
- setupPulsar\_GWsim, 143
- setupgeneralGW, 142
- sphharm, 143
- genrand\_int32
  - T2toolkit.h, 180
- genrand\_real1
  - T2toolkit.h, 180
- get\_EOP
  - tempo2.h, 216
- get\_OneobsCoord
  - tempo2.h, 216
- get\_blas\_cols
  - TKmatrix.h, 266
- get\_blas\_rows
  - TKmatrix.h, 267
- get\_constraint\_name
  - constraints.h, 131
- get\_obsCoord
  - tempo2.h, 216
- get\_obsCoord\_IAU2000B
  - tempo2.h, 216
- getCholeskyMatrix
  - tempo2.h, 217
- getClockCorrections
  - tempo2.h, 217
- getCorrection
  - tempo2.h, 217
- getCorrectionTT
  - tempo2.h, 217
- getInputs
  - tempo2.h, 217
- getObservatory
  - tempo2.h, 218
- getParamDeriv
  - tempo2.h, 218
- getParameterValue
  - tempo2.h, 218
- getprtj
  - TKspectrum.h, 272
- getweights
  - TKspectrum.h, 272
- globalNfit
  - pulsar, 81
- globalNoConstrain
  - pulsar, 81
- GM
  - tempo2.h, 194
- gwSrc, 44
  - across\_g, 45
  - across\_im\_g, 45
  - apls\_g, 45
  - apls\_im\_g, 45
  - dist\_bin, 45
  - GWsim.h, 137
  - h, 45
  - h\_im, 45
  - inc\_bin, 45
  - kg, 46
  - omega\_g, 46
  - phase\_g, 46
  - phi\_bin, 46
  - phi\_g, 46
  - phi\_polar\_g, 46
  - theta\_bin, 46
  - theta\_g, 46
- gwb\_decj
  - pulsar, 81
- gwb\_epoch
  - pulsar, 81
- gwb\_geom\_c
  - pulsar, 81
- gwb\_geom\_p
  - pulsar, 81
- gwb\_raj
  - pulsar, 81
- gwb\_width
  - pulsar, 82
- gwcs\_decj
  - pulsar, 82
- gwcs\_epoch
  - pulsar, 82
- gwcs\_geom\_c
  - pulsar, 82
- gwcs\_geom\_p
  - pulsar, 82
- gwcs\_raj
  - pulsar, 82
- gwcs\_width
  - pulsar, 82
- gwecc\_dec
  - pulsar, 82
- gwecc\_distance
  - pulsar, 83
- gwecc\_e
  - pulsar, 83
- gwecc\_epoch
  - pulsar, 83
- gwecc\_inc
  - pulsar, 83
- gwecc\_m1
  - pulsar, 83
- gwecc\_m2
  - pulsar, 83
- gwecc\_nodes\_orientation
  - pulsar, 83
- gwecc\_orbital\_period
  - pulsar, 83
- gwecc\_psrdist

- pulsar, [84](#)
- gwecc\_pulsarTermOn
  - pulsar, [84](#)
- gwecc\_ra
  - pulsar, [84](#)
- gwecc\_redshift
  - pulsar, [84](#)
- gwecc\_theta\_0
  - pulsar, [84](#)
- gwecc\_theta\_nodes
  - pulsar, [84](#)
- gwgenSpec, [43](#)
  - GWsim.h, [137](#)
  - sl\_alpha, [43](#)
  - sl\_amp, [43](#)
  - st\_alpha, [43](#)
  - st\_amp, [43](#)
  - tensor\_alpha, [43](#)
  - tensor\_amp, [44](#)
  - vl\_alpha, [44](#)
  - vl\_amp, [44](#)
- gwgeneralSrc, [39](#)
  - across\_g, [39](#)
  - across\_im\_g, [39](#)
  - aplust\_g, [40](#)
  - aplust\_im\_g, [40](#)
  - asl\_g, [40](#)
  - asl\_im\_g, [40](#)
  - ast\_g, [40](#)
  - ast\_im\_g, [40](#)
  - avx\_g, [40](#)
  - avx\_im\_g, [40](#)
  - avy\_g, [41](#)
  - avy\_im\_g, [41](#)
  - dist\_bin, [41](#)
  - GWsim.h, [137](#)
  - h, [41](#)
  - h\_im, [41](#)
  - inc\_bin, [41](#)
  - kg, [41](#)
  - omega\_g, [41](#)
  - phase\_g, [42](#)
  - phi\_bin, [42](#)
  - phi\_g, [42](#)
  - phi\_polar\_g, [42](#)
  - theta\_bin, [42](#)
  - theta\_g, [42](#)
- gwm\_decj
  - pulsar, [84](#)
- gwm\_dphase
  - pulsar, [84](#)
- gwm\_epoch
  - pulsar, [85](#)
- gwm\_phi
  - pulsar, [85](#)
- gwm\_raj
  - pulsar, [85](#)
- gwsrc\_across\_i
  - pulsar, [85](#)
- gwsrc\_across\_i\_e
  - pulsar, [85](#)
- gwsrc\_across\_r
  - pulsar, [85](#)
- gwsrc\_across\_r\_e
  - pulsar, [85](#)
- gwsrc\_aplust\_i
  - pulsar, [85](#)
- gwsrc\_aplust\_i\_e
  - pulsar, [86](#)
- gwsrc\_aplust\_r
  - pulsar, [86](#)
- gwsrc\_aplust\_r\_e
  - pulsar, [86](#)
- gwsrc\_dec
  - pulsar, [86](#)
- gwsrc\_epoch
  - pulsar, [86](#)
- gwsrc\_psrdist
  - pulsar, [86](#)
- gwsrc\_ra
  - pulsar, [86](#)
- h
  - gwSrc, [45](#)
  - gwgeneralSrc, [41](#)
- h\_im
  - gwSrc, [45](#)
  - gwgeneralSrc, [41](#)
- HAVE\_BLAS
  - config.h, [123](#)
- HAVE\_CFITSIO
  - config.h, [123](#)
- HAVE\_DLERROR
  - config.h, [123](#)
- HAVE\_DLFCN\_H
  - config.h, [123](#)
- HAVE\_FFTW3
  - config.h, [123](#)
- HAVE\_GWSIM\_H
  - tempo2.h, [195](#)
- HAVE\_INTTYPES\_H
  - config.h, [123](#)
- HAVE\_LAPACK
  - config.h, [124](#)
- HAVE\_LIBDLLOADER
  - config.h, [124](#)
- HAVE\_LIBDL
  - config.h, [124](#)
- HAVE\_LIBM
  - config.h, [124](#)
- HAVE\_MEMORY\_H
  - config.h, [124](#)
- HAVE\_PGPLOT
  - config.h, [124](#)
- HAVE\_PTHREAD
  - config.h, [124](#)
- HAVE\_STDINT\_H

- config.h, 124
- HAVE\_STDLIB\_H
  - config.h, 125
- HAVE\_STRING\_H
  - config.h, 125
- HAVE\_STRINGS\_H
  - config.h, 125
- HAVE\_SYS\_STAT\_H
  - config.h, 125
- HAVE\_SYS\_TYPES\_H
  - config.h, 125
- HAVE\_UNISTD\_H
  - config.h, 125
- header\_line
  - TabulatedFunction, 117
- height\_grs80
  - observatory, 64
- hms\_turn
  - tempo2.h, 218
  - tempo2Util.h, 245
- IF99\_TIMEEPH
  - tempo2.h, 195
- IFTE\_DeltaTDot
  - ifteph.h, 145
- IFTE\_DeltaT
  - ifteph.h, 145
- IFTE\_JD0
  - ifteph.h, 144
- IFTE\_KM1
  - ifteph.h, 144
- IFTE\_LC
  - ifteph.h, 144
- IFTE\_MJD0
  - ifteph.h, 144
- IFTE\_TEPH0
  - ifteph.h, 144
- IFTE\_close\_file
  - ifteph.h, 144
- IFTE\_get\_DeltaT\_DeltaTDot
  - ifteph.h, 145
- IFTE\_get\_vE\_vEDot
  - ifteph.h, 145
- IFTE\_get\_vEDot
  - ifteph.h, 145
- IFTE\_get\_vE
  - ifteph.h, 145
- IFTE\_init
  - ifteph.h, 146
- IFTE\_K
  - ifteph.h, 144
- IFTEPH\_FILE
  - tempo2.h, 195
- id\_residual
  - tempo2.h, 219
- ifile
  - jpl\_eph\_data, 49
- ifteph.h, 143
  - IFTE\_DeltaTDot, 145
  - IFTE\_DeltaT, 145
  - IFTE\_JD0, 144
  - IFTE\_KM1, 144
  - IFTE\_LC, 144
  - IFTE\_MJD0, 144
  - IFTE\_TEPH0, 144
  - IFTE\_close\_file, 144
  - IFTE\_get\_DeltaT\_DeltaTDot, 145
  - IFTE\_get\_vE\_vEDot, 145
  - IFTE\_get\_vEDot, 145
  - IFTE\_get\_vE, 145
  - IFTE\_init, 146
  - IFTE\_K, 144
- ifunc
  - ifunc.h, 146
- ifunc.h, 146
  - ifunc, 146
  - sinfunc, 146
- ifunc\_weights
  - pulsar, 86
- ifuncE
  - pulsar, 87
- ifuncN
  - pulsar, 87
- ifuncT
  - pulsar, 87
- ifuncV
  - pulsar, 87
- iinfo
  - jpl\_eph\_data, 50
- imag
  - complexVal, 33
- inc\_bin
  - gwSrc, 45
  - gwgeneralSrc, 41
- indexCounter
  - FitOutput, 38
- indexParam
  - FitOutput, 38
- indexPsr
  - FitOutput, 38
- indexx8
  - TKspectrum.h, 272
- init\_genrand
  - T2toolkit.h, 180
- initialise
  - tempo2.h, 219
- initialiseOne
  - tempo2.h, 219
- interpolation\_info, 47
  - n\_posn\_avail, 47
  - n\_vel\_avail, 47
  - posn\_coeff, 47
  - twot, 47
  - vel\_coeff, 48
- ipm
  - pulsar, 87
- ipt

- jpl\_eph\_data, 50
- JPL\_EPH\_FSEEK\_ERROR
  - jpleph.h, 149
- JPL\_EPH\_INVALID\_INDEX
  - jpleph.h, 149
- JPL\_EPH\_OUTSIDE\_RANGE
  - jpleph.h, 149
- JPL\_EPH\_QUANTITY\_NOT\_IN\_EPHEMERIS
  - jpleph.h, 149
- JPL\_EPH\_READ\_ERROR
  - jpleph.h, 149
- JPL\_EPHEM\_AU\_IN\_KM
  - jpleph.h, 149
- JPL\_EPHEM\_EARTH\_MOON\_RATIO
  - jpleph.h, 149
- JPL\_EPHEM\_END\_JD
  - jpleph.h, 149
- JPL\_EPHEM\_EPHEMERIS\_VERSION
  - jpleph.h, 150
- JPL\_EPHEM\_IPT\_ARRAY
  - jpleph.h, 150
- JPL\_EPHEM\_KERNEL\_NCOEFF
  - jpleph.h, 150
- JPL\_EPHEM\_KERNEL\_RECORD\_SIZE
  - jpleph.h, 150
- JPL\_EPHEM\_KERNEL\_SIZE
  - jpleph.h, 150
- JPL\_EPHEM\_KERNEL\_SWAP\_BYTES
  - jpleph.h, 150
- JPL\_EPHEM\_N\_CONSTANTS
  - jpleph.h, 150
- JPL\_EPHEM\_START\_JD
  - jpleph.h, 150
- JPL\_EPHEM\_STEP
  - jpleph.h, 151
- JPL\_EPHEMERIS
  - pulsar, 87
- JPL\_HEADER\_SIZE
  - jpl\_int.h, 147
- JPL\_INIT\_FILE\_CORRUPT
  - jpleph.h, 151
- JPL\_INIT\_FILE\_NOT\_FOUND
  - jpleph.h, 151
- JPL\_INIT\_FREAD2\_FAILED
  - jpleph.h, 151
- JPL\_INIT\_FREAD3\_FAILED
  - jpleph.h, 151
- JPL\_INIT\_FREAD4\_FAILED
  - jpleph.h, 151
- JPL\_INIT\_FREAD5\_FAILED
  - jpleph.h, 152
- JPL\_INIT\_FREAD\_FAILED
  - jpleph.h, 152
- JPL\_INIT\_FSEEK\_FAILED
  - jpleph.h, 152
- JPL\_INIT\_MEMORY\_FAILURE
  - jpleph.h, 152
- JPL\_INIT\_NO\_ERROR
  - jpleph.h, 152
- JPL\_INIT\_NOT\_CALLED
  - jpleph.h, 152
- JVmodel
  - tempo2.h, 219
- jboFormat
  - pulsar, 87
- jpl\_close\_ephemeris
  - jpleph.h, 152
- jpl\_eph\_data, 48
  - au, 48
  - cache, 49
  - curr\_cache\_loc, 49
  - emrat, 49
  - ephem\_end, 49
  - ephem\_start, 49
  - ephem\_step, 49
  - ephemeris\_version, 49
  - ifile, 49
  - iinfo, 50
  - ipt, 50
  - kernel\_size, 50
  - ncoeff, 50
  - ncon, 50
  - pvsun, 50
  - pvsun\_t, 50
  - resize, 50
  - swap\_bytes, 51
- jpl\_get\_constant
  - jpleph.h, 153
- jpl\_get\_double
  - jpleph.h, 153
- jpl\_get\_long
  - jpleph.h, 153
- jpl\_get\_pvsun
  - jpleph.h, 151
- jpl\_init\_ephemeris
  - jpleph.h, 153
- jpl\_init\_error\_code
  - jpleph.h, 153
- jpl\_int.h, 147
  - JPL\_HEADER\_SIZE, 147
  - MAX\_CHEBY, 147
- jpl\_pleph
  - jpleph.h, 153
- jpl\_state
  - jpleph.h, 154
- jpleph.h, 148
  - DLL\_FUNC, 148
  - JPL\_EPH\_FSEEK\_ERROR, 149
  - JPL\_EPH\_INVALID\_INDEX, 149
  - JPL\_EPH\_OUTSIDE\_RANGE, 149
  - JPL\_EPH\_QUANTITY\_NOT\_IN\_EPHEMERIS, 149
  - JPL\_EPH\_READ\_ERROR, 149
  - JPL\_EPHEM\_AU\_IN\_KM, 149
  - JPL\_EPHEM\_EARTH\_MOON\_RATIO, 149
  - JPL\_EPHEM\_END\_JD, 149

- JPL\_EPHEM\_EPHEMERIS\_VERSION, [150](#)
- JPL\_EPHEM\_IPT\_ARRAY, [150](#)
- JPL\_EPHEM\_KERNEL\_NCOEFF, [150](#)
- JPL\_EPHEM\_KERNEL\_RECORD\_SIZE, [150](#)
- JPL\_EPHEM\_KERNEL\_SIZE, [150](#)
- JPL\_EPHEM\_KERNEL\_SWAP\_BYTES, [150](#)
- JPL\_EPHEM\_N\_CONSTANTS, [150](#)
- JPL\_EPHEM\_START\_JD, [150](#)
- JPL\_EPHEM\_STEP, [151](#)
- JPL\_INIT\_FILE\_CORRUPT, [151](#)
- JPL\_INIT\_FILE\_NOT\_FOUND, [151](#)
- JPL\_INIT\_FREAD2\_FAILED, [151](#)
- JPL\_INIT\_FREAD3\_FAILED, [151](#)
- JPL\_INIT\_FREAD4\_FAILED, [151](#)
- JPL\_INIT\_FREAD5\_FAILED, [152](#)
- JPL\_INIT\_FREAD\_FAILED, [152](#)
- JPL\_INIT\_FSEEK\_FAILED, [152](#)
- JPL\_INIT\_MEMORY\_FAILURE, [152](#)
- JPL\_INIT\_NO\_ERROR, [152](#)
- JPL\_INIT\_NOT\_CALLED, [152](#)
- jpl\_close\_ephemeris, [152](#)
- jpl\_get\_constant, [153](#)
- jpl\_get\_double, [153](#)
- jpl\_get\_long, [153](#)
- jpl\_get\_pvsun, [151](#)
- jpl\_init\_ephemeris, [153](#)
- jpl\_init\_error\_code, [153](#)
- jpl\_pleph, [153](#)
- jpl\_state, [154](#)
- make\_sub\_ephem, [154](#)
- jump
  - observation, [57](#)
- jumpSAT
  - pulsar, [87](#)
- jumpStr
  - pulsar, [88](#)
- jumpVal
  - pulsar, [88](#)
- jumpValErr
  - pulsar, [88](#)
- jupiter\_earth
  - observation, [57](#)
- kernel\_size
  - jpl\_eph\_data, [50](#)
- kg
  - gwSrc, [46](#)
  - gwgneralSrc, [41](#)
- kind
  - T2Predictor, [116](#)
- LD\_PI
  - TKlongdouble.float128.h, [260](#)
  - TKlongdouble.h, [262](#)
  - TKlongdouble.ld.h, [264](#)
- LEAPSECOND\_FILE
  - tempo2.h, [195](#)
- LOG\_OUTFILE
  - TKlog.h, [254](#)
- LONGDOUBLE\_IS\_FLOAT128
  - TKlongdouble.float128.h, [260](#)
- LONGDOUBLE\_IS\_IEEE754
  - TKlongdouble.h, [263](#)
  - TKlongdouble.ld.h, [265](#)
- LONGDOUBLE\_ONE
  - TKlongdouble.float128.h, [260](#)
  - TKlongdouble.h, [263](#)
  - TKlongdouble.ld.h, [265](#)
- LT\_OBJDIR
  - config.h, [125](#)
- label
  - parameter, [67](#)
  - tempo2.h, [206](#)
- label\_str
  - enum\_str.h, [136](#)
- latitude\_grs80
  - observatory, [65](#)
- ld\_fprintf
  - TKlongdouble.float128.h, [261](#)
  - TKlongdouble.h, [262](#)
  - TKlongdouble.ld.h, [264](#)
- ld\_printf
  - TKlongdouble.float128.h, [261](#)
  - TKlongdouble.h, [262](#)
  - TKlongdouble.ld.h, [264](#)
- ld\_sprintf
  - TKlongdouble.float128.h, [261](#)
  - TKlongdouble.h, [262](#)
  - TKlongdouble.ld.h, [264](#)
- lib2toolkit API, [27](#)
- libtempo2 External API, [28](#)
- linkFrom
  - parameter, [67](#)
- linkTo
  - parameter, [67](#)
- log10rms
  - T1Polyco, [113](#)
- logall
  - TKlog.h, [254](#)
- logdbg
  - TKlog.h, [254](#)
- logerr
  - TKlog.h, [254](#)
- logerr\_check
  - TKlog.h, [257](#)
- logicFlag
  - tempo2.h, [219](#)
- logmsg
  - TKlog.h, [255](#)
- logtchk
  - TKlog.h, [255](#)
- logwarn
  - TKlog.h, [255](#)
- longdouble
  - TKlongdouble.float128.h, [260](#), [261](#)
  - TKlongdouble.h, [263](#)
  - TKlongdouble.ld.h, [265](#)

- longitude\_grs80
  - observatory, [65](#)
- lookup\_observatory\_alias
  - tempo2.h, [219](#)
- MASYR2RADS
  - tempo2.h, [195](#)
- MAX\_BPJ\_JUMPS
  - tempo2.h, [195](#)
- MAX\_CHEBY
  - jpl\_int.h, [147](#)
- MAX\_CLK\_CORR
  - tempo2.h, [196](#)
- MAX\_CLKCORR
  - tempo2.h, [196](#)
- MAX\_COEFF
  - tempo2.h, [196](#)
- MAX\_COMPANIONS
  - tempo2.h, [196](#)
- MAX\_DM\_DERIVATIVES
  - tempo2.h, [196](#)
- MAX\_DMX
  - tempo2.h, [196](#)
- MAX\_FILELEN
  - tempo2.h, [196](#)
- MAX\_FIT
  - tempo2.h, [196](#)
- MAX\_FLAG\_LEN
  - tempo2.h, [197](#)
- MAX\_FLAGS
  - tempo2.h, [197](#)
- MAX\_FREQ\_DERIVATIVES
  - tempo2.h, [197](#)
- MAX\_IFUNC
  - tempo2.h, [197](#)
- MAX\_JUMPS
  - tempo2.h, [197](#)
- MAX\_LEAPSEC
  - tempo2.h, [197](#)
- MAX\_MSG
  - tempo2.h, [197](#)
- MAX\_OBSN\_VAL
  - tempo2.h, [197](#)
- MAX\_OBSN
  - tempo2.h, [232](#)
- MAX\_PARAMS
  - tempo2.h, [198](#)
- MAX\_PSR\_VAL
  - tempo2.h, [198](#)
- MAX\_PSR
  - tempo2.h, [232](#)
- MAX\_QUAD
  - tempo2.h, [198](#)
- MAX\_SITE
  - tempo2.h, [198](#)
- MAX\_STOREPRECISION
  - tempo2.h, [198](#)
- MAX\_STRLEN
  - tempo2.h, [198](#)
- MAX\_SX
  - tempo2.h, [198](#)
- MAX\_T2EFAC
  - tempo2.h, [198](#)
- MAX\_T2EQUAD
  - tempo2.h, [199](#)
- MAX\_TEL\_CLK\_OFFS
  - tempo2.h, [199](#)
- MAX\_TEL\_DX
  - tempo2.h, [199](#)
- MAX\_TEL\_DY
  - tempo2.h, [199](#)
- MAX\_TEL\_DZ
  - tempo2.h, [199](#)
- MAX\_TNBN
  - tempo2.h, [199](#)
- MAX\_TNDMEv
  - tempo2.h, [199](#)
- MAX\_TNECORR
  - tempo2.h, [199](#)
- MAX\_TNEF
  - tempo2.h, [200](#)
- MAX\_TNEQ
  - tempo2.h, [200](#)
- MAX\_TNGN
  - tempo2.h, [200](#)
- MAX\_TNSQ
  - tempo2.h, [200](#)
- MAX\_TOFFSET
  - tempo2.h, [200](#)
- MAX\_WHITE
  - tempo2.h, [200](#)
- MAX
  - TKspectrum.h, [270](#)
- MIN
  - TKspectrum.h, [270](#)
- MSSmodel
  - tempo2.h, [220](#)
- make\_sub\_ephem
  - jpleph.h, [154](#)
- malloc\_2df
  - TKmatrix.h, [267](#)
- malloc\_blas
  - TKmatrix.h, [267](#)
- malloc\_uinv
  - TKmatrix.h, [267](#)
- mat20
  - TKspectrum.h, [272](#)
- matrixMult
  - GWsim.h, [142](#)
- minPrec
  - storePrecision, [111](#)
- mjd\_end
  - ChebyModel, [31](#)
- mjd\_mid
  - T1Polyco, [113](#)
- mjd\_start
  - ChebyModel, [31](#)

- modelset
  - T2Predictor, [116](#)
- n\_posn\_avail
  - interpolation\_info, [47](#)
- n\_vel\_avail
  - interpolation\_info, [47](#)
- nCompanion
  - pulsar, [88](#)
- nConstraints
  - FitInfo, [36](#)
- nDMEvents
  - pulsar, [88](#)
- NE\_SW\_DEFAULT
  - tempo2.h, [200](#)
- NEWFIT
  - tempo2.h, [232](#)
- nFit
  - pulsar, [89](#)
- nFlags
  - observation, [57](#)
- nGlobal
  - pulsar, [89](#)
- nJumps
  - pulsar, [89](#)
- nLinkFrom
  - parameter, [67](#)
- nLinkTo
  - parameter, [67](#)
- nParam
  - pulsar, [89](#)
- nParams
  - FitInfo, [36](#)
- nPhaseJump
  - pulsar, [90](#)
- nQuad
  - pulsar, [90](#)
- nStorePrecision
  - pulsar, [90](#)
- nSx
  - pulsar, [90](#)
- nT2efac
  - pulsar, [90](#)
- nT2equad
  - pulsar, [90](#)
- nTNBandNoise
  - pulsar, [91](#)
- nTNECORR
  - pulsar, [91](#)
- nTNEF
  - pulsar, [91](#)
- nTNEQ
  - pulsar, [91](#)
- nTNGroupNoise
  - pulsar, [91](#)
- nTNShapeletEvents
  - pulsar, [91](#)
- nTNSQ
  - pulsar, [91](#)
- nTelDX
  - pulsar, [90](#)
- nTelDY
  - pulsar, [90](#)
- nTelDZ
  - pulsar, [91](#)
- nToffset
  - pulsar, [92](#)
- nWhite
  - pulsar, [92](#)
- nWhite\_dm
  - pulsar, [92](#)
- nallocated
  - DynamicArray, [35](#)
- name
  - observatory, [65](#)
  - pulsar, [88](#)
- nclock\_correction
  - observation, [57](#)
- ncoeff
  - jpl\_eph\_data, [50](#)
  - T1Polyco, [113](#)
- ncon
  - jpl\_eph\_data, [50](#)
- nconstraints
  - pulsar, [88](#)
- ndmx
  - pulsar, [88](#)
- ne\_sw
  - pulsar, [89](#)
- nelem
  - DynamicArray, [35](#)
- neptune\_earth
  - observation, [57](#)
- nits
  - pulsar, [89](#)
- noWarnings
  - pulsar, [89](#)
- nobs
  - pulsar, [89](#)
- nphase
  - observation, [57](#)
- nsegments
  - ChebyModelSet, [32](#)
  - T1PolycoSet, [115](#)
- nutations
  - observation, [57](#)
- nx
  - Cheby2D, [29](#)
- ny
  - Cheby2D, [29](#)
- OBLQ
  - tempo2.h, [200](#)
- OBSSYS\_FILE
  - tempo2.h, [201](#)
- obsNjump
  - observation, [58](#)
- observation, [51](#)



- addedNoise, [53](#)
- averagebat, [53](#)
- averagedmbat, [53](#)
- averagedmerr, [53](#)
- averagedmres, [53](#)
- averageerr, [53](#)
- averageres, [54](#)
- bat, [54](#)
- batCorr, [54](#)
- bbat, [54](#)
- bline, [54](#)
- chisq, [54](#)
- clockCorr, [54](#)
- correctionTT\_TB, [55](#)
- correctionTT\_Teph, [55](#)
- correctionTT\_calcEph, [55](#)
- correctionUT1, [55](#)
- correctionsTT, [54](#)
- delayCorr, [55](#)
- deleted, [55](#)
- earth\_ssb, [55](#)
- earthMoonBary\_earth, [55](#)
- earthMoonBary\_ssb, [56](#)
- efac, [56](#)
- einsteinRate, [56](#)
- equad, [56](#)
- flagID, [56](#)
- flagVal, [56](#)
- fname, [56](#)
- freq, [56](#)
- freqSSB, [57](#)
- jump, [57](#)
- jupiter\_earth, [57](#)
- nFlags, [57](#)
- nclock\_correction, [57](#)
- neptune\_earth, [57](#)
- nphase, [57](#)
- nutations, [57](#)
- obsNjump, [58](#)
- observatory\_earth, [58](#)
- origErr, [58](#)
- origsat, [58](#)
- pet, [58](#)
- phase, [58](#)
- phaseOffset, [58](#)
- planet\_ssb, [58](#)
- planet\_ssb\_derv, [59](#)
- planet\_ssb\_tmr, [59](#)
- pnoise, [59](#)
- prefitResidual, [59](#)
- psrPos, [59](#)
- pulseN, [59](#)
- residual, [59](#)
- residualtn, [59](#)
- roemer, [60](#)
- sat, [60](#)
- sat\_day, [60](#)
- sat\_sec, [60](#)
- saturn\_earth, [60](#)
- shapiroDelayJupiter, [60](#)
- shapiroDelayNeptune, [60](#)
- shapiroDelaySaturn, [60](#)
- shapiroDelaySun, [61](#)
- shapiroDelayUranus, [61](#)
- shapiroDelayVenus, [61](#)
- shklovskii, [61](#)
- siteVel, [61](#)
- snr, [61](#)
- sun\_earth, [61](#)
- sun\_ssb, [61](#)
- TNDMErr, [62](#)
- TNDMSignal, [62](#)
- TNGroupErr, [62](#)
- TNGroupSignal, [62](#)
- TNRedErr, [62](#)
- TNRedSignal, [63](#)
- tdis1, [62](#)
- tdis2, [62](#)
- tellID, [62](#)
- tempo2.h, [204](#)
- toaDMErr, [63](#)
- toaErr, [63](#)
- tobs, [63](#)
- torb, [63](#)
- troposphericDelay, [63](#)
- uranus\_earth, [63](#)
- venus\_earth, [63](#)
- zenith, [64](#)
- observatory, [64](#)
  - clock\_name, [64](#)
  - code, [64](#)
  - height\_grs80, [64](#)
  - latitude\_grs80, [65](#)
  - longitude\_grs80, [65](#)
  - name, [65](#)
  - x, [65](#)
  - y, [65](#)
  - z, [65](#)
- observatory\_earth
  - observation, [58](#)
- obsn
  - pulsar, [92](#)
- offset
  - pulsar, [92](#)
- offset\_e
  - pulsar, [92](#)
- omega\_g
  - gwSrc, [46](#)
  - gwgeneralSrc, [41](#)
- open\_file
  - read\_fortran.h, [155](#)
- open\_file2
  - read\_fortran2.h, [157](#)
- origErr
  - observation, [58](#)
- origsat

- observation, [58](#)
- output
  - FitInfo, [37](#)
- outputTMatrix
  - pulsar, [92](#)
- PACKAGE\_BUGREPORT
  - config.h, [126](#)
- PACKAGE\_NAME
  - config.h, [126](#)
- PACKAGE\_STRING
  - config.h, [126](#)
- PACKAGE\_TARNAME
  - config.h, [126](#)
- PACKAGE\_URL
  - config.h, [126](#)
- PACKAGE\_VERSION
  - config.h, [126](#)
- PACKAGE
  - config.h, [125](#)
- PCM
  - tempo2.h, [201](#)
- param
  - constraint\_param\_info, [34](#)
  - pulsar, [92](#)
- param\_k
  - constraint\_param\_info, [34](#)
- param\_label
  - tempo2.h, [204](#)
- paramCounters
  - FitInfo, [37](#)
- paramDerivFunc
  - tempo2.h, [204](#)
- paramDerivs
  - FitInfo, [37](#)
- paramIndex
  - FitInfo, [37](#)
- paramSet
  - parameter, [67](#)
- paramUpdateFunc
  - tempo2.h, [204](#)
- parameter, [66](#)
  - aSize, [66](#)
  - err, [66](#)
  - fitFlag, [66](#)
  - label, [67](#)
  - linkFrom, [67](#)
  - linkTo, [67](#)
  - nLinkFrom, [67](#)
  - nLinkTo, [67](#)
  - paramSet, [67](#)
  - prefit, [67](#)
  - prefitErr, [67](#)
  - shortlabel, [68](#)
  - tempo2.h, [204](#)
  - val, [68](#)
- parameterEstimates
  - FitOutput, [38](#)
- parse\_longdouble
  - TKlongdouble.float128.h, [261](#)
  - TKlongdouble.h, [263](#)
  - TKlongdouble.ld.h, [265](#)
- passStr
  - pulsar, [93](#)
- pet
  - observation, [58](#)
- phase
  - observation, [58](#)
- phase\_g
  - gwSrc, [46](#)
  - gwgeneralSrc, [42](#)
- phaseJump
  - pulsar, [93](#)
- phaseJumpDir
  - pulsar, [93](#)
- phaseJumpID
  - pulsar, [93](#)
- phaseOffset
  - observation, [58](#)
- phi\_bin
  - gwSrc, [46](#)
  - gwgeneralSrc, [42](#)
- phi\_g
  - gwSrc, [46](#)
  - gwgeneralSrc, [42](#)
- phi\_polar\_g
  - gwSrc, [46](#)
  - gwgeneralSrc, [42](#)
- planet\_ssb
  - observation, [58](#)
- planet\_ssb\_derv
  - observation, [59](#)
- planet\_ssb\_tmr
  - observation, [59](#)
- planetShapiro
  - pulsar, [93](#)
- pnoise
  - observation, [59](#)
- polyco
  - tempo2.h, [220](#)
- posPulsar
  - pulsar, [93](#)
- posn\_coeff
  - interpolation\_info, [47](#)
- powl
  - TKlongdouble.float128.h, [260](#)
- preProcess
  - tempo2.h, [220](#)
- preProcessSimple
  - tempo2.h, [220](#)
- preProcessSimple1
  - tempo2.h, [220](#)
- preProcessSimple2
  - tempo2.h, [221](#)
- preProcessSimple3
  - tempo2.h, [221](#)
- prefit

- parameter, 67
- prefitErr
  - parameter, 67
- prefitResidual
  - observation, 59
- processFlag
  - tempo2.h, 221
- processSimultaneous
  - tempo2.h, 221
- psrPos
  - observation, 59
- psrangle
  - GWsim.h, 142
- psrname
  - ChebyModel, 31
  - T1Polyco, 114
- pulsar, 68
  - addTNGlobalEQ, 74
  - auto\_constraints, 74
  - AverageDMResiduals, 74
  - AverageEpochWidth, 74
  - AverageFlag, 74
  - AverageResiduals, 74
  - binaryModel, 74
  - bootStrap, 75
  - brace, 75
  - calcShapiro, 75
  - cgw\_angpol, 75
  - cgw\_cosinc, 75
  - cgw\_h0, 75
  - cgw\_mc, 75
  - clk\_offsE, 76
  - clk\_offsT, 76
  - clk\_offsV, 76
  - clkOffsN, 76
  - clock, 76
  - clockFromOverride, 76
  - constraint\_efactor, 76
  - constraint\_special, 77
  - constraints, 77
  - correctTroposphere, 77
  - covar, 77
  - decjStrPost, 77
  - decjStrPre, 77
  - decsim, 77
  - deleteFileName, 77
  - detUinv, 78
  - dilateFreq, 78
  - dmOffset, 79
  - dmoffsCM\_error, 78
  - dmoffsCM\_mjd, 78
  - dmoffsCM\_weight, 78
  - dmoffsCMnum, 78
  - dmoffsCM, 78
  - dmoffsDM\_error, 79
  - dmoffsDM\_mjd, 79
  - dmoffsDM\_weight, 79
  - dmoffsDMnum, 79
  - dmoffsDM, 78
  - eclCoord, 79
  - eopc04\_file, 79
  - ephemeris, 79
  - filterStr, 80
  - fitChisq, 80
  - fitFunc, 80
  - fitJump, 80
  - fitMode, 80
  - fitNfree, 80
  - fitinfo, 80
  - fixedFormat, 80
  - fjumpID, 81
  - globalNfit, 81
  - globalNoConstrain, 81
  - gwb\_decj, 81
  - gwb\_epoch, 81
  - gwb\_geom\_c, 81
  - gwb\_geom\_p, 81
  - gwb\_raj, 81
  - gwb\_width, 82
  - gwcs\_decj, 82
  - gwcs\_epoch, 82
  - gwcs\_geom\_c, 82
  - gwcs\_geom\_p, 82
  - gwcs\_raj, 82
  - gwcs\_width, 82
  - gwecc\_dec, 82
  - gwecc\_distance, 83
  - gwecc\_e, 83
  - gwecc\_epoch, 83
  - gwecc\_inc, 83
  - gwecc\_m1, 83
  - gwecc\_m2, 83
  - gwecc\_nodes\_orientation, 83
  - gwecc\_orbital\_period, 83
  - gwecc\_psrdist, 84
  - gwecc\_pulsarTermOn, 84
  - gwecc\_ra, 84
  - gwecc\_redshift, 84
  - gwecc\_theta\_0, 84
  - gwecc\_theta\_nodes, 84
  - gwm\_decj, 84
  - gwm\_dphase, 84
  - gwm\_epoch, 85
  - gwm\_phi, 85
  - gwm\_raj, 85
  - gwsrc\_across\_i, 85
  - gwsrc\_across\_i\_e, 85
  - gwsrc\_across\_r, 85
  - gwsrc\_across\_r\_e, 85
  - gwsrc\_aplus\_i, 85
  - gwsrc\_aplus\_i\_e, 86
  - gwsrc\_aplus\_r, 86
  - gwsrc\_aplus\_r\_e, 86
  - gwsrc\_dec, 86
  - gwsrc\_epoch, 86
  - gwsrc\_psrdist, 86

gwsrsrc\_ra, 86  
 ifunc\_weights, 86  
 ifuncE, 87  
 ifuncN, 87  
 ifuncT, 87  
 ifuncV, 87  
 ipm, 87  
 JPL\_EPHEMERIS, 87  
 jboFormat, 87  
 jumpSAT, 87  
 jumpStr, 88  
 jumpVal, 88  
 jumpValErr, 88  
 nCompanion, 88  
 nDMEvents, 88  
 nFit, 89  
 nGlobal, 89  
 nJumps, 89  
 nParam, 89  
 nPhaseJump, 90  
 nQuad, 90  
 nStorePrecision, 90  
 nSx, 90  
 nT2efac, 90  
 nT2equad, 90  
 nTNBandNoise, 91  
 nTNECORR, 91  
 nTNEF, 91  
 nTNEQ, 91  
 nTNGroupNoise, 91  
 nTNShapeletEvents, 91  
 nTNSQ, 91  
 nTelDX, 90  
 nTelDY, 90  
 nTelDZ, 91  
 nToffset, 92  
 nWhite, 92  
 nWhite\_dm, 92  
 name, 88  
 nconstraints, 88  
 ndmx, 88  
 ne\_sw, 89  
 nits, 89  
 noWarnings, 89  
 nobs, 89  
 obsn, 92  
 offset, 92  
 offset\_e, 92  
 outputTMatrix, 92  
 param, 92  
 passStr, 93  
 phaseJump, 93  
 phaseJumpDir, 93  
 phaseJumpID, 93  
 planetShapiro, 93  
 posPulsar, 93  
 quad\_across\_i, 93  
 quad\_across\_i\_e, 93  
 quad\_across\_r, 94  
 quad\_across\_r\_e, 94  
 quad\_aplus\_i, 94  
 quad\_aplus\_i\_e, 94  
 quad\_aplus\_r, 94  
 quad\_aplus\_r\_e, 94  
 quad\_ifunc\_c\_DEC, 94  
 quad\_ifunc\_c\_RA, 94  
 quad\_ifunc\_geom\_c, 95  
 quad\_ifunc\_geom\_p, 95  
 quad\_ifunc\_p\_DEC, 95  
 quad\_ifunc\_p\_RA, 95  
 quad\_ifuncE\_c, 95  
 quad\_ifuncE\_p, 95  
 quad\_ifuncN\_c, 95  
 quad\_ifuncN\_p, 95  
 quad\_ifuncT\_c, 96  
 quad\_ifuncT\_p, 96  
 quad\_ifuncV\_c, 96  
 quad\_ifuncV\_p, 96  
 quadDEC, 96  
 quadEpoch, 96  
 quadRA, 96  
 rajStrPost, 96  
 rajStrPre, 97  
 rasim, 97  
 rescaleErrChisq, 97  
 rmsPost, 97  
 rmsPre, 97  
 rmstn, 97  
 robust, 97  
 setTelVelX, 97  
 setTelVelY, 98  
 setTelVelZ, 98  
 setUnits, 98  
 simflag, 98  
 sorted, 98  
 storePrec, 98  
 swm, 98  
 t2cMethod, 99  
 T2efacFlagID, 99  
 T2efacFlagVal, 99  
 T2efacVal, 99  
 T2equadFlagID, 99  
 T2equadFlagVal, 99  
 T2equadVal, 99  
 T2globalEfac, 99  
 TNBandDMAmp, 102  
 TNBandDMGam, 102  
 TNBandDMC, 102  
 TNBandNoiseAmp, 102  
 TNBandNoiseGam, 102  
 TNBandNoiseHF, 102  
 TNBandNoiseLF, 103  
 TNBandNoiseC, 102  
 TNDMAmp, 103  
 TNDMCoeffs, 103  
 TNDMEvAmp, 103

TNDMEvGam, 103  
 TNDMEvLength, 103  
 TNDMEvLin, 103  
 TNDMEvOff, 104  
 TNDMEvQuad, 104  
 TNDMEvStart, 104  
 TNDMGam, 104  
 TNDMC, 103  
 TNECORRFlagID, 104  
 TNECORRFlagVal, 104  
 TNECORRVal, 104  
 TNEFFlagID, 104  
 TNEFFlagVal, 105  
 TNEFVal, 105  
 TNEQFlagID, 105  
 TNEQFlagVal, 105  
 TNEQVal, 105  
 TNGlobalEF, 105  
 TNGlobalEQ, 105  
 TNGroupNoiseAmp, 105  
 TNGroupNoiseFlagID, 106  
 TNGroupNoiseFlagVal, 106  
 TNGroupNoiseGam, 106  
 TNGroupNoiseC, 106  
 TNRedAmp, 106  
 TNRedCoeffs, 106  
 TNRedCorner, 106  
 TNRedFLow, 107  
 TNRedGam, 107  
 TNRedC, 106  
 TNSQFlagID, 107  
 TNSQFlagVal, 107  
 TNSQVal, 108  
 TNShapeletEvFScale, 107  
 TNShapeletEvPos, 107  
 TNShapeletEvWidth, 107  
 TNShapeletEvN, 107  
 TNsubtractDM, 108  
 TNsubtractRed, 108  
 tOffset, 108  
 tOffset\_f1, 108  
 tOffset\_f2, 108  
 tOffset\_t1, 108  
 tOffset\_t2, 109  
 tOffsetFlags, 109  
 tOffsetSite, 109  
 telDX\_e, 100  
 telDX\_t, 100  
 telDX\_v, 100  
 telDX\_vel, 100  
 telDX\_vel\_e, 100  
 telDY\_e, 100  
 telDY\_t, 100  
 telDY\_v, 100  
 telDY\_vel, 101  
 telDY\_vel\_e, 101  
 telDZ\_e, 101  
 telDZ\_t, 101  
 telDZ\_v, 101  
 telDZ\_vel, 101  
 telDZ\_vel\_e, 101  
 tempo1, 101  
 tempo2.h, 204  
 timeEphemeris, 102  
 ToAextraCovar, 108  
 tzrsite, 109  
 units, 109  
 useCalceph, 109  
 useTNOOrth, 109  
 velPulsar, 109  
 wave\_cos, 110  
 wave\_cos\_dm, 110  
 wave\_cos\_dm\_err, 110  
 wave\_cos\_err, 110  
 wave\_sine, 110  
 wave\_sine\_dm, 110  
 wave\_sine\_dm\_err, 110  
 wave\_sine\_err, 110  
 waveScale, 111  
 whiteNoiseModelFile, 111  
 pulseN  
     observation, 59  
 pvsun  
     jpl\_eph\_data, 50  
 pvsun\_t  
     jpl\_eph\_data, 50  
 QR\_DEFAULT  
     config.h, 126  
 quad\_across\_i  
     pulsar, 93  
 quad\_across\_i\_e  
     pulsar, 93  
 quad\_across\_r  
     pulsar, 94  
 quad\_across\_r\_e  
     pulsar, 94  
 quad\_aplus\_i  
     pulsar, 94  
 quad\_aplus\_i\_e  
     pulsar, 94  
 quad\_aplus\_r  
     pulsar, 94  
 quad\_aplus\_r\_e  
     pulsar, 94  
 quad\_ifunc\_c\_DEC  
     pulsar, 94  
 quad\_ifunc\_c\_RA  
     pulsar, 94  
 quad\_ifunc\_geom\_c  
     pulsar, 95  
 quad\_ifunc\_geom\_p  
     pulsar, 95  
 quad\_ifunc\_p\_DEC  
     pulsar, 95  
 quad\_ifunc\_p\_RA  
     pulsar, 95

- quad\_ifuncE\_c
  - pulsar, 95
- quad\_ifuncE\_p
  - pulsar, 95
- quad\_ifuncN\_c
  - pulsar, 95
- quad\_ifuncN\_p
  - pulsar, 95
- quad\_ifuncT\_c
  - pulsar, 96
- quad\_ifuncT\_p
  - pulsar, 96
- quad\_ifuncV\_c
  - pulsar, 96
- quad\_ifuncV\_p
  - pulsar, 96
- quadDEC
  - pulsar, 96
- quadEpoch
  - pulsar, 96
- quadRA
  - pulsar, 96
- quietFlag
  - TKlog.h, 257
- README.md, 158
- RESETCOLOR
  - TKlog.h, 255
- rajStrPost
  - pulsar, 96
- rajStrPre
  - pulsar, 97
- rasim
  - pulsar, 97
- read\_char
  - read\_fortran.h, 155
- read\_character
  - read\_fortran.h, 155
- read\_character2
  - read\_fortran2.h, 157
- read\_double
  - read\_fortran.h, 155
- read\_double2
  - read\_fortran2.h, 157
- read\_float
  - read\_fortran.h, 155
- read\_float2
  - read\_fortran2.h, 157
- read\_fortran.h, 154
  - c\_fileptr, 156
  - close\_file, 155
  - open\_file, 155
  - read\_char, 155
  - read\_character, 155
  - read\_double, 155
  - read\_float, 155
  - read\_int, 155
  - read\_record\_int, 156
  - swapByte, 156
- read\_fortran2.h, 156
  - c\_fileptr2, 158
  - close\_file2, 157
  - open\_file2, 157
  - read\_character2, 157
  - read\_double2, 157
  - read\_float2, 157
  - read\_int2, 157
  - read\_record\_int2, 157
  - swapByte2, 158
- read\_int
  - read\_fortran.h, 155
- read\_int2
  - read\_fortran2.h, 157
- read\_record\_int
  - read\_fortran.h, 156
- read\_record\_int2
  - read\_fortran2.h, 157
- readEphemeris
  - tempo2.h, 221
- readEphemeris\_calceph
  - tempo2.h, 221
- readJBO\_bat
  - tempo2.h, 222
- readObsFile
  - tempo2.h, 222
- readOneEphemeris
  - tempo2.h, 222
- readParfile
  - tempo2.h, 222
- readParfileGlobal
  - tempo2.h, 222
- readSimpleParfile
  - tempo2.h, 223
- readTimfile
  - tempo2.h, 223
- real
  - complexVal, 33
- recordPrecision
  - tempo2.h, 223
- resize
  - jpl\_eph\_data, 50
- reference\_phase
  - T1Polyco, 114
- rescaleErrChisq
  - pulsar, 97
- residual
  - observation, 59
- residualtn
  - observation, 59
- rmsPost
  - pulsar, 97
- rmsPre
  - pulsar, 97
- rmstn
  - pulsar, 97
- robust
  - pulsar, 97

- roemer
  - observation, [60](#)
- routine
  - storePrecision, [111](#)
- Rs
  - GWsim.h, [142](#)
- SECDAYI
  - tempo2.h, [201](#)
- SECDAY
  - tempo2.h, [201](#)
- SI\_UNITS
  - tempo2.h, [201](#)
- SOLAR\_MASS
  - tempo2.h, [201](#)
- SOLAR\_RADIUS
  - tempo2.h, [201](#)
- SPEED\_LIGHT
  - tempo2.h, [201](#)
- STDC\_HEADERS
  - config.h, [126](#)
- samples
  - TabulatedFunction, [117](#)
- sat
  - observation, [60](#)
- sat\_day
  - observation, [60](#)
- sat\_sec
  - observation, [60](#)
- saturn\_earth
  - observation, [60](#)
- secularMotion
  - tempo2.h, [223](#)
- segments
  - ChebyModelSet, [32](#)
  - T1PolycoSet, [115](#)
- setPlugPath
  - tempo2.h, [223](#)
- setStart
  - tempo2.h, [223](#)
- setTelVelX
  - pulsar, [97](#)
- setTelVelY
  - pulsar, [98](#)
- setTelVelZ
  - pulsar, [98](#)
- setUnits
  - pulsar, [98](#)
- setupGW
  - GWsim.h, [142](#)
- setupParameterFileDefaults
  - tempo2.h, [224](#)
- setupPulsar\_GWsim
  - GWsim.h, [143](#)
- setupgeneralGW
  - GWsim.h, [142](#)
- shapiro\_delay
  - tempo2.h, [224](#)
- shapiroDelayJupiter
  - observation, [60](#)
- shapiroDelayNeptune
  - observation, [60](#)
- shapiroDelaySaturn
  - observation, [60](#)
- shapiroDelaySun
  - observation, [61](#)
- shapiroDelayUranus
  - observation, [61](#)
- shapiroDelayVenus
  - observation, [61](#)
- shklovskii
  - observation, [61](#)
- shortlabel
  - parameter, [68](#)
- simflag
  - pulsar, [98](#)
- simplePlot
  - tempo2.h, [224](#)
- sineFunc
  - TKspectrum.h, [272](#)
- sinefunc
  - ifunc.h, [146](#)
- sinl
  - TKlongdouble.float128.h, [260](#)
- siteVel
  - observation, [61](#)
- sitename
  - ChebyModel, [31](#)
  - T1Polyco, [114](#)
- sl\_alpha
  - gwgenSpec, [43](#)
- sl\_amp
  - gwgenSpec, [43](#)
- snr
  - observation, [61](#)
- solarWindModel
  - tempo2.h, [224](#)
- sortToAs
  - tempo2.h, [224](#)
- sorted
  - pulsar, [98](#)
- span
  - T1Polyco, [114](#)
- sphharm
  - GWsim.h, [143](#)
- st\_alpha
  - gwgenSpec, [43](#)
- st\_amp
  - gwgenSpec, [43](#)
- standardConstraintFunctions
  - constraints.h, [131](#)
- storePrec
  - pulsar, [98](#)
- storePrecision, [111](#)
  - comment, [111](#)
  - minPrec, [111](#)
  - routine, [111](#)

- tempo2.h, [205](#)
- sun\_earth
  - observation, [61](#)
- sun\_ssb
  - observation, [61](#)
- swap\_bytes
  - jpl\_eph\_data, [51](#)
- swapByte
  - read\_fortran.h, [156](#)
- swapByte2
  - read\_fortran2.h, [158](#)
- swm
  - pulsar, [98](#)
- t1
  - T2Predictor, [116](#)
- T1Polyco, [112](#)
  - binary\_frequency, [112](#)
  - binary\_phase, [112](#)
  - coeff, [112](#)
  - date\_string, [113](#)
  - dm, [113](#)
  - doppler, [113](#)
  - frequency\_obs, [113](#)
  - frequency\_psr\_0, [113](#)
  - log10rms, [113](#)
  - mjd\_mid, [113](#)
  - ncoeff, [113](#)
  - psrname, [114](#)
  - reference\_phase, [114](#)
  - sitename, [114](#)
  - span, [114](#)
  - utc\_string, [114](#)
- T1Polyco\_GetFrequency
  - tempo2pred\_int.h, [243](#)
- T1Polyco\_GetPhase
  - tempo2pred\_int.h, [244](#)
- T1Polyco\_Read
  - tempo2pred\_int.h, [244](#)
- T1Polyco\_Write
  - tempo2pred\_int.h, [244](#)
- T1PolycoSet, [114](#)
  - nsegments, [115](#)
  - segments, [115](#)
- T1PolycoSet\_Destroy
  - tempo2pred\_int.h, [244](#)
- T1PolycoSet\_GetFrequency
  - tempo2pred\_int.h, [244](#)
- T1PolycoSet\_GetNearest
  - tempo2pred\_int.h, [244](#)
- T1PolycoSet\_GetPhase
  - tempo2pred\_int.h, [245](#)
- T1PolycoSet\_Read
  - tempo2pred\_int.h, [245](#)
- T1PolycoSet\_Write
  - tempo2pred\_int.h, [245](#)
- T2\_PTAmode1
  - tempo2.h, [224](#)
- T2C\_IAU2000B
  - tempo2.h, [202](#)
- T2C\_TEMPO
  - tempo2.h, [202](#)
- t2Fit
  - t2fit.h, [160](#)
- t2Fit\_buildConstraintsMatrix
  - t2fit.h, [161](#)
- t2Fit\_buildDesignMatrix
  - t2fit.h, [161](#)
- t2Fit\_fillFitInfo
  - t2fit.h, [161](#)
- t2Fit\_fillGlobalFitInfo
  - t2fit.h, [161](#)
- t2Fit\_getFitData
  - t2fit.h, [161](#)
- t2Fit\_getParamDeriv
  - t2fit.h, [162](#)
- t2Fit\_getParamMatrixRow
  - t2fit.h, [162](#)
- t2Fit\_updateParameters
  - t2fit.h, [162](#)
- t2FitFunc\_binaryModels
  - t2fit\_stdFitFuncs.h, [174](#)
- t2FitFunc\_dmmodelCM
  - t2fit\_dmmodel.h, [163](#)
- t2FitFunc\_dmmodelDM
  - t2fit\_dmmodel.h, [163](#)
- t2FitFunc\_dmsinusoids
  - t2fit\_dmother.h, [164](#)
- t2FitFunc\_dmx
  - t2fit\_dmother.h, [164](#)
- t2FitFunc\_fd
  - t2fit\_dmother.h, [164](#)
- t2FitFunc\_fddc
  - t2fit\_dmother.h, [164](#)
- t2FitFunc\_fitwaves
  - t2fit\_fitwaves.h, [165](#)
- t2FitFunc\_gwb\_amp
  - t2fit\_gw.h, [167](#)
- t2FitFunc\_gwcs\_amp
  - t2fit\_gw.h, [167](#)
- t2FitFunc\_gwm\_amp
  - t2fit\_gw.h, [167](#)
- t2FitFunc\_gwsingle
  - t2fit\_gw.h, [168](#)
- t2FitFunc\_ifunc
  - t2fit\_ifunc.h, [169](#)
  - t2fit\_stdFitFuncs.h, [174](#)
- t2FitFunc\_jump
  - t2fit\_stdFitFuncs.h, [175](#)
- t2FitFunc\_ne\_sw
  - t2fit\_dmother.h, [165](#)
- t2FitFunc\_nestlike\_band
  - t2fit\_nestlike.h, [170](#)
- t2FitFunc\_nestlike\_group
  - t2fit\_nestlike.h, [170](#)
- t2FitFunc\_nestlike\_jitter
  - t2fit\_nestlike.h, [171](#)



t2FitFunc\_nestlike\_red  
     t2fit\_nestlike.h, 171  
 t2FitFunc\_nestlike\_red\_dm  
     t2fit\_nestlike.h, 171  
 t2FitFunc\_notImplemented  
     t2fit\_stdFitFuncs.h, 175  
 t2FitFunc\_planet  
     t2fit\_stdFitFuncs.h, 175  
 t2FitFunc\_quad\_om  
     t2fit\_gw.h, 168  
 t2FitFunc\_sifunc  
     t2fit\_ifunc.h, 169  
 t2FitFunc\_stdDm  
     t2fit\_stdFitFuncs.h, 175  
 t2FitFunc\_stdFreq  
     t2fit\_stdFitFuncs.h, 176  
 t2FitFunc\_stdGlitch  
     t2fit\_glitch.h, 166  
 t2FitFunc\_stdPosition  
     t2fit\_position.h, 173  
 t2FitFunc\_telPos  
     t2fit\_stdFitFuncs.h, 176  
 t2FitFunc\_telPos\_delta  
     t2fit\_stdFitFuncs.h, 176  
 t2FitFunc\_zero  
     t2fit\_stdFitFuncs.h, 176  
 T2Predictor, 115  
     cheby, 115  
     kind, 116  
     modelset, 116  
     t1, 116  
 T2Predictor\_Copy  
     tempo2pred.h, 235  
 T2Predictor\_Destroy  
     tempo2pred.h, 235  
 T2Predictor\_FRead  
     tempo2pred.h, 235  
 T2Predictor\_FWrite  
     tempo2pred.h, 235  
 T2Predictor\_GetEndFreq  
     tempo2pred.h, 235  
 T2Predictor\_GetEndMJD  
     tempo2pred.h, 235  
 T2Predictor\_GetFrequency  
     tempo2pred.h, 236  
 T2Predictor\_GetPSRName  
     tempo2pred.h, 236  
 T2Predictor\_GetPhase  
     tempo2pred.h, 236  
 T2Predictor\_GetPlan  
     tempo2pred.h, 236  
 T2Predictor\_GetPlan\_Ext  
     tempo2pred.h, 236  
 T2Predictor\_GetSiteName  
     tempo2pred.h, 237  
 T2Predictor\_GetStartFreq  
     tempo2pred.h, 237  
 T2Predictor\_GetStartMJD  
     tempo2pred.h, 237  
 T2Predictor\_Init  
     tempo2pred.h, 237  
 T2Predictor\_Insert  
     tempo2pred.h, 237  
 T2Predictor\_Keep  
     tempo2pred.h, 237  
 T2Predictor\_Kind  
     tempo2pred.h, 237  
 T2Predictor\_Read  
     tempo2pred.h, 238  
 T2Predictor\_Write  
     tempo2pred.h, 238  
 T2PredictorKind  
     tempo2pred.h, 234  
 t2UpdateFunc\_binaryModels  
     t2fit\_stdFitFuncs.h, 177  
 t2UpdateFunc\_dmmodelCM  
     t2fit\_dmmodel.h, 163  
 t2UpdateFunc\_dmmodelDM  
     t2fit\_dmmodel.h, 163  
 t2UpdateFunc\_fitwaves  
     t2fit\_fitwaves.h, 166  
 t2UpdateFunc\_gwsingle  
     t2fit\_gw.h, 168  
 t2UpdateFunc\_ifunc  
     t2fit\_ifunc.h, 169  
     t2fit\_stdFitFuncs.h, 177  
 t2UpdateFunc\_jump  
     t2fit\_stdFitFuncs.h, 177  
 t2UpdateFunc\_ne\_sw  
     t2fit\_dmother.h, 165  
 t2UpdateFunc\_nestlike\_band  
     t2fit\_nestlike.h, 171  
 t2UpdateFunc\_nestlike\_group  
     t2fit\_nestlike.h, 172  
 t2UpdateFunc\_nestlike\_jitter  
     t2fit\_nestlike.h, 172  
 t2UpdateFunc\_nestlike\_red  
     t2fit\_nestlike.h, 172  
 t2UpdateFunc\_nestlike\_red\_dm  
     t2fit\_nestlike.h, 172  
 t2UpdateFunc\_notImplemented  
     t2fit\_stdFitFuncs.h, 177  
 t2UpdateFunc\_quad\_om  
     t2fit\_gw.h, 168  
 t2UpdateFunc\_simpleAdd  
     t2fit\_stdFitFuncs.h, 178  
 t2UpdateFunc\_simpleMinus  
     t2fit\_stdFitFuncs.h, 178  
 t2UpdateFunc\_stdFreq  
     t2fit\_stdFitFuncs.h, 178  
 t2UpdateFunc\_stdGlitch  
     t2fit\_glitch.h, 166  
 t2UpdateFunc\_stdPosition  
     t2fit\_position.h, 173  
 t2UpdateFunc\_telPos\_delta  
     t2fit\_stdFitFuncs.h, 178

- t2UpdateFunc\_zero
  - t2fit\_stdFitFuncs.h, 179
- T2accel.h, 158
  - ACCEL\_LSQ, 159
  - ACCEL\_MULTMATRIX, 159
  - ACCEL\_UINV, 159
  - accel\_lsq\_qr, 159
  - accel\_multMatrix, 159
  - accel\_multMatrixVec, 159
  - accel\_uinv, 160
  - useT2accel, 160
- t2cMethod
  - pulsar, 99
- T2efacFlagID
  - pulsar, 99
- T2efacFlagVal
  - pulsar, 99
- T2efacVal
  - pulsar, 99
- T2equadFlagID
  - pulsar, 99
- T2equadFlagVal
  - pulsar, 99
- T2equadVal
  - pulsar, 99
- t2fit.h, 160
  - t2Fit, 160
  - t2Fit\_buildConstraintsMatrix, 161
  - t2Fit\_buildDesignMatrix, 161
  - t2Fit\_fillFitInfo, 161
  - t2Fit\_fillGlobalFitInfo, 161
  - t2Fit\_getFitData, 161
  - t2Fit\_getParamDeriv, 162
  - t2Fit\_getParamMatrixRow, 162
  - t2Fit\_updateParameters, 162
- t2fit\_dmmodel.h, 162
  - t2FitFunc\_dmmodelCM, 163
  - t2FitFunc\_dmmodelDM, 163
  - t2UpdateFunc\_dmmodelCM, 163
  - t2UpdateFunc\_dmmodelDM, 163
- t2fit\_dmother.h, 164
  - t2FitFunc\_dmsinusoids, 164
  - t2FitFunc\_dmx, 164
  - t2FitFunc\_fd, 164
  - t2FitFunc\_fddc, 164
  - t2FitFunc\_ne\_sw, 165
  - t2UpdateFunc\_ne\_sw, 165
- t2fit\_fitwaves.h, 165
  - t2FitFunc\_fitwaves, 165
  - t2UpdateFunc\_fitwaves, 166
- t2fit\_glitch.h, 166
  - t2FitFunc\_stdGlitch, 166
  - t2UpdateFunc\_stdGlitch, 166
- t2fit\_gw.h, 167
  - t2FitFunc\_gwb\_amp, 167
  - t2FitFunc\_gwcs\_amp, 167
  - t2FitFunc\_gwm\_amp, 167
  - t2FitFunc\_gwsingle, 168
- t2FitFunc\_quad\_om, 168
- t2UpdateFunc\_gwsingle, 168
- t2UpdateFunc\_quad\_om, 168
- t2fit\_ifunc.h, 169
  - t2FitFunc\_ifunc, 169
  - t2FitFunc\_sifunc, 169
  - t2UpdateFunc\_ifunc, 169
- t2fit\_nestlike.h, 170
  - t2FitFunc\_nestlike\_band, 170
  - t2FitFunc\_nestlike\_group, 170
  - t2FitFunc\_nestlike\_jitter, 171
  - t2FitFunc\_nestlike\_red, 171
  - t2FitFunc\_nestlike\_red\_dm, 171
  - t2UpdateFunc\_nestlike\_band, 171
  - t2UpdateFunc\_nestlike\_group, 172
  - t2UpdateFunc\_nestlike\_jitter, 172
  - t2UpdateFunc\_nestlike\_red, 172
  - t2UpdateFunc\_nestlike\_red\_dm, 172
- t2fit\_position.h, 173
  - t2FitFunc\_stdPosition, 173
  - t2UpdateFunc\_stdPosition, 173
- t2fit\_stdFitFuncs.h, 174
  - t2FitFunc\_binaryModels, 174
  - t2FitFunc\_ifunc, 174
  - t2FitFunc\_jump, 175
  - t2FitFunc\_notImplemented, 175
  - t2FitFunc\_planet, 175
  - t2FitFunc\_stdDm, 175
  - t2FitFunc\_stdFreq, 176
  - t2FitFunc\_telPos, 176
  - t2FitFunc\_telPos\_delta, 176
  - t2FitFunc\_zero, 176
  - t2UpdateFunc\_binaryModels, 177
  - t2UpdateFunc\_ifunc, 177
  - t2UpdateFunc\_jump, 177
  - t2UpdateFunc\_notImplemented, 177
  - t2UpdateFunc\_simpleAdd, 178
  - t2UpdateFunc\_simpleMinus, 178
  - t2UpdateFunc\_stdFreq, 178
  - t2UpdateFunc\_telPos\_delta, 178
  - t2UpdateFunc\_zero, 179
- T2globalEfac
  - pulsar, 99
- T2model
  - tempo2.h, 225
- T2toolkit.h, 179
  - genrand\_int32, 180
  - genrand\_real1, 180
  - init\_genrand, 180
  - TKconvertFloat1, 180
  - TKconvertFloat2, 181
  - TKfindMax\_d, 181
  - TKfindMax\_f, 181
  - TKfindMedian\_d, 181
  - TKfindMedian\_f, 181
  - TKfindMin\_d, 181
  - TKfindMin\_f, 182
  - TKfindRMS\_d, 182

- TKfindRMS\_f, 182
- TKfindRMSweight\_d, 182
- TKgaussDev, 182
- TKmean\_d, 182
- TKmean\_f, 183
- TKranDev, 183
- TKrange\_d, 183
- TKrange\_f, 183
- TKretMax\_d, 183
- TKretMax\_f, 183
- TKretMin\_d, 184
- TKretMin\_f, 184
- TKretMin\_i, 184
- TKsetSeed, 184
- TKsign\_d, 184
- TKsort\_2f, 184
- TKsort\_3d, 185
- TKsort\_d, 185
- TKsort\_f, 185
- TKvariance\_d, 185
- TKzeromean\_d, 185
- TDB\_UNITS
  - tempo2.h, 202
- TDBTDT\_FILE
  - tempo2.h, 202
- TEMPO2\_ARCH
  - config.h, 127
- TEMPO2\_ENVIRON
  - tempo2.h, 233
- TEMPO2\_h\_HASH
  - tempo2.h, 202
- TEMPO2\_h\_MAJOR\_VER
  - tempo2.h, 202
- TEMPO2\_h\_MINOR\_VER
  - tempo2.h, 202
- TEMPO2\_h\_VER
  - tempo2.h, 202
- TK\_MAX\_ERROR\_LEN
  - TKlog.h, 255
- TK\_MAX\_ERRORS
  - TKlog.h, 255
- TK\_STORE\_ERROR
  - TKlog.h, 256
- TK\_STORE\_WARNING
  - TKlog.h, 256
- TK\_dft
  - TKspectrum.h, 273
- TK\_errorCount
  - TKlog.h, 258
- TK\_errorlog
  - TKlog.h, 258
- TK\_fft
  - TKspectrum.h, 273
- TK\_fitSine
  - TKspectrum.h, 273
- TK\_fitSinusoids
  - TKspectrum.h, 273
- TK\_warnCount
  - TKlog.h, 258
- TK\_warnlog
  - TKlog.h, 258
- TK\_weightLS
  - TKspectrum.h, 274
- TKaveragePts
  - TKspectrum.h, 274
- TKbacksubstitution\_svd
  - TKsvd.h, 277
- TKbidiagonal
  - TKsvd.h, 277
- TKboxcar
  - TKspectrum.h, 274
- TKcholesky.h, 246
  - cholesky\_covarFunc2matrix, 246
  - cholesky\_dmModel, 246
  - cholesky\_dmModelCovarParam, 247
  - cholesky\_ecm, 247
  - cholesky\_formUinv, 247
  - cholesky\_powerlawModel, 247
  - cholesky\_powerlawModel\_withBeta, 248
  - cholesky\_readFromCovarianceFunction, 248
- TKcmonot
  - TKspectrum.h, 274
- TKconstrainedLeastSquares
  - TKfit.h, 249
- TKconvertFloat1
  - T2toolkit.h, 180
- TKconvertFloat2
  - T2toolkit.h, 181
- TKfindMax\_d
  - T2toolkit.h, 181
- TKfindMax\_f
  - T2toolkit.h, 181
- TKfindMedian\_d
  - T2toolkit.h, 181
- TKfindMedian\_f
  - T2toolkit.h, 181
- TKfindMin\_d
  - T2toolkit.h, 181
- TKfindMin\_f
  - T2toolkit.h, 182
- TKfindPoly\_d
  - TKfit.h, 249
- TKfindRMS\_d
  - T2toolkit.h, 182
- TKfindRMS\_f
  - T2toolkit.h, 182
- TKfindRMSweight\_d
  - T2toolkit.h, 182
- TKfirstDifference
  - TKspectrum.h, 275
- TKfit.h, 248
  - TKconstrainedLeastSquares, 249
  - TKfindPoly\_d, 249
  - TKfitPoly, 249
  - TKleastSquares, 250
  - TKleastSquares\_svd, 250

- TKleastSquares\_svd\_noErr, [250](#)
- TKremovePoly\_d, [250](#)
- TKremovePoly\_f, [251](#)
- TKrobustConstrainedLeastSquares, [251](#)
- TKrobustDefConstrainedLeastSquares, [251](#)
- TKrobustLeastSquares, [252](#)
- TKfitPoly
  - TKfit.h, [249](#)
- TKgaussDev
  - T2toolkit.h, [182](#)
- TKhann
  - TKspectrum.h, [275](#)
- TKinterpolateSplineSmoothFixedXPts
  - TKspectrum.h, [275](#)
- TKleastSquares
  - TKfit.h, [250](#)
- TKleastSquares\_svd
  - TKfit.h, [250](#)
- TKleastSquares\_svd\_noErr
  - TKfit.h, [250](#)
- TKlog.h, [252](#)
  - \_LOG, [253](#)
  - \_TKchklog, [257](#)
  - BOLDCOLOR, [253](#)
  - DEPRECATED, [253](#)
  - debugFlag, [257](#)
  - ENDERR, [254](#)
  - ENDL, [254](#)
  - ERRORCOLOR, [254](#)
  - LOG\_OUTFILE, [254](#)
  - logall, [254](#)
  - logdbg, [254](#)
  - logerr, [254](#)
  - logerr\_check, [257](#)
  - logmsg, [255](#)
  - logtchk, [255](#)
  - logwarn, [255](#)
  - quietFlag, [257](#)
  - RESETCOLOR, [255](#)
  - TK\_MAX\_ERROR\_LEN, [255](#)
  - TK\_MAX\_ERRORS, [255](#)
  - TK\_STORE\_ERROR, [256](#)
  - TK\_STORE\_WARNING, [256](#)
  - TK\_errorCount, [258](#)
  - TK\_errorlog, [258](#)
  - TK\_warnCount, [258](#)
  - TK\_warnlog, [258](#)
  - tcheck, [257](#)
  - timer\_clk, [258](#)
  - WARNCOLOR, [256](#)
  - WHEREARG, [256](#)
  - WHEREERR, [256](#)
  - WHERESTR, [256](#)
  - WHERECHK, [256](#)
  - WHEREWARN, [257](#)
  - writeResiduals, [258](#)
- TKlomb\_d
  - TKspectrum.h, [275](#)
- TKlongdouble.float128.h, [258](#)
  - cosl, [259](#)
  - FMT\_LD, [259](#)
  - fabsl, [259](#)
  - floorl, [259](#)
  - LD\_PI, [260](#)
  - LONGDOUBLE\_IS\_FLOAT128, [260](#)
  - LONGDOUBLE\_ONE, [260](#)
  - ld\_fprintf, [261](#)
  - ld\_printf, [261](#)
  - ld\_sprintf, [261](#)
  - longdouble, [260](#), [261](#)
  - parse\_longdouble, [261](#)
  - powl, [260](#)
  - sinl, [260](#)
  - USE\_BUILTIN\_LONGDOUBLE, [260](#)
- TKlongdouble.h, [262](#)
  - LD\_PI, [262](#)
  - LONGDOUBLE\_IS\_IEEE754, [263](#)
  - LONGDOUBLE\_ONE, [263](#)
  - ld\_fprintf, [262](#)
  - ld\_printf, [262](#)
  - ld\_sprintf, [262](#)
  - longdouble, [263](#)
  - parse\_longdouble, [263](#)
  - USE\_BUILTIN\_LONGDOUBLE, [263](#)
- TKlongdouble.ld.h, [264](#)
  - LD\_PI, [264](#)
  - LONGDOUBLE\_IS\_IEEE754, [265](#)
  - LONGDOUBLE\_ONE, [265](#)
  - ld\_fprintf, [264](#)
  - ld\_printf, [264](#)
  - ld\_sprintf, [264](#)
  - longdouble, [265](#)
  - parse\_longdouble, [265](#)
  - USE\_BUILTIN\_LONGDOUBLE, [265](#)
- TKmatrix.h, [266](#)
  - free\_2df, [266](#)
  - free\_blas, [266](#)
  - free\_uinv, [266](#)
  - get\_blas\_cols, [266](#)
  - get\_blas\_rows, [267](#)
  - malloc\_2df, [267](#)
  - malloc\_blas, [267](#)
  - malloc\_uinv, [267](#)
  - TKmultMatrix, [267](#)
  - TKmultMatrix\_sq, [267](#)
  - TKmultMatrixVec, [268](#)
  - TKmultMatrixVec\_sq, [268](#)
- TKmean\_d
  - T2toolkit.h, [182](#)
- TKmean\_f
  - T2toolkit.h, [183](#)
- TKmultMatrix
  - TKmatrix.h, [267](#)
- TKmultMatrix\_sq
  - TKmatrix.h, [267](#)
- TKmultMatrixVec

- TKmatrix.h, 268
- TKmultMatrixVec\_sq
  - TKmatrix.h, 268
- TKpythag
  - TKsvd.h, 277
- TKranDev
  - T2toolkit.h, 183
- TKrange\_d
  - T2toolkit.h, 183
- TKrange\_f
  - T2toolkit.h, 183
- TKremovePoly\_d
  - TKfit.h, 250
- TKremovePoly\_f
  - TKfit.h, 251
- TKretMax\_d
  - T2toolkit.h, 183
- TKretMax\_f
  - T2toolkit.h, 183
- TKretMin\_d
  - T2toolkit.h, 184
- TKretMin\_f
  - T2toolkit.h, 184
- TKretMin\_i
  - T2toolkit.h, 184
- TKrobust
  - TKrobust.h, 268
- TKrobust.h, 268
  - TKrobust, 268
- TKrobustConstrainedLeastSquares
  - TKfit.h, 251
- TKrobustDefConstrainedLeastSquares
  - TKfit.h, 251
- TKrobustLeastSquares
  - TKfit.h, 252
- TKsetSeed
  - T2toolkit.h, 184
- TKsign\_d
  - T2toolkit.h, 184
- TKsingularValueDecomposition\_Isq
  - TKsvd.h, 278
- TKsort\_2f
  - T2toolkit.h, 184
- TKsort\_3d
  - T2toolkit.h, 185
- TKsort\_d
  - T2toolkit.h, 185
- TKsort\_f
  - T2toolkit.h, 185
- TKsortit
  - TKspectrum.h, 275
- TKspectrum
  - TKspectrum.h, 276
- TKspectrum.h, 269
  - ABS, 270
  - calcSpectra, 271
  - calcSpectraErr, 271
  - calcSpectraErr\_complex, 271
  - complexVal, 271
  - fit4, 271
  - getprtj, 272
  - getweights, 272
  - indexx8, 272
  - MAX, 270
  - MIN, 270
  - mat20, 272
  - sineFunc, 272
  - TK\_dft, 273
  - TK\_fft, 273
  - TK\_fitSine, 273
  - TK\_fitSinusoids, 273
  - TK\_weightLS, 274
  - TKaveragePts, 274
  - TKboxcar, 274
  - TKcmonot, 274
  - TKfirstDifference, 275
  - TKhann, 275
  - TKinterpolateSplineSmoothFixedXPts, 275
  - TKlomb\_d, 275
  - TKsortit, 275
  - TKspectrum, 276
  - TKspline\_interpolate, 276
  - verbose\_calc\_spectra, 276
- TKspline\_interpolate
  - TKspectrum.h, 276
- TKsvd.h, 277
  - TKbacksubstitution\_svd, 277
  - TKbidiagonal, 277
  - TKpythag, 277
  - TKsingularValueDecomposition\_Isq, 278
- TKvariance\_d
  - T2toolkit.h, 185
- TKzeromean\_d
  - T2toolkit.h, 185
- TNBandDMamp
  - pulsar, 102
- TNBandDMGam
  - pulsar, 102
- TNBandDMC
  - pulsar, 102
- TNBandNoiseAmp
  - pulsar, 102
- TNBandNoiseGam
  - pulsar, 102
- TNBandNoiseHF
  - pulsar, 102
- TNBandNoiseLF
  - pulsar, 103
- TNBandNoiseC
  - pulsar, 102
- TNDMAmp
  - pulsar, 103
- TNDMCoeffs
  - pulsar, 103
- TNDMErr
  - observation, 62

- TNDMEvAmp
  - pulsar, [103](#)
- TNDMEvGam
  - pulsar, [103](#)
- TNDMEvLength
  - pulsar, [103](#)
- TNDMEvLin
  - pulsar, [103](#)
- TNDMEvOff
  - pulsar, [104](#)
- TNDMEvQuad
  - pulsar, [104](#)
- TNDMEvStart
  - pulsar, [104](#)
- TNDMGam
  - pulsar, [104](#)
- TNDMSignal
  - observation, [62](#)
- TNDMC
  - pulsar, [103](#)
- TNECORRFlagID
  - pulsar, [104](#)
- TNECORRFlagVal
  - pulsar, [104](#)
- TNECORRVal
  - pulsar, [104](#)
- TNEFFFlagID
  - pulsar, [104](#)
- TNEFFFlagVal
  - pulsar, [105](#)
- TNEFVal
  - pulsar, [105](#)
- TNEQFlagID
  - pulsar, [105](#)
- TNEQFlagVal
  - pulsar, [105](#)
- TNEQVal
  - pulsar, [105](#)
- TNGlobalEF
  - pulsar, [105](#)
- TNGlobalEQ
  - pulsar, [105](#)
- TNGroupErr
  - observation, [62](#)
- TNGroupNoiseAmp
  - pulsar, [105](#)
- TNGroupNoiseFlagID
  - pulsar, [106](#)
- TNGroupNoiseFlagVal
  - pulsar, [106](#)
- TNGroupNoiseGam
  - pulsar, [106](#)
- TNGroupNoiseC
  - pulsar, [106](#)
- TNGroupSignal
  - observation, [62](#)
- TNRedAmp
  - pulsar, [106](#)
- TNRedCoeffs
  - pulsar, [106](#)
- TNRedCorner
  - pulsar, [106](#)
- TNRedErr
  - observation, [62](#)
- TNRedFlow
  - pulsar, [107](#)
- TNRedGam
  - pulsar, [107](#)
- TNRedSignal
  - observation, [63](#)
- TNRedC
  - pulsar, [106](#)
- TNSQFlagID
  - pulsar, [107](#)
- TNSQFlagVal
  - pulsar, [107](#)
- TNSQVal
  - pulsar, [108](#)
- TNShapeletEvFScale
  - pulsar, [107](#)
- TNShapeletEvPos
  - pulsar, [107](#)
- TNShapeletEvWidth
  - pulsar, [107](#)
- TNShapeletEvN
  - pulsar, [107](#)
- TNsubtractDM
  - pulsar, [108](#)
- TNsubtractRed
  - pulsar, [108](#)
- tOffset
  - pulsar, [108](#)
- tOffset\_f1
  - pulsar, [108](#)
- tOffset\_f2
  - pulsar, [108](#)
- tOffset\_t1
  - pulsar, [108](#)
- tOffset\_t2
  - pulsar, [109](#)
- tOffsetFlags
  - pulsar, [109](#)
- tOffsetSite
  - pulsar, [109](#)
- TSUN
  - tempo2.h, [203](#)
- TabulatedFunction, [116](#)
  - fileName, [116](#)
  - header\_line, [117](#)
  - samples, [117](#)
- TabulatedFunction\_getEndX
  - tabulatedfunction.h, [186](#)
- TabulatedFunction\_getStartX
  - tabulatedfunction.h, [186](#)
- TabulatedFunction\_getValue
  - tabulatedfunction.h, [186](#)

- TabulatedFunction\_load
  - tabulatedfunction.h, 186
- TabulatedFunctionSample, 117
  - x, 117
  - y, 117
- tabulatedfunction.h, 186
  - TabulatedFunction\_getEndX, 186
  - TabulatedFunction\_getStartX, 186
  - TabulatedFunction\_getValue, 186
  - TabulatedFunction\_load, 186
- tai2tt
  - tempo2.h, 225
- tai2ut1
  - tempo2.h, 225
- tcheck
  - TKlog.h, 257
- tdis1
  - observation, 62
- tdis2
  - observation, 62
- telDX\_e
  - pulsar, 100
- telDX\_t
  - pulsar, 100
- telDX\_v
  - pulsar, 100
- telDX\_vel
  - pulsar, 100
- telDX\_vel\_e
  - pulsar, 100
- telDY\_e
  - pulsar, 100
- telDY\_t
  - pulsar, 100
- telDY\_v
  - pulsar, 100
- telDY\_vel
  - pulsar, 101
- telDY\_vel\_e
  - pulsar, 101
- telDZ\_e
  - pulsar, 101
- telDZ\_t
  - pulsar, 101
- telDZ\_v
  - pulsar, 101
- telDZ\_vel
  - pulsar, 101
- telDZ\_vel\_e
  - pulsar, 101
- telID
  - observation, 62
- tempo1
  - pulsar, 101
- tempo2.h, 187
  - AU\_DIST, 193
  - AULTSC, 193
  - allocateMemory, 210
  - autoConstraints, 210
  - BIG\_G, 193
  - BTJmodel, 210
  - BTXmodel, 210
  - BTmodel, 210
  - bootstrap, 210
  - CVSdisplayVersion, 211
  - calcRMS, 211
  - calculate\_bclt, 211
  - compute\_tropospheric\_delays, 211
  - constraint, 205
  - constraint\_label, 203
  - constraintDerivFunc, 203
  - copyPSR, 211
  - copyParam, 211
  - covarFuncFile, 232
  - DDGRmodel, 212
  - DDHmodel, 212
  - DDKmodel, 212
  - DDSmodel, 212
  - DDmodel, 212
  - DM\_CONST\_SI, 194
  - DM\_CONST, 193
  - dcmFile, 232
  - defineClockCorrectionSequence, 213
  - destroyMemory, 213
  - destroyOne, 213
  - displayCVSversion, 232
  - displayMsg, 213
  - displayParameters, 213
  - dm\_delays, 213
  - dms\_turn, 214
  - doFitAll, 214
  - dotproduct, 214
  - ECLIPTIC\_OBLIQUITY\_VAL, 194
  - ECLIPTIC\_OBLIQUITY, 232
  - ELL1Hmodel, 214
  - ELL1kmodel, 214
  - ELL1model, 215
  - equ2ecl, 215
  - FB90\_TIMEEPH, 194
  - FitInfo, 203
  - FitOutput, 203
  - forceGlobalFit, 232
  - formBats, 215
  - formBatsAll, 215
  - formResiduals, 215
  - fortran\_mod, 215
  - fortran\_nint, 216
  - fortran\_nlong, 216
  - GM\_C3, 194
  - GMJ\_C3, 194
  - GMN\_C3, 194
  - GMS\_C3, 194
  - GMU\_C3, 195
  - GMV\_C3, 195
  - get\_EOP, 216
  - get\_OneobsCoord, 216

- get\_obsCoord, 216
- get\_obsCoord\_IAU2000B, 216
- getCholeskyMatrix, 217
- getClockCorrections, 217
- getCorrection, 217
- getCorrectionTT, 217
- getInputs, 217
- getObservatory, 218
- getParamDeriv, 218
- getParameterValue, 218
- GM, 194
- HAVE\_GWSIM\_H, 195
- hms\_turn, 218
- IF99\_TIMEEPH, 195
- IFTEPH\_FILE, 195
- id\_residual, 219
- initialise, 219
- initialiseOne, 219
- JVmodel, 219
- LEAPSECOND\_FILE, 195
- label, 206
- logicFlag, 219
- lookup\_observatory\_alias, 219
- MASYR2RADS, 195
- MAX\_BPJ\_JUMPS, 195
- MAX\_CLK\_CORR, 196
- MAX\_CLKCORR, 196
- MAX\_COEFF, 196
- MAX\_COMPANIONS, 196
- MAX\_DM\_DERIVATIVES, 196
- MAX\_DMX, 196
- MAX\_FILELEN, 196
- MAX\_FIT, 196
- MAX\_FLAG\_LEN, 197
- MAX\_FLAGS, 197
- MAX\_FREQ\_DERIVATIVES, 197
- MAX\_IFUNC, 197
- MAX\_JUMPS, 197
- MAX\_LEAPSEC, 197
- MAX\_MSG, 197
- MAX\_OBSN\_VAL, 197
- MAX\_OBSN, 232
- MAX\_PARAMS, 198
- MAX\_PSR\_VAL, 198
- MAX\_PSR, 232
- MAX\_QUAD, 198
- MAX\_SITE, 198
- MAX\_STOREPRECISION, 198
- MAX\_STRLEN, 198
- MAX\_SX, 198
- MAX\_T2EFAC, 198
- MAX\_T2EQUAD, 199
- MAX\_TEL\_CLK\_OFFS, 199
- MAX\_TEL\_DX, 199
- MAX\_TEL\_DY, 199
- MAX\_TEL\_DZ, 199
- MAX\_TNBN, 199
- MAX\_TNDMEv, 199
- MAX\_TNECORR, 199
- MAX\_TNEF, 200
- MAX\_TNEQ, 200
- MAX\_TNGN, 200
- MAX\_TNSQ, 200
- MAX\_TOFFSET, 200
- MAX\_WHITE, 200
- MSSmodel, 220
- NE\_SW\_DEFAULT, 200
- NEWFIT, 232
- OBLQ, 200
- OBSSYS\_FILE, 201
- observation, 204
- PCM, 201
- param\_label, 204
- paramDerivFunc, 204
- paramUpdateFunc, 204
- parameter, 204
- polyco, 220
- preProcess, 220
- preProcessSimple, 220
- preProcessSimple1, 220
- preProcessSimple2, 221
- preProcessSimple3, 221
- processFlag, 221
- processSimultaneous, 221
- pulsar, 204
- readEphemeris, 221
- readEphemeris\_calceph, 221
- readJBO\_bat, 222
- readObsFile, 222
- readOneEphemeris, 222
- readParfile, 222
- readParfileGlobal, 222
- readSimpleParfile, 223
- readTimfile, 223
- recordPrecision, 223
- SECDAYI, 201
- SECDAY, 201
- SI\_UNITS, 201
- SOLAR\_MASS, 201
- SOLAR\_RADIUS, 201
- SPEED\_LIGHT, 201
- secularMotion, 223
- setPlugPath, 223
- setStart, 223
- setupParameterFileDefaults, 224
- shapiro\_delay, 224
- simplePlot, 224
- solarWindModel, 224
- sortToAs, 224
- storePrecision, 205
- T2\_PTAmode, 224
- T2C\_IAU2000B, 202
- T2C\_TEMPO, 202
- T2model, 225
- TDB\_UNITS, 202
- TDBTDT\_FILE, 202



- TEMPO2\_ENVIRON, [233](#)
- TEMPO2\_h\_HASH, [202](#)
- TEMPO2\_h\_MAJOR\_VER, [202](#)
- TEMPO2\_h\_MINOR\_VER, [202](#)
- TEMPO2\_h\_VER, [202](#)
- TSUN, [203](#)
- tai2tt, [225](#)
- tai2ut1, [225](#)
- tempo2\_clock\_path, [233](#)
- tempo2\_plug\_path, [233](#)
- tempo2\_plug\_path\_len, [233](#)
- tempo2MachineType, [233](#)
- textOutput, [225](#)
- toa2utc, [225](#)
- transform\_units, [226](#)
- tt2tb, [226](#)
- tt2tb\_calceph, [226](#)
- turn\_deg, [226](#)
- turn\_dms, [226](#)
- turn\_hms, [226](#)
- UT1\_FILE, [203](#)
- updateBTJ, [227](#)
- updateBTX, [227](#)
- updateBatsAll, [227](#)
- updateBT, [227](#)
- updateDDGR, [228](#)
- updateDDH, [228](#)
- updateDDK, [228](#)
- updateDDS, [228](#)
- updateDD, [227](#)
- updateELL1, [228](#)
- updateELL1H, [229](#)
- updateELL1k, [229](#)
- updateEpoch, [229](#)
- updateEpoch\_str, [229](#)
- updateJV, [229](#)
- updateMSS, [230](#)
- updateT2, [230](#)
- updateT2\_PTA, [230](#)
- useSelectFile, [230](#)
- utc2tai, [230](#)
- vectorPulsar, [231](#)
- vectorscale, [231](#)
- vectorsum, [231](#)
- veryFast, [233](#)
- writeTim, [231](#)
- zoom\_graphics, [231](#)
- tempo2\_clock\_path
  - tempo2.h, [233](#)
- tempo2\_plug\_path
  - tempo2.h, [233](#)
- tempo2\_plug\_path\_len
  - tempo2.h, [233](#)
- tempo2MachineType
  - tempo2.h, [233](#)
- tempo2Util.h, [245](#)
  - dms\_turn, [245](#)
  - hms\_turn, [245](#)
  - turn\_deg, [246](#)
- tempo2pred.h, [234](#)
  - ChebyModelSet\_OutOfRange, [238](#)
  - T2Predictor\_Copy, [235](#)
  - T2Predictor\_Destroy, [235](#)
  - T2Predictor\_FRead, [235](#)
  - T2Predictor\_FWrite, [235](#)
  - T2Predictor\_GetEndFreq, [235](#)
  - T2Predictor\_GetEndMJD, [235](#)
  - T2Predictor\_GetFrequency, [236](#)
  - T2Predictor\_GetPSRName, [236](#)
  - T2Predictor\_GetPhase, [236](#)
  - T2Predictor\_GetPlan, [236](#)
  - T2Predictor\_GetPlan\_Ext, [236](#)
  - T2Predictor\_GetSiteName, [237](#)
  - T2Predictor\_GetStartFreq, [237](#)
  - T2Predictor\_GetStartMJD, [237](#)
  - T2Predictor\_Init, [237](#)
  - T2Predictor\_Insert, [237](#)
  - T2Predictor\_Keep, [237](#)
  - T2Predictor\_Kind, [237](#)
  - T2Predictor\_Read, [238](#)
  - T2Predictor\_Write, [238](#)
  - T2PredictorKind, [234](#)
- tempo2pred\_int.h, [238](#)
  - Cheby2D\_Construct, [239](#)
  - Cheby2D\_Construct\_x\_Derivative, [239](#)
  - Cheby2D\_Test, [240](#)
  - ChebyModel\_Construct, [240](#)
  - ChebyModel\_Copy, [240](#)
  - ChebyModel\_Destroy, [240](#)
  - ChebyModel\_GetFrequency, [240](#)
  - ChebyModel\_GetPhase, [240](#)
  - ChebyModel\_Init, [241](#)
  - ChebyModel\_Read, [241](#)
  - ChebyModel\_Test, [241](#)
  - ChebyModel\_Write, [241](#)
  - ChebyModelSet\_Construct, [241](#)
  - ChebyModelSet\_Destroy, [242](#)
  - ChebyModelSet\_GetFrequency, [242](#)
  - ChebyModelSet\_GetNearest, [242](#)
  - ChebyModelSet\_GetPhase, [242](#)
  - ChebyModelSet\_Init, [242](#)
  - ChebyModelSet\_Insert, [243](#)
  - ChebyModelSet\_Keep, [243](#)
  - ChebyModelSet\_Read, [243](#)
  - ChebyModelSet\_Test, [243](#)
  - ChebyModelSet\_Write, [243](#)
  - T1Polyco\_GetFrequency, [243](#)
  - T1Polyco\_GetPhase, [244](#)
  - T1Polyco\_Read, [244](#)
  - T1Polyco\_Write, [244](#)
  - T1PolycoSet\_Destroy, [244](#)
  - T1PolycoSet\_GetFrequency, [244](#)
  - T1PolycoSet\_GetNearest, [244](#)
  - T1PolycoSet\_GetPhase, [245](#)
  - T1PolycoSet\_Read, [245](#)
  - T1PolycoSet\_Write, [245](#)

- tensor\_alpha
  - gwgenSpec, 43
- tensor\_amp
  - gwgenSpec, 44
- textOutput
  - tempo2.h, 225
- theta\_bin
  - gwSrc, 46
  - gwgeneralSrc, 42
- theta\_g
  - gwSrc, 46
  - gwgeneralSrc, 42
- timeEphemeris
  - pulsar, 102
- timer\_clk
  - TKlog.h, 258
- ToAextraCovar
  - pulsar, 108
- toa2utc
  - tempo2.h, 225
- toaDMErr
  - observation, 63
- toaErr
  - observation, 63
- tobs
  - observation, 63
- torb
  - observation, 63
- totalNfit
  - FitOutput, 38
- transform\_units
  - tempo2.h, 226
- troposphericDelay
  - observation, 63
- tt2tb
  - tempo2.h, 226
- tt2tb\_calceph
  - tempo2.h, 226
- turn\_deg
  - tempo2.h, 226
  - tempo2Util.h, 246
- turn\_dms
  - tempo2.h, 226
- turn\_hms
  - tempo2.h, 226
- twot
  - interpolation\_info, 47
- tzrsite
  - pulsar, 109
- USE\_BUILTIN\_LONGDOUBLE
  - TKlongdouble.float128.h, 260
  - TKlongdouble.h, 263
  - TKlongdouble.ld.h, 265
- UT1\_FILE
  - tempo2.h, 203
- units
  - pulsar, 109
- updateBTJ
  - tempo2.h, 227
- updateBTX
  - tempo2.h, 227
- updateBatsAll
  - tempo2.h, 227
- updateBT
  - tempo2.h, 227
- updateDDGR
  - tempo2.h, 228
- updateDDH
  - tempo2.h, 228
- updateDDK
  - tempo2.h, 228
- updateDDS
  - tempo2.h, 228
- updateDD
  - tempo2.h, 227
- updateELL1
  - tempo2.h, 228
- updateELL1H
  - tempo2.h, 229
- updateELL1k
  - tempo2.h, 229
- updateEpoch
  - tempo2.h, 229
- updateEpoch\_str
  - tempo2.h, 229
- updateFunctions
  - FitInfo, 37
- updateJV
  - tempo2.h, 229
- updateMSS
  - tempo2.h, 230
- updateT2
  - tempo2.h, 230
- updateT2\_PTA
  - tempo2.h, 230
- uranus\_earth
  - observation, 63
- useCalceph
  - pulsar, 109
- useSelectFile
  - tempo2.h, 230
- useT2accel
  - T2accel.h, 160
- useTNOrth
  - pulsar, 109
- utc2tai
  - tempo2.h, 230
- utc\_string
  - T1Polyco, 114
- VERSION
  - config.h, 127
- val
  - constraint\_param\_info, 34
  - parameter, 68
- vectorPulsar
  - tempo2.h, 231

- vectorscale
  - tempo2.h, [231](#)
- vectorsum
  - tempo2.h, [231](#)
- vel\_coeff
  - interpolation\_info, [48](#)
- velPulsar
  - pulsar, [109](#)
- venus\_earth
  - observation, [63](#)
- verbose\_calc\_spectra
  - TKspectrum.h, [276](#)
- veryFast
  - tempo2.h, [233](#)
- vl\_alpha
  - gwgenSpec, [44](#)
- vl\_amp
  - gwgenSpec, [44](#)
- WARNCOLOR
  - TKlog.h, [256](#)
- WHEREARG
  - TKlog.h, [256](#)
- WHEREERR
  - TKlog.h, [256](#)
- WHERESTR
  - TKlog.h, [256](#)
- WHERECHK
  - TKlog.h, [256](#)
- WHEREWARN
  - TKlog.h, [257](#)
- wave\_cos
  - pulsar, [110](#)
- wave\_cos\_dm
  - pulsar, [110](#)
- wave\_cos\_dm\_err
  - pulsar, [110](#)
- wave\_cos\_err
  - pulsar, [110](#)
- wave\_sine
  - pulsar, [110](#)
- wave\_sine\_dm
  - pulsar, [110](#)
- wave\_sine\_dm\_err
  - pulsar, [110](#)
- wave\_sine\_err
  - pulsar, [110](#)
- waveScale
  - pulsar, [111](#)
- whiteNoiseModelFile
  - pulsar, [111](#)
- writeResiduals
  - TKlog.h, [258](#)
- writeTim
  - tempo2.h, [231](#)
- x
  - observatory, [65](#)
  - TabulatedFunctionSample, [117](#)
- y
  - observatory, [65](#)
  - TabulatedFunctionSample, [117](#)
- z
  - observatory, [65](#)
- zenith
  - observation, [64](#)
- zoom\_graphics
  - tempo2.h, [231](#)